SIAM J. SCI. STAT. COMPUT.
Vol. 6, No. 1, January 1985

018

BLOCK PRECONDITIONING FOR THE CONJUGATE GRADIENT METHOD*

P. CONCUS†, G. H. GOLUB‡ AND G. MEURANT§

Abstract. Block preconditionings for the conjugate gradient method are investigated for solving positive definite block tridiagonal systems of linear equations arising from discretization of boundary value problems for elliptic partial differential equations. The preconditionings rest on the use of sparse approximate matrix inverses to generate incomplete block Cholesky factorizations. Carrying out of the factorizations can be guaranteed under suitable conditions. Numerical experiments on test problems for two dimensions indicate that a particularly attractive preconditioning, which uses special properties of tridiagonal matrix inverses, can be computationally more efficient for the same computer storage than other preconditionings, including the popular point incomplete Cholesky factorization.

Key words. conjugate gradient method, elliptic partial differential equations, incomplete factorization, iterative methods, preconditioning, sparse matrices

1. Introduction. In this paper we study some preconditioning techniques for the conjugate gradient method to solve the linear systems of equations that arise from the discretization of partial differential equations. We consider for example elliptic equations such as

$$-\sum_{i=1}^{d} \frac{\partial}{\partial \xi_i}\left(\lambda_i(\xi)\frac{\partial u}{\partial \xi_i}\right) + \sigma(\xi)u = f \quad \text{in } \Omega \subset R^d,\ \xi = (\xi_1, \xi_2, \ldots, \xi_d) \tag{1}$$

with

$$u(\xi) = g(\xi) \quad \text{or} \quad \frac{\partial u}{\partial n} = g(\xi) \qquad \text{on } \partial\Omega,$$

where n is the exterior normal, $\lambda_i(\xi) > 0$, and $\sigma(\xi) \geqslant 0$. The techniques that we describe are suitable for standard finite-difference discretizations of equations such as the above that yield certain symmetric positive definite block tridiagonal linear systems of the form

$$Ax = b, \tag{2}$$

where

$$A = \begin{pmatrix} D_1 & A_2^T & & & \\ A_2 & D_2 & A_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & D_{n-1} & A_n^T \\ & & & A_n & D_n \end{pmatrix}.$$

*Received by the editors May 2, 1983, and in revised form January 16, 1984. This paper is an abridged version of [2], which was presented at the SIAM 30th Anniversary Meeting, Stanford University, Stanford, California, 1982. It was typeset at the Lawrence Berkeley Laboratory using a *troff* program running under UNIX. The final copy was produced on July 6, 1984. This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under contracts DE-AC03-76SF00098 and DE-AC03-76SF00515 and by the National Science Foundation under grant MCS-78-11985.

†Lawrence Berkeley Laboratory and Department of Mathematics, University of California, Berkeley, California 94720.

‡Computer Science Department, Stanford University, Stanford, California 94305.

§Commissariat à l'Energie Atomique, Limeil 94190 Villeneuve-Saint-Georges, France, and Computer Science Department, Stanford University, Stanford, California 94305.

Such equations can arise also from finite element discretizations (for example, see [11]).

The prototype model problem in two dimensions is the Dirichlet problem, $\sigma \equiv 0$, $\lambda_i \equiv 1$, $g \equiv 0$, Ω the unit square,

$$-\Delta u = f,$$

$$u = 0 \quad \text{on the boundary,}$$

with standard five-point differencing on a uniform mesh of width h. We focus attention on the matrix structure obtained for natural ordering, which yields for the model problem (after multiplication by h^2)

$$A_i = -I, \quad D_i = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

In three dimensions, standard 7-point differencing with this ordering would yield D_i that have two additional nonzero diagonals. Different orderings or higher order approximations would give rise to different structures, to which our techniques could be applied also.

