Electronic Transactions on Numerical Analysis.
Volume 22, pp. 163-183, 2006.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

PROBING METHODS FOR SADDLE-POINT PROBLEMS*

CHRIS SIEFERT† AND ERIC DE STURLER‡

Abstract. Several Schur complement-based preconditioners have been proposed for solving (generalized) saddle-point problems. We consider matrices where the Schur complement has rapid decay over some graph known *a priori*. This occurs for many matrices arising from the discretization of systems of partial differential equations, and this graph is then related to the mesh. We propose the use of probing methods to approximate these Schur complements in preconditioners for saddle-point problems. We demonstrate these techniques for the block-diagonal and constraint preconditioners proposed by [Murphy, Golub and Wathen '00], [de Sturler and Liesen '04] and [Siefert and de Sturler '05]. However, these techniques are applicable to many other preconditioners as well. We discuss the implementation of probing methods, and we consider the application of those approximations in preconditioners for Navier-Stokes problems and metal deformation problems. Finally, we study eigenvalue clustering for the preconditioned matrices, and we present convergence and timing results for various problem sizes. These results demonstrate the effectiveness of the proposed preconditioners with probing-based approximate Schur complements.

Key words. saddle-point systems, constraint preconditioners, Krylov methods, Schur complements, probing

AMS subject classifications. 65F10, 65F50, 05C15

1. Introduction.

We are concerned with preconditioning problems of the form

$$\mathcal{A}\begin{bmatrix} x \\ y \end{bmatrix} \equiv \begin{bmatrix} A & B^T \\ C & D \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $D \in \mathbb{R}^{m\times m}$, and here $n > m$. These problems are referred to as (generalized) *saddle-point problems*. Specifically, we are concerned with problems where the Schur complement matrix has rapid decay over some *a priori* known graph. These problems can have a number of different characteristics. For some problems, especially those arising in constrained optimization, $D = 0$. For others, such as those arising from stabilized finite elements [2, 12, 24], $D \neq 0$, but $\|D\|$ is small. For still others, the non-zero D arises from another source, such as a very slight compressibility in metal deformation problems [29]. We consider problems where $D = 0$ or $\|D\|$ is small, so that our problems retain the character of a (generalized) saddle-point problem. In addition, certain finite element stabilization schemes yield $B \neq C$ [1, 22] and [24, Sections 7.5 and 9.4], while many other schemes yield $B = C$. We consider the problem and preconditioners in the generalized form, $B \neq C$, mainly to emphasize the general applicability of the proposed methods. However, the methods proposed in this paper are equally applicable to special cases, especially the symmetric ($B = C$) case. A number of preconditioners for this class of problems have been developed that employ a Schur complement [3, 9, 10, 18, 21], or some approximation thereof [11, 13, 19, 23, 27]. Section 2 will present a summary of the convergence results for one such family of preconditioners [27].

For some problems, an approximation to the Schur complement in the preconditioner that yields good convergence is known, like the pressure mass matrix for the Navier-Stokes

*Received April 7, 2005. Accepted for publication March 29, 2006. Recommended by R. Lehoucq. This material is based upon work supported by the National Science Foundation under Award No. DMR-0325939 through the Materials Computation Center at the University of Illinois at Urbana-Champaign (UIUC), with additional support by the US Department of Energy under Award No. DEFG02-91ER45439 through the Frederick Seitz Materials Research Laboratory at UIUC, and by the US Department of Energy under Award No. LLNL B341494 through the Center for Simulation of Advanced Rockets at UIUC.

†Sandia National Laboratories, MS 1110, P.O. Box 5800, Albuquerque, NM 87185-1110 (csiefer@sandia.gov).

‡Department of Mathematics, 460 McBryde, Virginia Tech, Blacksburg, VA 24061-0123 (sturler@vt.edu).

ETNA
Kent State University
etna@mcs.kent.edu

problem [28]. For other problems, however, no such approximation is known. Probing [4, 17] provides a general, algebraic method for building matrix approximations. Based on the key idea of exploiting decay properties in the underlying operator to efficiently capture large matrix entries, it was used successfully to build narrowly banded approximations to Schur complement matrices corresponding to 1-D interfaces arising in 2-D non-overlapping domain decomposition with an appropriate ordering of the unknowns [4]. Following the same underlying idea, we can extend this approach to applications with a more complicated decay structure. For these applications, we require approximations with a more general sparsity pattern. Recently, graph coloring techniques used by the optimization community for sparse Jacobian and Hessian approximations [6, 7, 15, 16, 20] have been adapted to allow the approximation of any matrix, so long as the sparsity pattern is known *a priori* [8].

We propose these techniques for approximating Schur complements that have entries which decay with distance on some underlying graph. For example, a graph derived from an underlying finite element mesh might be a candidate for these more advanced probing methods. This relationship will be explained in more detail in Sections 3 and 4.

A potential problem is that the matrices formed by these more general probing techniques may be expensive to factor. However, in previous experiments we found that replacing an exact Schur complement by an incomplete factorization has a negligible effect on convergence [27]. Therefore, we propose to use incomplete factorizations for these probing-based approximate Schur complements. This keeps the total cost of constructing and applying the approximate Schur complement to $O(m)$. Algorithmic details will be the focus of Section 5.

Approximating Schur complements by probing with general sparsity patterns combined with incomplete factorizations of the resulting approximations gives cheap and effective preconditioners with excellent performance. We demonstrate our preconditioners for two applications, the first in fluid flow and the second in metal deformation. Analysis of the eigenvalues of the preconditioned systems, as well as GMRES convergence results and timings will be provided in Section 6. Section 7 summarizes our conclusions and presents directions for future work.

2. Preconditioning Saddle-Point Problems. In [27], we developed two classes of preconditioners for (1.1) and provided eigenvalue bounds for the corresponding preconditioned systems that allow for the use of approximate Schur complements. Both classes of preconditioners require the choice of a splitting of the (1,1) block of (1.1), namely $A = F - E$, where F is cheap to solve with. We can make such a splitting even in the case where A is singular or ill-conditioned. Let $S_1 = -(D - CF^{-1}B^T)$ be the Schur complement of the matrix

$$\begin{bmatrix} F & B^T \\ C & D \end{bmatrix}.$$

We will refer to S_1 as the exact Schur complement for this preconditioner. Let S_2 be an approximation to S_1, and, let $\mathcal{E} = S_2^{-1}S_1 - I$. If we precondition from the left using the block-diagonal preconditioner from [27], we get,

$$\begin{bmatrix} F^{-1} & 0 \\ 0 & S_2^{-1} \end{bmatrix} \begin{bmatrix} A & B^T \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} I-T & N \\ M & Q \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ \tilde{g} \end{bmatrix}, \tag{2.1}$$

where $T = F^{-1}E$, $N = F^{-1}B^T$, $M = S_2^{-1}C$ and $Q = S_2^{-1}D$. We refer to this system as the block-diagonally preconditioned system. The second preconditioned system, which will be referred to as the *related system*, is derived from a further splitting of the block-diagonally preconditioned system (2.1) as follows [27],

$$\begin{bmatrix} I-T & N \\ M & Q \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \left(\begin{bmatrix} I & N \\ M & MN - I \end{bmatrix} - \begin{bmatrix} T & 0 \\ 0 & \mathcal{E} \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ \tilde{g} \end{bmatrix}. \tag{2.2}$$

Note that,

$$\begin{bmatrix} I & N \\ M & MN - I \end{bmatrix}^{-1} = \begin{bmatrix} I - NM & N \\ M & -I \end{bmatrix}. \tag{2.3}$$

Multiplying (2.2) by (2.3) yields

$$\begin{bmatrix} I - (I - NM)T & -N\mathcal{E} \\ -MT & I + \mathcal{E} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \hat{f} \\ \hat{g} \end{bmatrix}. \tag{2.4}$$

This is the related system for the fixed-point iteration associated with the splitting in (2.2). Multiplying by this matrix is more expensive than with the block-diagonally preconditioned matrix. It requires two additional applications of F^{-1}, and one additional multiplication with C and B^T. However, the greatly improved eigenvalue clustering of (2.4) usually makes the second preconditioner much cheaper in practice, and in general we recommend solving the related system over solving the block-diagonally preconditioned system.

In [27], we also provide bounds that describe the eigenvalue clustering of the proposed preconditioned systems. Eigenvalue clustering is a key factor in the convergence of Krylov subspace methods, although for non-symmetric problems the eigenvector matrix also plays a role. Tight clustering of the eigenvalues generally leads to fast convergence. We now briefly summarize the eigenvalue bounds from [27].