To solve (2) we use the generalized or preconditioned conjugate gradient method, which may be written as follows [3]. Let x^0 be given, define p^{-1} arbitrarily, and let $r^0 = b - Ax^0$. For $k = 0, 1, \ldots$ perform the steps

$$Mz^k = r^k,$$

$$\beta_k = \frac{(z^k, Mz^k)}{(z^{k-1}, Mz^{k-1})}, \quad k \geqslant 1, \qquad \beta_0 = 0,$$

$$p^k = z^k + \beta_k p^{k-1},$$

$$\alpha_k = \frac{(z^k, Mz^k)}{(p^k, Ap^k)},$$

$$x^{k+1} = x^k + \alpha_k p^k,$$

$$r^{k+1} = r^k - \alpha_k A p^k.$$

The matrix M is the preconditioning matrix, which should be in some sense an approximation of A. It is known that the preconditioned conjugate gradient method converges rapidly if the condition number $\kappa(M^{-1}A)$, which is the ratio of the largest to the smallest eigenvalue of $M^{-1}A$, is small or if the eigenvalues are clustered (e.g., [3] and the references therein).

The goal of this study is to devise good preconditioning matrices M. For this purpose we exploit the structure of A in constructing some block preconditionings, one special case of which is the one introduced by R. R. Underwood [20].

In §2 to motivate the use of our block techniques we recall some results on block Cholesky factorization. Section 3 deals with the main problem—finding good approximate inverses for tridiagonal matrices that are diagonally dominant.

New block techniques for two-dimensional problems ($d = 2$ in (1)) are introduced in §4. Three-dimensional problems will be discussed in detail in a subsequent study.

In §5 we present the results of numerical experiments for several test problems, including comparisons with point preconditioning techniques. We compare techniques on the basis of number of iterations and number of floating point operations required for convergence. Also, we illustrate graphically the spectral properties of the matrices corresponding to the various preconditionings.

2. Block Cholesky factorization. Let A be the symmetric positive definite block tridiagonal matrix of (2). Let m_i be the order of the i^{th} square diagonal block D_i and $N = \sum_{i=1}^{n} m_i$ the order of A. We denote

$$D = \begin{pmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & D_{n-1} & \\ & & & & D_n \end{pmatrix}, \quad L = \begin{pmatrix} 0 & & & & \\ A_2 & 0 & & & \\ & \ddots & \ddots & & \\ & & A_{n-1} & 0 & \\ & & & A_n & 0 \end{pmatrix},$$

$$A = D + L + L^T,$$

and we denote by a_{ij} the elements (pointwise) of A.

Since A is positive definite, there holds $a_{ii} > 0$, $i = 1, \ldots, N$. We assume that the following holds also.

HYPOTHESIS (H1).

(a) *The off-diagonal elements a_{ij}, $i \neq j$ of A are nonpositive.*

(b) *A is (weakly) diagonally dominant; i.e., there holds*

$$a_{ii} \geqslant \sum_{j \neq i} |a_{ij}|, \quad i = 1, \ldots, N,$$

and there exists at least one k, $1 \leqslant k \leqslant N$, such that

$$a_{kk} > \sum_{j \neq k} |a_{kj}|.$$

(c) *Each column of A_i, $i = 2, \ldots, n$, has at least one nonzero element.*

Hypothesis (H1)(a) implies that the positive definite symmetric matrix A is a Stieltjes matrix, i.e., a positive definite M-matrix [21], [22].

If the inequality of Hypothesis (H1)(b) holds strictly for all rows,

$$a_{ii} > \sum_{j \neq i} |a_{ij}|, \quad i = 1, \ldots, N,$$

then A is termed *strictly diagonally dominant.*

Let Σ be the symmetric block diagonal matrix with $m_i \times m_i$ blocks Σ_i satisfying

$$\begin{aligned} \Sigma_1 &= D_1, \\ \Sigma_i &= D_i - A_i \Sigma_{i-1}^{-1} A_i^T, \quad 2 \leqslant i \leqslant n. \end{aligned} \tag{3}$$

Then the block Cholesky factorization of A can be written as

$$A = (\Sigma + L)\Sigma^{-1}(\Sigma + L^T).$$

The factor $\Sigma + L$ is block lower bidiagonal. Since A is positive definite symmetric, the factorization can be carried out.

The following results concerning the properties of the Σ_i are well known, but as we did not find them in the literature in a form suitable for our application, we give them here for completeness. These properties provide guidance in our selection of preconditioning matrices for the conjugate gradient method.

Let

$$B = \begin{pmatrix} B_1 & -C^T \\ -C & B_2 \end{pmatrix},$$

with B_1 and B_2 square, be a symmetric positive definite M-matrix, which implies that the diagonal elements are positive and the off-diagonal elements are nonpositive.