For the related system (2.4), the eigenvalues are clustered around 1, and we have the following bound [27, Theorem 4.2]. Let λ_R be an eigenvalue of the matrix in (2.4). Then

$$|\lambda_R - 1| \le \sqrt{1 + \|N\|_2^2}\sqrt{1 + \|M\|_2^2} \max\left(\|T\|_2, \|\mathcal{E}\|_2\right). \tag{2.5}$$

Eigenvalue bounds for the block-diagonally preconditioned system (2.1) are discussed in detail in [27]. For both preconditioned systems, the eigenvalue clustering depends on two key factors, $\|T\|_2$ and $\|\mathcal{E}\|_2$. As (2.5) shows, both of these must be small in order to assure good clustering and rapid convergence. This means that the splitting of the (1,1) block, F, must be chosen such that $\|T\|_2 = \|F^{-1}E\|_2$ is small, and S_2 must be a good approximation of the Schur complement S_1. There is little to be gained by making one approximation significantly more accurate than the other, and our results in Section 6 will illustrate this.

3. Probing. The key idea behind the probing method in [4] is to efficiently capture large entries in the Schur complement matrices arising from the 1-D interfaces in 2-D non-overlapping domain decomposition methods without explicitly forming the Schur complements. For this isotropic diffusion application, with an appropriate ordering of the unknowns, the entries of the exact Schur complement decay with distance from the diagonal as $O(|i-j|^{-2})$ (in the anisotropic case, things are slightly different, see [17]). Thus, a banded approximation computed by probing approximates the large entries in the Schur complement accurately and yields a good approximation. In fact, if all of the non-zero entries occur within the chosen band width for probing, then probing is exact. If not, probing yields a banded approximation that lumps matrix entries outside the band into the band of the approximate matrix. In such a case, probing for a banded matrix will miss important entries outside the band and pollute entries inside the band with contributions from entries outside the band. For this reason, it is clear that to accurately capture the large entries for matrices with more general decay patterns we need to derive or compute a good pattern to probe for. In this respect, we note that Chan and Mathew [4] did not propose to use banded approximations per se; however, they used probing for a problem where a banded approximation is very appropriate.

The probing method approximates a matrix using only matrix-vector multiplication. We multiply a few carefully constructed vectors by the matrix, and then construct the approximate

ETNA
Kent State University
etna@mcs.kent.edu

matrix from the results, based on an *a priori* chosen sparsity pattern. For example, consider the tridiagonal matrix shown in Figure 3.1. We choose $e_1 + e_4$, $e_2 + e_5$ and e_3 as our probing vectors and we multiply these by the matrix. The resulting product recovers all the non-zero entries in the original matrix exactly. To approximate large entries in more general matrices,

$$\begin{bmatrix} a_1 & b_2 & & & \\ c_1 & a_2 & b_3 & & \\ & c_2 & a_3 & b_4 & \\ & & c_3 & a_4 & b_5 \\ & & & c_4 & a_5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a_1 & b_2 & 0 \\ c_1 & a_2 & b_3 \\ b_4 & c_2 & a_3 \\ a_4 & b_5 & c_3 \\ c_4 & a_5 & 0 \end{bmatrix}$$

FIG. 3.1. *Probing on a tridiagonal matrix using the vectors* $e_1 + e_4$,$e_2 + e_5$ *and* e_3.

we need to extend this approach to more general probing patterns. However, we will show that probing can be applied successfully, so long as the entries of the matrix decay sufficiently rapidly over some underlying graph.

As an example, consider the partial differential equation

$$-\Delta u + .25u = 0 \tag{3.1}$$

on the unit square with homogeneous Dirichlet boundary conditions. We discretize the problem with finite differences, using 15 grid points in each dimension. Figure 3.2 plots a typical column of the inverse of the matrix from (3.1) over the finite difference grid. Let this be the i−th column. We can see that this column has entries j that rapidly decay in magnitude as the distance between nodes i and j on the grid increases. Such a matrix can be approx-

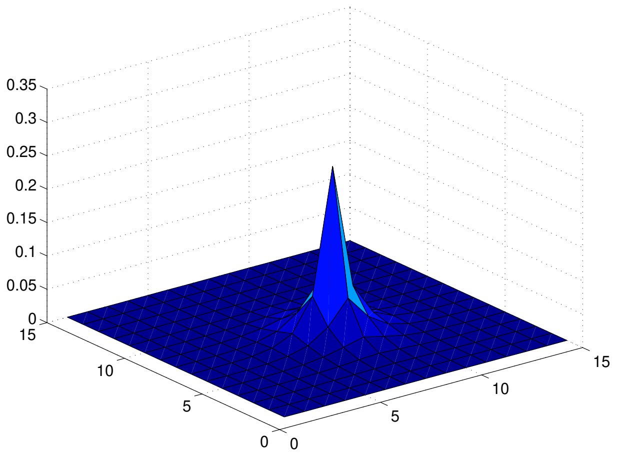


FIG. 3.2. *One column of the inverse of the matrix from (3.1) graphed over the underlying finite difference mesh.*

imated accurately with a probing-based method as long as an appropriate sparsity pattern is chosen and the right probing vectors are chosen. Following the original idea of capturing the large entries, we can extend the probing approach using more general probing patterns to capture more general decay patterns. This is even more important for problems in three spatial dimensions and for systems of partial differential equations. A pictorial example of the proposed extension to probing for matrices with more general decay patterns is shown in

ETNA
Kent State University
etna@mcs.kent.edu

Figure 3.3 (see discussion below). To show that for more general matrices probing for more general sparsity patterns is worth the extra work, we also include an approximation computed by probing for a banded matrix in Figure 3.3. We refer to probing for banded matrices as *banded probing*, and in the discussion below and Section 6 we include some examples of banded probing mainly to motivate the additional work in our more general probing algorithms. For completeness, we note that banded probing should not be confused with probing as suggested by Chan and Mathew [4]. They suggested probing for large entries in Schur complements arising in domain decomposition, and applied this idea to matrices that are well approximated by a banded matrix.

The partial matrix estimation technique [8], which has been used in the estimation of sparse Jacobians and Hessians under various names [6, 7, 15, 16, 20], provides a way for approximating more general matrices. In the context of approximating Schur complements, we shall refer to this method as *structured probing*. In this technique, we first choose a sparsity pattern, $H \in \{0,1\}^{n\times n}$, for the approximate matrix, based on some *a priori* knowledge of the matrix we are approximating (see Section 4). Second, we use graph coloring techniques to compute probing vectors such that a matrix of our chosen sparsity pattern would be reconstructed exactly (see Section 5). Third, we multiply the probing vectors by the matrix. Finally, we use the results of the matrix-vector multiplications to approximate the matrix with the sparsity pattern, H. Algorithm 1 outlines the process for a given input matrix K.

Algorithm 1: $\widetilde{K}$ = Structured Probing$(K \in \mathbb{R}^{n\times n}, H \in \{0,1\}^{n\times n})$

1: Compute a graph G derived from H [See Section 4].
2: Perform a graph coloring on G to generate a mapping $\phi : \{1,\ldots,n\} \to \{1,\ldots,p\}$, where p is the number of colors. The color for vertex i is given by $\phi(i)$ [See Section 5].
3: Generate the matrix of probing vectors $X \in \{0,1\}^{n\times p}$ such that

$$X_{i,j} = \begin{cases} 1 & \text{if } \phi(i) = j, \\ 0 & \text{otherwise.} \end{cases}$$

4: Compute $W = KX$.
5: Build the approximation $\widetilde{K}$,

$$\widetilde{K}_{i,j} = \begin{cases} W_{i,\phi(j)} & \text{if } H_{i,j} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

To illustrate Algorithm 1, we use probing to approximate the inverse of the matrix that arises from discretizing the problem (3.1). Figure 3.3 shows a column of this inverse as well as the corresponding columns in the approximations generated by banded and structured probing. For structured probing, we assume a 13-pt stencil, which for our particular choice of graph coloring, requires 15 vectors. For proper comparison, we also use 15 vectors for banded probing. This serves to illustrate the effect of choosing an appropriate sparsity pattern rather than using a banded pattern for a problem where it is not appropriate. The banded reconstruction will only capture decay in one direction, corresponding to neighbors that are "close" in the node numbering.