LEMMA 1. *$B_2' = B_2 - CB_1^{-1}C^T$ is a symmetric positive definite M-matrix.*

B_2' is called the Schur complement of B_1 in B. For properties of the Schur complement see [4].

Proof. We can write

$$\begin{pmatrix} B_1 & 0 \\ 0 & B_2 - CB_1^{-1}C^T \end{pmatrix} = \begin{pmatrix} I & 0 \\ CB_1^{-1} & I \end{pmatrix} B \begin{pmatrix} I & B_1^{-1}C^T \\ 0 & I \end{pmatrix}.$$

Since the leading principal minors of B are unchanged by the transformation on the right side of the equality, the matrix on the left side is positive definite, and hence so is B_2'. In particular the diagonal elements of B_2' are positive and, as $B_1^{-1} > 0$ and $C \geqslant 0$ hold, it follows that the off diagonal elements are nonpositive.

It can be shown easily that *if B_2 is strictly diagonally dominant, then B_2' is also.*

Now we apply these results to A with $B_1 = D_1$, $-C^T = (A_2^T\, 0 \cdots 0)$, and

$$B_2 = \begin{pmatrix} D_2 & A_3^T & & & \\ A_3 & D_3 & A_4^T & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & D_{n-1} & A_n^T \\ & & & A_n & D_n \end{pmatrix}.$$

We have

$$B_2' = \begin{pmatrix} D_2 - A_2 D_1^{-1} A_2^T & A_3^T & & & \\ A_3 & D_3 & A_4^T & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & D_{n-1} & A_n^T \\ & & & A_n & D_n \end{pmatrix}.$$

There follows

THEOREM 1. *Under Hypothesis* (H1) *all the Σ_i are symmetric strictly diagonally dominant M-matrices.*

It is of interest to note, that in the particular case of the model problem, the block Cholesky factorization can be shown to reduce to a Fast Poisson Solver [18].

3. Incomplete block Cholesky factorization. Because of the work and storage that may be required in large problems for computing the Σ_i, carrying out the complete block Cholesky factorization is not of interest to us here as a general means for solving (2). For example, for the two-dimensional model problem, although $\Sigma_1 = D_1$ is tridiagonal, Σ_1^{-1} and hence Σ_i, $i \geqslant 2$, are dense.

In this paper our interest focuses on approximate block Cholesky factorizations obtained by using in (3) instead of Σ_{i-1}^{-1} a sparse approximation Λ_{i-1}. One thereby obtains instead of Σ the block diagonal matrix Δ with $m_i \times m_i$ blocks Δ_i satisfying

$$\Delta_1 = D_1, \tag{4a}$$

$$\Delta_i = D_i - A_i \Lambda_{i-1} A_i^T, \quad 2 \leqslant i \leqslant n, \tag{4b}$$

where for each i in (4b), Λ_{i-1} is the sparse approximation to Δ_{i-1}^{-1}. The incomplete block Cholesky preconditioning matrix for use with the conjugate gradient algorithm is then

$$M = (\Delta + L)\Delta^{-1}(\Delta + L^T). \tag{5}$$

One has

$$M = A + \Delta - D + L\Delta^{-1}L^T = A + R,$$

where R is a block diagonal matrix

$$R = \begin{pmatrix} R_1 & & & & \\ & R_2 & & & \\ & & \ddots & & \\ & & & R_{n-1} & \\ & & & & R_n \end{pmatrix}$$

with

$$R_1 = \Delta_1 - D_1 = 0,$$

$$R_i = \Delta_i - D_i + A_i \Delta_{i-1}^{-1} A_i^T, \quad 2 \leqslant i \leqslant n.$$

The factor $\Delta + L$ in (5) is lower block bidiagonal. Using the Cholesky factors L_i of Δ_i,

$$\Delta_i = L_i L_i^T,$$

one can express M in terms of (point) lower and upper triangular factors

$$M = \begin{pmatrix} L_1 & & & \\ W_2 & L_2 & & 0 \\ & \ddots & \ddots & \\ & & W_{n-1} & L_{n-1} \\ & & & W_n \quad L_n \end{pmatrix} \begin{pmatrix} L_1^T & W_2^T & & \\ & L_2^T & W_3^T & \\ & & \ddots & \ddots \\ 0 & & L_{n-1}^T & W_n^T \\ & & & L_n^T \end{pmatrix}, \tag{6}$$

where

$$W_i = A_i L_{i-1}^{-T}, \quad i = 2, \ldots, n.$$

This form is generally more efficient computationally than is (5). For specific Λ_i of interest, we show in subsequent sections that all the Δ_i are positive definite,

which implies that the above factorization can be carried out.