4. Choosing the Sparsity Pattern H. Choosing a good sparsity pattern, H, in Step 1 of Algorithm 1, requires some *a priori* knowledge, but not detailed knowledge, about the

ETNA
Kent State University
etna@mcs.kent.edu

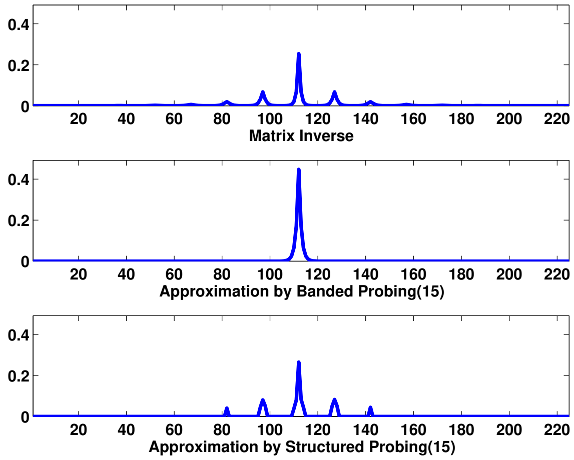


FIG. 3.3. *A typical column of the inverse of the matrix from (3.1) and the approximations generated by banded and structured probing with 15 vectors.*

matrix K. If the "big" entries in K are in a certain pattern, we can choose the *a priori* sparsity pattern accordingly. For many applications these large entries are related to locality on some graph. If the problem is derived from a finite element discretization, the finite element mesh or a subset thereof may provide such a graph. If such a graph is not available immediately from the problem, we can often derive a suitable graph from the matrices involved at some additional cost.

In our case we are interested in matrices $K = (D - CF^{-1}B^T)$, where $F - E$ is the splitting of A in (1.1). If A is the discretization of an elliptic partial differential operator, we can expect the entries of F^{-1} to decay with distance over the graph of F or A. These same principles have been exploited successfully in the computation of sparse approximate inverse preconditioners [5]. If the matrices B, C, and D are also sparse, the entries of K will decay with distance over the graph of $(D - CFB^T)$. Since D is explicitly known, we will focus on the large entries of $CF^{-1}B^T$.

Consider the (i, j) entry of the matrix $CF^{-1}B^T$. For example, assume that C and B^T have four non-zero entries in row i (columns k_1, k_2, k_3, k_4) and column j (rows l_1, l_2, l_3, l_4), respectively, and that none of the entries is large in a relative sense. Let $\bar{k} = \{k_1, k_2, k_3, k_4\}$ and $\bar{l} = \{l_1, l_2, l_3, l_4\}$. Then,

$$(CF^{-1}B^T)_{i,j} = [C_{i,k_1}\ C_{i,k_2}\ C_{i,k_3}\ C_{i,k_4}](F^{-1})_{\bar{k}\times\bar{l}} \begin{bmatrix} (B^T)_{l_1,j} \\ (B^T)_{l_2,j} \\ (B^T)_{l_3,j} \\ (B^T)_{l_4,j} \end{bmatrix}. \tag{4.1}$$

If $\max_{\{k_1,k_2,k_3,k_4\}\times\{l_1,l_2,l_3,l_4\}} |F^{-1}_{k,l}| < \epsilon$, a small threshold value, then $|(CF^{-1}B^T)_{i,j}|$ will also be small. This will occur if i and j are "far" from each other on the graph of CFB^T. In many cases, we have good estimates of the decay of F^{-1}.

When information on the underlying discretization is available, the selection of the sparsity pattern is cheap. When it is not, the sparsity pattern of $-(D - CFB^T)$ or $-(D - CF^kB^T)$, for some k, can provide a sparsity pattern for H. Since we do not need the actual entries of these matrices, in most cases the costs remain modest.

Figure 4.1 shows a staggered finite difference mesh. As D is typically very simple, we consider $D = 0$ for our example in Figure 4.1. The x-variables in (1.1) are represented

ETNA
Kent State University
etna@mcs.kent.edu

by blue x's and the y-variables are represented by blue circles. We consider the y-variable corresponding to node j and the jth column of $CF^{-1}B^T$, and represent it by a black node in Figure 4.1. Black arrows point to the x-variables, l, such that $(B^T)_{l,j} \neq 0$. These are the x-variables "affected" by the y-variable at node j. The neighboring x-variables are labeled $l_1, \ldots, l_4$, as in (4.1). Blue arrows show the effect of the large entries of F^{-1}, pointing to the x-variables k, such that $|(F^{-1}B^T)_{k,j}|$ is large. Here we assume that all entries that are further than one link away on the adjacency graph of F are small. Red arrows show the effect of C, pointing to the y-variables, i, such that $|(CF^{-1}B^T)_{i,j}|$ is large. Note that point i_1 is "far" from j on the graph of CFB^T, and therefore $|(CF^{-1}B^T)_{i_1,j}|$ is small. Point i_2 is "close" to j, and therefore $|(CF^{-1}B^T)_{i_2,j}|$ may be large. In both cases, we have labeled the neighboring x-unknowns $k_1, \ldots, k_4$, as in (4.1). In a situation like this, we can derive appropriate stencils for the Schur complement over the finite difference or finite element mesh as the sparsity pattern for H. This is the approach we take in Section 6.

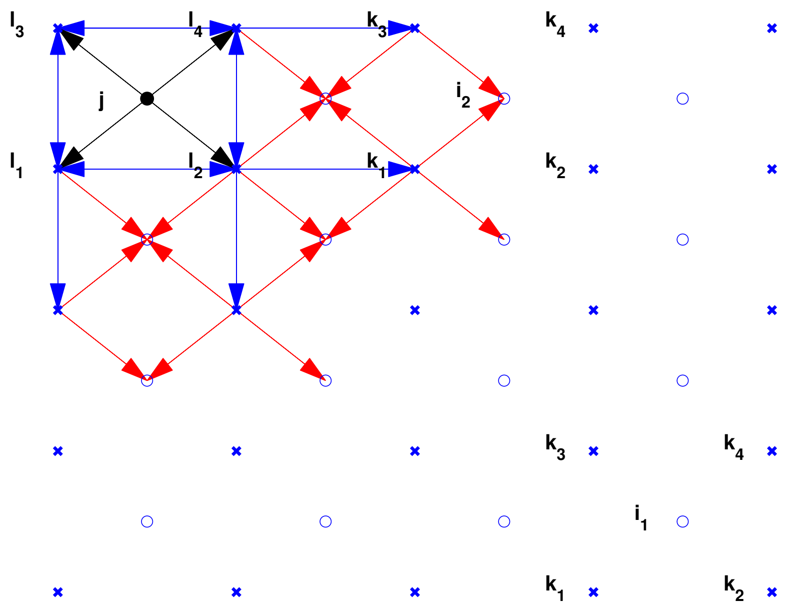


FIG. 4.1. *Entries of the Schur complement affected by node j through B^T (black arrows), $F^{-1}B^T$ (blue arrows), assuming that the entries of F^{-1} decay sufficiently (become small) over the distance of one edge on the adjacency graph and $CF^{-1}B^T$ (red arrows). The value $|(CF^{-1}B^T)_{i_1,j}|$ is guaranteed to be small and $|(CF^{-1}B^T)_{i_2,j}|$ may be large.*

5. Graphs and Colorings. The goal of our graph coloring is to ensure that any single probing vector does not capture data from columns of H with overlapping nonzero patterns. Methods known as substitution methods [8, 16] allow this condition to be relaxed. However, they introduce other complications, and we will not consider them in this paper. With the above condition in mind, we must consider two important issues. The first is which graph can we use to generate vectors that meet this requirement. The second is how should we color this graph.

A graph $G = (V, E)$ is defined by a set of vertices V and a set of edges E connecting vertices. Let $N_1(v_i)$ represent the set of neighbors of vertex v_i, namely $\{v_j \in V \mid (v_i, v_j) \in E\}$. Let $N_2(v_i)$ represent the set of distance-2 neighbors of v_i, namely the set of all $v_j \in V$ such that (1) $v_j \in N_1(v_i)$ or (2) $v_j \in N_1(v_k)$ for some $v_k \in N_1(v_i)$.

ETNA
Kent State University
etna@mcs.kent.edu

We consider approximating the sparse matrix K with at most b non-zero entries per row, using a matrix with the same sparsity as $H \in \{0,1\}^{n\times n}$. We discuss two different graphs in Section 5.1, but they have the same vertex set, $V = \{v_1,\dots,v_n\}$, corresponding to the n unknowns or n rows/columns of the matrix. For each algorithm, let p be the number of colors used.

Graph coloring defines a mapping $\phi : V \to \{1,\dots,p\}$ that assigns a color to each vertex. This mapping is generated in Step 2 of Algorithm 1, and it is subsequently used in the construction of the probing vectors. Various graph coloring algorithms are detailed in Section 5.2.

¿From the perspective of efficiency, we would like these methods to have about the same cost as the necessary matrix-vector multiplications. For a sparse matrix K with at most b non-zero entries per row, the matrix-vector multiplications cost $O(nbp)$ [1]. For all problems, it is known that $p \geq b+1$ [16].

5.1. Choosing a Graph to Color. The first graph we consider is the *adjacency graph* of H, also referred to as "the graph of H", $G_1(H) = (V, E_1)$, where $(v_i, v_j) \in E_1$ if and only if $H_{i,j} = 1$ or $H_{j,i} = 1$ [7, 20]. A distance-1 coloring of G_1 does not guarantee that our condition on the probing vectors holds; however, a distance-2 coloring does. In a distance-2 coloring, $\forall v_k \in N_2(v_i)$, $\phi(v_k) \neq \phi(v_i)$ [20]. If H is stored in a suitable sparse format, generating the graph has negligible cost.

The second graph we consider is the *column intersection* graph, $G_2(H) = (V, E_2)$, where $(v_i, v_j) \in E_2$ if and only if there is a k such that $H_{k,i} = 1$ and $H_{k,j} = 1$ [6]. For this graph, a distance-1 graph coloring satisfies our condition for the probing vectors. This graph has $O(nb^2)$ edges, so it takes at least that much work to construct [15]. If the graphs of both H and H^T are available, one can color the column intersection graph without forming it explicitly [6]. In general, coloring the column intersection graph takes more work than coloring the adjacency graph [16]. Therefore, we will use the adjacency graph for the experiments in Section 6.

5.2. Choosing the Coloring Algorithm. After we choose the graph to color, we must choose an algorithm to compute the coloring. We present three such algorithms. Since we use the adjacency graph of H (see Section 5.1), the algorithms discussed in this section will be distance-2 colorings. The first and simplest choice is the greedy algorithm. Following the description in [16], the greedy algorithm for distance-2 coloring is described in Algorithm 2. If $\bar{\delta}_2$ is the average number of distance-2 neighbors for $v \in V$, then this algorithm takes $O(n\bar{\delta}_2)$ time [16, Lemma 3.9]. The second algorithm, balanced coloring, as described in

Algorithm 2: ϕ = Greedy Distance-2 Coloring$(G = (V, E))$

```
Initialize forbiddenColors with some a ∉ V.
For i = 1, ..., n
    For each colored vertex w ∈ N_2(v_i)
        forbiddenColors[φ(w)] = v_i.
    Set φ(v_i) = minimum c s.t. forbiddenColors[c] ≠ v_i.
```

[8], may yield a more efficient coloring. In this approach, we balance the number of nodes assigned to each color. Our version of this heuristic algorithm assigns a color to each node sequentially. It builds a list of forbidden colors for each node, and then chooses the valid

[1] In practice, of course, K is only available implicitly through matrix-vector products. The cost then depends on how the required data is available and the implementation of the matrix-vector products.

ETNA
Kent State University
etna@mcs.kent.edu

color that has been used to color the smallest number of nodes. Since we need at least as many colors as the highest degree of any node plus one [15], we start the algorithm with that many colors. We increase the number of colors when a node cannot be assigned any of the existing colors. Algorithm 3 outlines this method for computing a balanced distance-2 coloring of a graph. Computing the forbidden colors list for each node takes $O(b^2)$ time,

Algorithm 3: $\phi =$ Balanced Distance-2 Coloring$(G = (V, E))$

Set $p = b + 1$, the initial number of colors to consider.
Set $c_j = 0$, for $j = 1, \ldots, p$, be the number of times each color has been used.
Initialize forbiddenColors with some $a \notin V$.
For $i = 1, \ldots, n$
    For each colored vertex $w \in N_2(v_i)$
        forbiddenColors$[\phi(w)] = v_i$.
    Set $colored =$ false and $nuses = \infty$.
    For $j = 1, \ldots, p$
        if forbiddenColors$[j] \neq v_i$ and $c_j < nuses$ then
            Set $nuses = c_j$, $colored =$ true and $\phi(v_i) = j$.
    if $colored =$ false then
        Set $\phi(v_i) = p + 1$, $c_{p+1} = 0$.
        $p = p + 1$.
    Set $c_{\phi(v_i)} = c_{\phi(v_i)} + 1$.

and choosing the appropriate color takes $O(p)$ time. Therefore, the overall time complexity of doing a balanced distance-2 coloring is $O(n(b^2 + p))$.

The third algorithm we propose uses probing vectors with the same $\{0, 1\}$ patterns as banded probing [4], but picks the number of vectors such that we can exactly reconstruct the desired sparsity pattern. We choose the number of colors p such that p is relatively prime to all elements of the set $\{\ i - j \mid \exists k :\ \ H_{k,i} = 1 \text{ and } H_{k,j} = 1\}$. We refer to this method as the *prime divisor coloring*. This algorithm implicitly performs a distance-2 coloring on the adjacency graph of H, although we do not use the graph explicitly. Instead, we operate directly on H. The algorithm is given as Algorithm 4. The great advantage of this approach

Algorithm 4: $\phi =$ Prime Divisor Distance-2 Coloring$(H \in \{0, 1\}^{n,n})$

Initialize illegalP with the empty set.
For $i = 1, \ldots, n$
    For j s.t. $H_{i,j} \neq 0$
        For k s.t. $H_{i,k} \neq 0$ and $k > j$
            illegalP $\leftarrow (k - j)$
For $i = 2, \ldots, n$
    if $\forall j \in$ illegalP, $j \bmod i \neq 0$ then
        $p = i$
        break
For $i = 1, \ldots, n$
    Set $\phi(v_i) = i \bmod p$.

is that if H comes from a fixed stencil on a regular grid, we need only consider a single "representative" row of the matrix (i.e., a row for a point away from the boundaries), and use that row to choose the number of colors p. For such problems, this algorithm takes $O(p+b^2)$ work. However, this algorithm takes $O(n(p+b^2))$ work for unstructured meshes. It can be accelerated somewhat by only using prime numbers in Step 5 of Algorithm 4, but that does not improve the asymptotic efficiency.

Finally, as noted in [6], graph coloring heuristics are sensitive to the ordering of the nodes. A good ordering reduces the number of colors, and thus makes the underlying probing process computationally less expensive. One strategy is to order the nodes so that all of the high-degree vertices are numbered first. This is referred to as the largest-first ordering (LFO) in [6]. More complicated orderings, where nodes are colored based on the topology of particular subgraphs, are discussed in [6].

6. Results. We consider two applications for our numerical experiments. The first application models a leaky lid-driven cavity using the Navier-Stokes equations. For this application we use the MATLAB software of [12]. This particular problem has $A \neq A^T$, $B = C$ and $D \neq 0$ in the notation of (1.1). To analyze the eigenvalue clustering and convergence of the preconditioned systems with various choices for the approximate Schur complement, we use a 16×16 grid with viscosity parameter $\nu = 0.1$ and stabilization parameter $\beta = 0.25$. After removing the constant pressure mode, this system has 705 unknowns. For the splitting of the (1,1) block, $A = F - E$, we use one multi-grid V-cycle with three SOR-Jacobi pre- and post-smoothing steps and relaxation parameter $\omega = 0.25$. We also consider the effects of h-refinement on the preconditioned systems. We consider $N \times N$ grids with $N = 16, 32, 64, 128$. For all of these problems we order velocities in x-direction first, then velocities in y-direction, and then the pressures; within each group of unknowns we use a standard lexicographic ordering.

In addition to demonstrating the effectiveness of our preconditioners with approximate Schur complements generated by probing, we use the Navier-Stokes problem to illustrate the importance of using general probing and reconstruction techniques that capture the decay pattern for matrices with complex decay patterns. Specifically, we compare structured probing with banded probing as a cheap and simple alternative to focus on the role of the sparsity pattern chosen for the approximate Schur complement matrix. We use the prime divisor method for graph coloring to isolate the role of this chosen sparsity pattern. In fact, this method allows us to use the same probing vectors (X) for structured probing and for banded probing; so, the only difference between the two methods is in the sparsity pattern that we use for the construction of the approximate matrix. The resulting experiments demonstrate that even using the same vectors as banded probing, reconstruction based on a more appropriate sparsity pattern leads to much better eigenvalue clustering and convergence. This provides a justification for the computational expense of the graph coloring. Furthermore, we examine the use of incomplete factorizations (ILU(0)) for the approximate Schur complement matrices generated by structured probing as a means of further reducing the cost of the preconditioners (2.1) and (2.4).

The second application uses the modified Hart's model [14] to model elastic, plastic, anelastic, micro-plastic and micro-anelastic strain and their effects on the permanent deformation of bent beams [29]. We first consider a small linear system with 6422 unknowns that arises from a Newton iteration to solve the nonlinear problem at each timestep. For this problem, we study the following structural splittings of the (1,1) block A: the diagonal of A, a banded splitting of A with a semi-bandwidth of four, and the ILU(0) factorization of A. In addition, we compare the convergence using an exact Schur complement and using approximations from structured probing. We also use ILU(0) to factor the approximate Schur complements generated by structured probing. We provide timing results for the iterations with approximate Schur complements. The structured probing approximations were obtained

ETNA
Kent State University
etna@mcs.kent.edu

with the distance-2 balanced coloring algorithm of [8]. For this problem we focus on GMRES convergence and wall-clock time. Finally, we consider the effects of h-refinement on the convergence of the related system (2.4) using various algorithmic choices in computing the structured probing approximations to the Schur complement. For all of these problems we use a minimum bandwidth ordering.

Our experiments are run using our structured probing software library [26]. All the graphs and coloring methods discussed above are included in the software package. A more extensive C++ library (focusing on sparse Jacobians and Hessians) is under development by Alex Pothen and his collaborators. We use GMRES as the iterative solver and we iterate until we obtain a relative residual tolerance of $1\mathrm{e}-10$.

6.1. Eigenvalue Clustering and Convergence for the Navier-Stokes Problem. We first analyze the eigenvalue clustering and convergence for a small Navier-Stokes problem. Figures 6.1(a) and 6.1(b) show the eigenvalue distributions for the related system (2.4) with banded probing and with structured probing using 13 probing vectors. For scaling purposes, we exclude two negative eigenvalues for the banded probing case at approximately $(-73, 0)$ and $(-113, 0)$. We use a nine-point stencil on the element connectivity graph to define the sparsity pattern H for structured probing (SP). The prime divisor method described in Section 5, yields a graph coloring requiring 13 vectors. The vectors used for structured probing and banded probing are the same. The only difference between the two methods is in the construction of the approximate Schur complement matrix.

Structured probing yields much better clustering than banded probing, especially near the origin. Structured probing has only one small eigenvalue (about 0.01); the others are well separated from zero. Banded probing has many eigenvalues clustering near the origin. This leads to worse convergence behavior; see Figure 6.3(a).

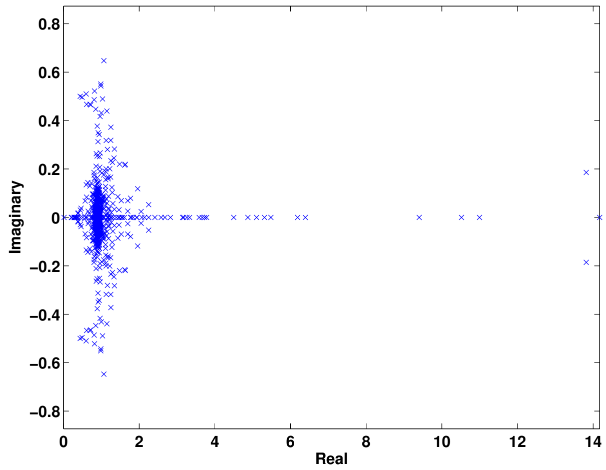
We see similar results for the eigenvalues of the block-diagonally preconditioned system (2.1) with banded and structured probing; see Figures 6.2(a) and 6.2(b). Both banded probing and structured probing lead to one small eigenvalue (about 0.01), but structured probing yields eigenvalue clusters much further away from the origin.

Figures 6.3(a) and 6.3(b) show the convergence of GMRES for the preconditioned systems. The difference between banded and structured probing is quite pronounced. For both preconditioned systems, structured probing using a five-point stencil with seven vectors results in a lower iteration count than banded probing even with thirteen vectors. We also note that using the related system (2.4) leads to significantly faster convergence than using the block-diagonally preconditioned system (2.1) for all probing variants.

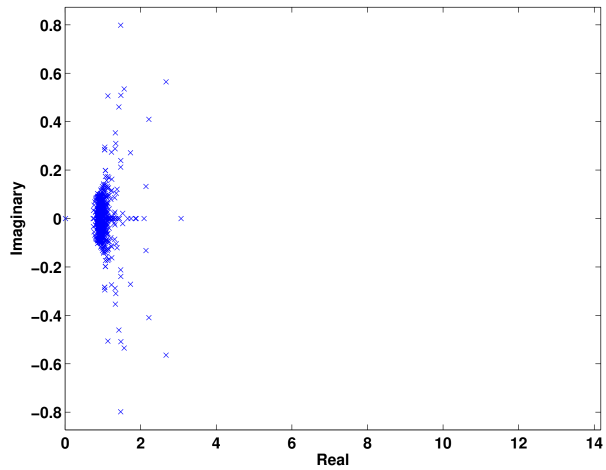
Figures 6.4(a) and 6.4(b) show the eigenvalues for the related system and block-diagonally preconditioned system for structured probing with both five-point (seven vector) and nine-point (thirteen vector) stencils. Note that, barring a few outliers, for both types of preconditioners the eigenvalue clustering is significantly better for the nine-point stencil, especially near the origin. Krylov-subspace methods tend to find and “remove” outlying eigenvalues quickly. Therefore, these eigenvalues do not affect the convergence rate after a small number of initial iterations. Thus, the significantly better eigenvalue clustering obtained using the nine-point stencil leads to a significantly improved convergence rate for GMRES.

6.2. ILU(0) for the Approximate Schur Complement from Structured Probing. The benefits of structured probing come at the cost of having an approximate Schur complement that is more expensive to factor. For instance, if the chosen sparsity pattern looks similar to a d-dimensional Laplacian (if A and hence F are related to a Laplacian), the large bandwidth can yield significant fill-in, making the exact factorization of the approximate Schur complement matrix expensive to compute and apply. Therefore, we use an inexact factorization

ETNA
Kent State University
etna@mcs.kent.edu



(a) Banded Probing (13 vectors)



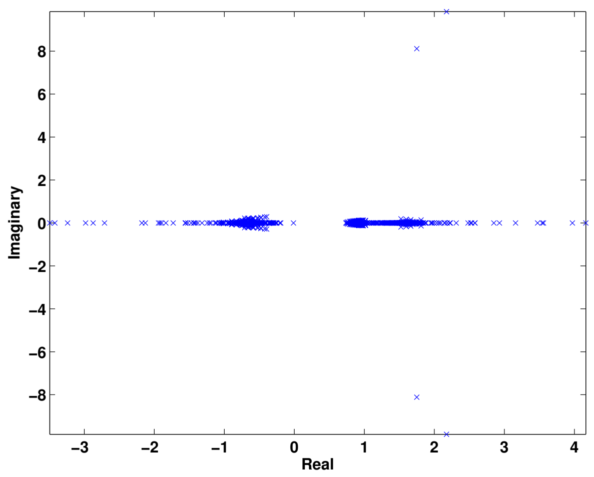
(b) Structured Probing (9pt. stencil / 13 vectors)

FIG. 6.1. *Eigenvalues for the related system (2.4) derived from the Navier-Stokes problem with one V-cycle as splitting of the (1,1) block and with approximate Schur complements from banded and structured probing with exact factorizations.*

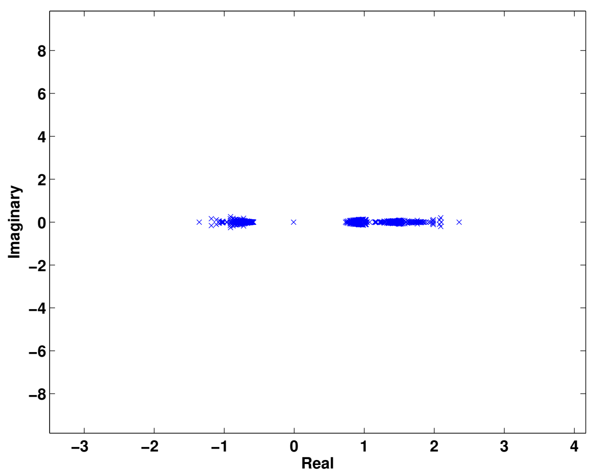
of the approximate Schur complement to define S_2^{-1}. In practice, this leads to a negligible deterioration in convergence while reducing the overhead of applying structured probing significantly. We use an ILU(0) factorization for this problem. For symmetric problems an IC(0) factorization should be used. Since ILU(0) and IC(0) have linear cost in the number of unknowns, the overall cost remains $O(m)$.

Figure 6.5 shows the eigenvalue distributions for both types of preconditioned systems with structured probing. We use a nine point stencil (13 vectors) and the exact as well as the ILU(0) factorization of the approximate Schur complement. Figure 6.6 shows the con-

ETNA
Kent State University
etna@mcs.kent.edu



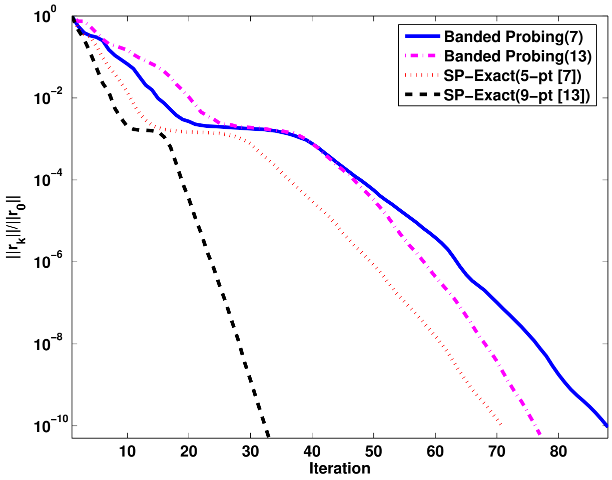
(a) Banded Probing (13 vectors)



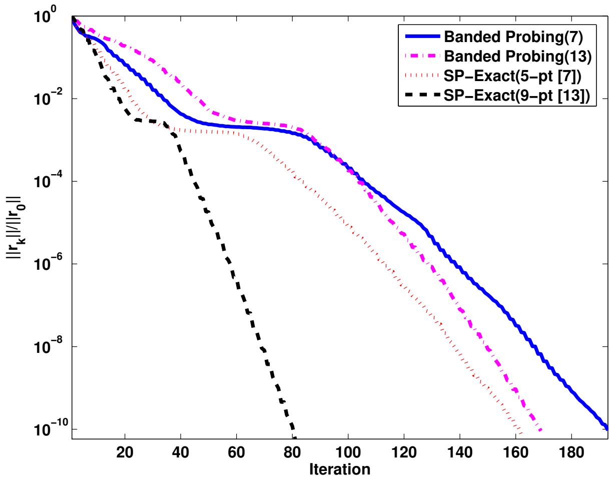
(b) Structured Probing (9pt. stencil / 13 vectors)

FIG. 6.2. *Eigenvalues for the block-diagonally preconditioned system (2.1) derived from the Navier-Stokes problem with one V-cycle as splitting of the (1,1) block and approximate Schur complements using banded and structured probing with exact factorizations.*

vergence results for both types of preconditioned systems, using structured probing with 9 and with 13 vectors. Using ILU(0) instead of an exact factorization changes the eigenvalue distribution slightly, but leaves the clustering essentially equivalent. The impact of such a change on the convergence behavior is negligible. Given the significant difference in cost between exact and inexact factorizations, using ILU(0) is more cost-effective than an exact factorization.



(a) Related system (2.4)

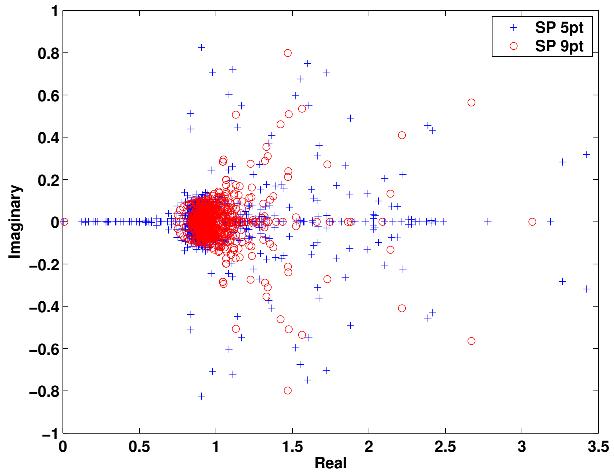


(b) Block-diagonally preconditioned system (2.1)

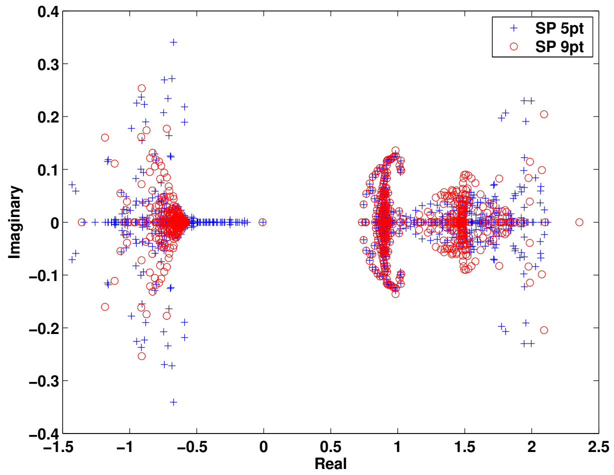
FIG. 6.3. *GMRES convergence for the Navier-Stokes problem with one V-cycle as splitting of the (1,1) block and approximate Schur complements using banded and structured probing with exact factorizations.*

6.3. Probing and h-refinement for the Navier-Stokes Problem. We consider the effect of h-refinement on the convergence of the related system (2.4). We vary the number of elements from 16×16 to 128×128, with the largest system having $48,641$ unknowns. As before, we use a single multigrid V-cycle as the splitting of the (1,1) block of (1.1). We consider structured probing using 5, 9 and 13 point stencils to define the sparsity pattern, H, for the approximate Schur complement. We compare the number of (GMRES) iterations with those for banded probing using the same number of vectors as used for structured probing. Table 6.1 shows the convergence results in terms of GMRES iterations.

ETNA
Kent State University
etna@mcs.kent.edu



(a) Related system (2.4)

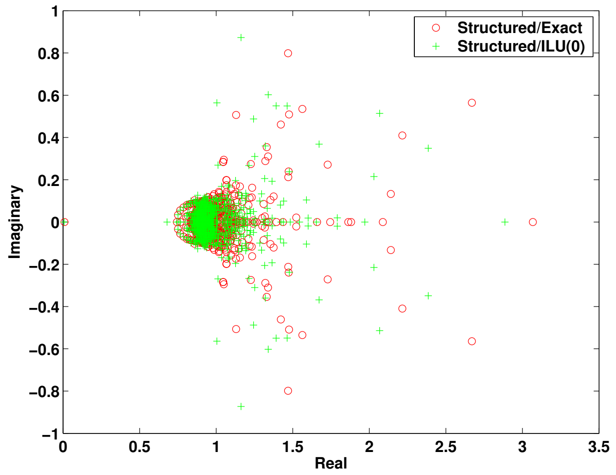


(b) Block-diagonally preconditioned system (2.1)

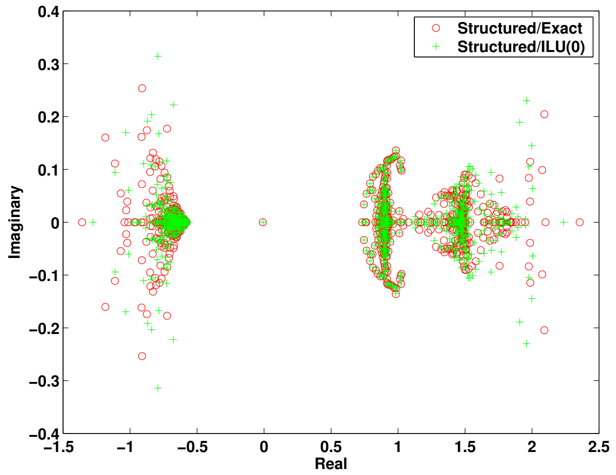
FIG. 6.4. *Eigenvalues for the related system (2.4) and the block-diagonally preconditioned system (2.1) derived from the Navier-Stokes problem with one V-cycle as splitting of the (1,1) block and approximate Schur complements using structured probing with exact factorizations.*

We see that the related system (2.4) with structured probing has mild dependence on $h = 1/N$. This is true for each of the sparsity patterns we consider; however, note that the increase in GMRES iterations slowly declines with increasing N. Probing for sparsity patterns, H, in the Schur complement derived from discretization stencils (structured probing) yields faster convergence than probing for banded patterns (banded probing). This is explained by the fact that the decay of F^{-1} has a two-dimensional structure. Our results show that the original ideas behind probing can be extended effectively to more general patterns.

ETNA
Kent State University
etna@mcs.kent.edu



(a) Related system (2.4)

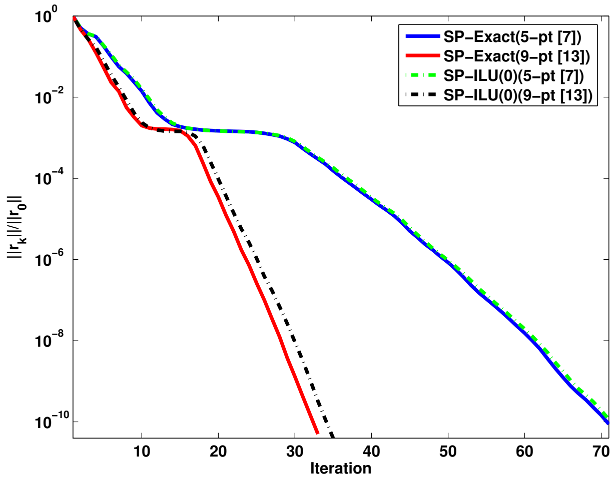


(b) Block-diagonally preconditioned system (2.1)

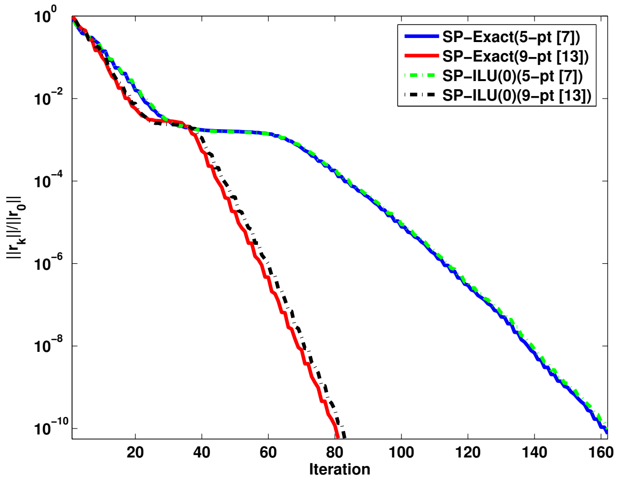
FIG. 6.5. *Eigenvalues for the related system (2.4) and the block-diagonally preconditioned system (2.1) derived from the Navier-Stokes problem with one V-cycle as splitting of the (1,1) block and approximate Schur complements using structured probing (13 vectors) with exact and ILU(0) factorizations.*

Even though the test problems have only modest convection, the banded Schur complement approximations are progressively worse conditioned, culminating in some numerically singular approximations for the largest problems. The approximations based on a 2-D stencil are significantly better conditioned and lead to rapid GMRES convergence.

6.4. Computational Results for Metal Deformation. The metal deformation problem arises from a finite element mesh, where the second set of variables corresponds to nodes in the center of the elements. For structured probing, we approximate the Schur complement



(a) Related system (2.4)



(b) Block-diagonally preconditioned system (2.1)

FIG. 6.6. *GMRES convergence for the Navier-Stokes problem with one V-cycle as the splitting of the (1,1) block and with an inexact Schur complement based on structured probing, using both exact and ILU(0) factorizations.*

with a matrix that has the sparsity pattern, H, of the element-element connectivity graph of the original problem. Using a distance-2 balanced coloring on the adjacency graph of H, we can build our approximation using only nine probing vectors.

We present GMRES convergence results and wall clock time for a single linear system from the metal deformation problem in Figures 6.7(a) and 6.7(b), respectively. For the convergence results in Figure 6.7(a), we use an ILU(0) splitting of the (1,1) block, A, and we compare the convergence for the exact Schur complement, and an approximate Schur complement using structured probing with both exact and ILU(0) factorizations. Figure 6.7(a)

TABLE 6.1
Number of GMRES iterations for the related system (2.4) using multigrid V-Cyles as a splitting of the (1,1) block and structured probing with an ILU(0) factorization or banded probing to approximate the Schur complement, for various levels of h-refinement in lid-driven cavity problem modelled with Navier-Stokes. Asterisks ($*$) indicate very ill-conditioned preconditioners.

Splitting	N	Structured Probing H Stencil 5-pt	9-pt	13-pt	Banded Probing w/nvecs equal to 5-pt	9-pt	13-pt
1 V-Cycle	16	75	36	32	107	71	107
	32	103	55	49	140	111	272
	64	122	79	71	542	169	542
	128	134	96	90	$*$	1307	$*$

shows that for this problem, probing-based approximations to the Schur complement lead to effective preconditioners. In addition, using an incomplete decomposition of the approximate Schur complement (by structured probing) does not lead to a deterioration of convergence. Figure 6.7(b) shows that use of an inexact factorization with structured probing saves approximately 5% of execution time.

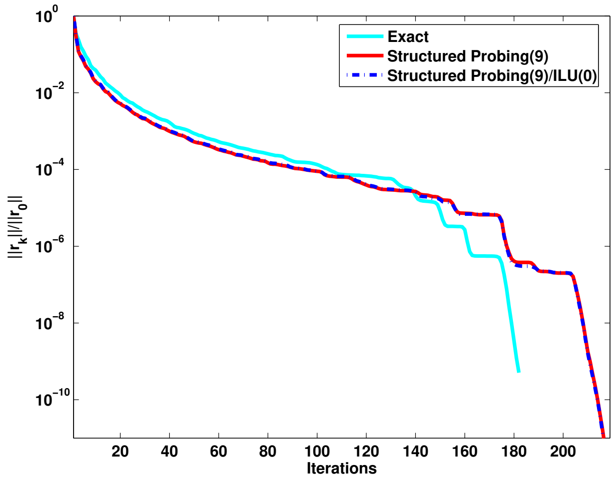
Next, we consider how our preconditioners perform with respect to h-refinement. We refine the metal deformation problem by multiplying the number of variables per dimension by 2, 3 and 4, which yields systems up to 273,258 unknowns. For this study, we consider three splittings for the (1,1) block, the ILU(0) splitting previously discussed, smoothed-aggregation algebraic multigrid (AMG) using ML [25] and the exact splitting $F = A$. For ML, we do two V-cycles with a single step of SOR-Jacobi as pre- and post-smoother and weighting parameter $\omega = 0.5$. We also set the aggregation threshold to 0.1, meaning that entries are dropped in the coarsening phase if $|A_{i,j}| \leq 0.1\sqrt{|A_{i,i}A_{j,j}|}$. All other parameters, including the uncoupled coarsening technique, are set to the ML defaults. For each of these problems, we use structured probing with the prime divisor, balanced and greedy colorings and an ILU(0) factorization of the approximate Schur complement.

Table 6.2 shows the number of GMRES iterations necessary to solve the related system (2.4) to a tolerance of $1\mathrm{e}-10$. Table 6.3 shows the corresponding timing results. The timing results represent a minimum over five runs on a machine with a 2.6GHz Intel Xeon CPU and 2 GB of RAM running version 2.6.8 of the Linux kernel.

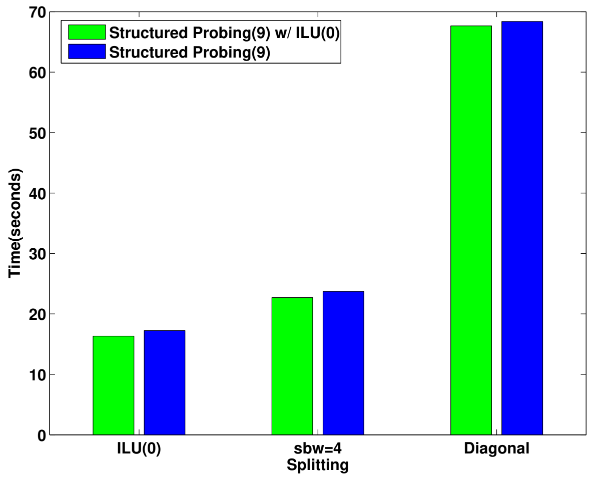
It is well-known that elastic-plastic problems are difficult to precondition. Therefore, it is not surprising that ILU(0) and AMG are not particularly effective splittings for this problem. As stated in (2.5), the effectiveness of the overall preconditioner is limited by the accuracy of both the splitting and the approximate Schur complement. If the splitting is poor, a better approximation to the Schur complement is unlikely to yield a significant improvement in convergence. We see this effect in Table 6.2. The difference between the prime divisor coloring, which uses more vectors than the balanced or greedy colorings for a given stencil, is most pronounced when we use the exact splitting, $F = A$.

7. Conclusions and Future Work. We have shown that probing provides an effective technique for approximating the Schur complement matrices that arise in saddle point preconditioners. The probing technique proposed by Chan and Mathew [4] for applied to 2-D domain decomposition problems can be adapted to reconstruct exactly matrices of arbitrary sparsity structure or to approximate matrices that have a suitable decay property relative to a chosen sparsity structure. This makes probing a very powerful technique in preconditioning saddle-point problems, if a good estimate of the probing pattern can be made *a priori*. The results presented in this paper show the effectiveness of using structured probing with these

ETNA
Kent State University
etna@mcs.kent.edu



(a) GMRES Convergence for ILU(0) Splitting



(b) Wall Clock Time

FIG. 6.7. *GMRES convergence and wall clock time for a single non-linear iteration of the metal deformation problem for various probing-based inexact Schur complements in the related system (2.4) and for three different splittings of the (1,1) block (ILU(0), banded matrix with semibandwidth 4, and diagonal).*

preconditioners in terms of eigenvalue clustering, rate of convergence and execution time.

As future work, we seek to develop estimates for $||\mathcal{E}||_2$ for structured probing methods. We also seek to dynamically adapt the *a priori* chosen sparsity structure for structured probing. Furthermore, we plan to identify additional problems that yield decay properties that can be exploited by structured probing, especially based on more algebraic criteria. We will also work on developing an accurate eigenvalue analysis for systems preconditioned using structured probing. Finally, we intend to look at schemes for updating and reusing (generalized)

TABLE 6.2

Number of GMRES iterations for the related system (2.4) using an ILU(0) or AMG splitting of the (1,1) block and structured probing to approximate the Schur complement, for various levels of h-refinement on a single non-linear iteration in the metal deformation problem. Dashes indicate insufficient memory to run that particular combination.

Splitting	# Unknowns	Structured Probing Coloring		
		Prime Divisor	Balanced	Greedy
Exact	6422	18	32	25
	39722	23	38	31
	121466	25	34	35
	273258	—	—	—
ILU(0)	6422	219	230	225
	39722	432	477	478
	121466	644	700	730
	273258	—	—	—
AMG	6422	153	159	159
	39722	215	227	227
	121466	223	241	249
	273258	266	285	307

TABLE 6.3

Wall-clock time (seconds) for the related system (2.4) using an ILU(0) or AMG splitting of the (1,1) block and structured probing to approximate the Schur complement, for various levels of h-refinement on a single non-linear iteration in the metal deformation problem. Dashes indicate insufficient memory to run that particular combination.

Splitting	# Unknowns	Structured Probing Coloring		
		Prime Divisor	Balanced	Greedy
Exact	6422	7.79e-01	1.07e+00	8.91e-01
	39722	1.32e+01	1.68e+01	1.46e+01
	121466	9.91e+01	1.03e+02	1.04e+02
	273258	—	—	—
ILU(0)	6422	4.09e+00	4.31e+00	4.22e+00
	39722	1.17e+02	1.40e+02	1.38e+02
	121466	1.44e+03	9.89e+02	1.08e+03
	273258	—	—	—
AMG	6422	8.87e+00	9.10e+00	9.06e+00
	39722	9.80e+01	1.03e+02	1.03e+02
	121466	3.80e+02	4.09e+02	4.25e+02
	273258	8.20e+02	9.12e+02	1.01e+03

saddle point preconditioners with probing-based inexact Schur complements.

Acknowledgments. We thank Howard Elman, Alison Ramage, David Sylvester and Andrew Wathen for making the MATLAB implementation of the 2-D Navier-Stokes lid-driven cavity problem available [12].

REFERENCES

[1] C. BERNARDI, C. CANUTO, AND Y. MADAY, *Generalized inf-sup conditions for Chebyshev spectral approximation of the Stokes problem*, SIAM J. on Numer. Anal., 25 (1988), pp. 1237–1271.

[2] D. BRAESS, *Finite Elements: Theory, fast solvers and applications in solid mechanics*, Cambridge University Press, 2nd ed., 2001.

[3] J. BRAMBLE AND J. PASCIAK, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17.

[4] T. CHAN AND T. MATHEW, *The interface probing technique in domain decomposition*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 212–238.

[5] E. CHOW, *A priori sparsity patterns for parallel sparse approximate inverse preconditioners*, SIAM J. Sci. Comput., 21 (2000), pp. 1804–1822.

[6] T. COLEMAN AND J. MORÉ, *Estimation of sparse Jacobian matrices and graph coloring problems*, SIAM J. Numer. Anal., 20 (1983), pp. 287–309.

[7] ———, *Estimation of sparse Hessian matrices and graph coloring problems*, Math. Program., 28 (1984), pp. 243–270.

[8] J. CULLUM AND M. TŮMA, *Matrix-free preconditioning using partial matrix estimation*, Tech. Rep. 898, Institute of Computer Science, Academy of Sciences of the Czech Republic, April 2004.

[9] E. DE STURLER AND J. LIESEN, *Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. Part I: Theory*, SIAM J. Sci. Comput., 26 (2005), pp. 1598–1619.

[10] H. ELMAN, *Preconditioning for the steady-state Navier-Stokes equations with low viscosity*, SIAM J. Sci. Comput., 20 (1999), pp. 1299–1316.

[11] H. ELMAN AND D. SILVESTER, *Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations*, SIAM J. Sci. Comput., 17 (1996), pp. 33–46.

[12] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Iterative methods for problems in computational fluid dynamics*, in Winter School on Iterative Methods in Scientific Computing and Applications, Chinese University of Hong Kong, 1996.

[13] ———, *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations*, Numer. Math., 90 (2002), pp. 665–688.

[14] H. GARMESTANI, M. VAGHAR, AND E. HART, *A unified model for inelastic deformation of polycrystalline materials — application to transient behavior in cyclic loading and relaxation*, Internat. J. Plasticity, 17 (2001), pp. 1367–1391.

[15] A. GEBREMEDHIN, F. MANNE, AND A. POTHEN, *Graph coloring in optimization revisited*, Tech. Rep. 226, Department of Informatics, University of Bergen, January 2002.

[16] ———, *What color is your Jacobian? Graph coloring for computing derivatives*, SIAM Rev., 47 (2005), pp. 629–705.

[17] L. GIRAUD AND R. TUMINARO, *Schur complement preconditioners for anisotropic problems*, IMA J. Numer. Anal., 19 (1999), pp. 1–17.

[18] I. IPSEN, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[19] D. LOGHIN AND A. WATHEN, *Schur complement preconditioners for elliptic systems of partial differential equations*, Numer. Linear Algebra Appl., 10 (2003), pp. 423–443.

[20] S. MCCORMICK, *Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem*, Math. Program., 26 (1983), pp. 153–171.

[21] M. MURPHY, G. GOLUB, AND A. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 2969–2972.

[22] R. NICOLAIDES, *Existence, uniqueness and approximation for generalized saddle point problems*, SIAM J. Numer. Anal., 19 (1982), pp. 349–357.

[23] I. PERUGIA AND V. SIMONCINI, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.

[24] A. QUARTERONI AND A. VALLI, *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, 2nd ed., 1997.

[25] M. SALA, M. GEE, J. HU, AND R. TUMINARO, *ML 4.0 Smoothed Aggregation User's Guide*, Tech. Rep. SAND2005-4819, Sandia National Laboratories, Albuquerque, NM USA, 2005.

[26] C. SIEFERT, *Structured Probing Toolkit*, 2005. http://www.cse.uiuc.edu/˜siefert/structured_probing/.

[27] C. SIEFERT AND E. DE STURLER, *Preconditioners for generalized saddle-point problems*, Tech. Rep. UIUCDCS-R-2004-2448, Department of Computer Science, University of Illinois at Urbana-Champaign, June 2004. Accepted for publication in SIAM J. Numer. Anal.

[28] J. TANNEHILL, D. ANDERSON, AND R. PLETCHER, *Computational Fluid Mechanics and Heat Transfer*, Taylor & Francis, Philadelphia, 2nd ed., 1997.

[29] L. ZHU, A. BEAUDOIN, AND S. MACEWAN, *A study of kinetics in stress relaxation of AA 5182*, in Proceedings of TMS Fall 2001: Microstructural Modeling and Prediction During Thermomechanical Processing, 2001, pp. 189–199.