Note that in the conjugate gradient algorithm M is not required explicitly, only the linear system $Mz^k = r^k$ need be solved for z^k. Since this can be done with block backward and forward substitution, the block off-diagonal elements W_i need not be computed explicitly. The requisite products with vectors can be obtained by solving linear systems with triangular coefficient matrices L_i and L_i^T. Generally, for preconditionings of interest, the Δ_i, and correspondingly the L_i, will be sparse. These features were first used in this context by R. R. Underwood in [20], where block incomplete Cholesky preconditioning for the conjugate gradient algorithm was introduced.

For the standard five-point discretization of (1) in two dimensions, D_i is tridiagonal, and A_i is diagonal. This is the case on which this paper focuses. Of central interest is the choice that the Λ_{i-1} be tridiagonal, so that all the Δ_i in (4b) are tridiagonal. Correspondingly, in the remainder of this section we discuss techniques for approximating the inverse of a tridiagonal, diagonally-dominant matrix.

Let

$$T = \begin{pmatrix} a_1 & -b_1 & & & \\ -b_1 & a_2 & -b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -b_{m-2} & a_{m-1} & -b_{m-1} \\ & & & -b_{m-1} & a_m \end{pmatrix} \tag{7}$$

be a nonsingular tridiagonal matrix. We assume that the following holds.

HYPOTHESIS (H2). *The elements a_i and b_i of T satisfy*

$$a_i > 0, \quad 1 \leq i \leq m,$$
$$b_i > 0, \quad 1 \leq i \leq m-1,$$

and T is strictly diagonally dominant.

3.1. Diagonal approximation. The simplest approximation $\tilde{T}_1$ of T^{-1} we consider is the diagonal matrix whose elements are

$$(\tilde{T}_1)_{ii} = \frac{1}{(T)_{ii}}. \tag{8}$$

3.2. Banded approximation from the exact inverse. One can do much better than the diagonal approximation $\tilde{T}_1$ by using the following powerful result, which characterizes the inverses of symmetric tridiagonal matrices, (cf. [1], [10]).

THEOREM 2. *There exist two vectors u and $v \in R^m$ such that*

$$(T^{-1})_{ij} = u_i v_j \qquad \text{for} \quad i \leq j.$$

Since the inverse of T is

$$T^{-1} = \begin{pmatrix} u_1v_1 & u_1v_2 & \cdots & u_1v_m \\ u_1v_2 & u_2v_2 & \cdots & u_2v_m \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ u_1v_m & u_2v_m & \cdots & u_mv_m \end{pmatrix},$$

one can compute recursively the components of u and v. Under Hypothesis (H2) T is positive definite, so that T^{-1} is also, which implies that $u_i \neq 0$, $v_i \neq 0$, for all i. We remark that all of Hypothesis (H2), which will be used later, is not required for Theorem 2. It is necessary only that T in (7) be nonsingular and irreducible (all of the b_i nonzero).

LEMMA 2. *The components of u and v can be computed as follows:*

$$
\begin{aligned}
u_1 &= 1, \quad u_2 = \frac{a_1}{b_1}, \\
u_i &= \frac{a_{i-1}u_{i-1} - b_{i-2}u_{i-2}}{b_{i-1}}, \quad 3 \leqslant i \leqslant m, \\
v_m &= \frac{1}{-b_{m-1}u_{m-1} + a_m u_m}, \\
v_i &= \frac{1 + b_i u_i v_{i+1}}{a_i u_i - b_{i-1}u_{i-1}}, \quad 2 \leqslant i \leqslant m-1, \\
v_1 &= \frac{1 + b_1 u_1 v_2}{a_1 u_1}.
\end{aligned} \tag{9}
$$

Proof. By substitution.

Alternative recurrences for generating u and v can be obtained by several means, such as by computing the first and last columns of T^{-1} from the Cholesky factors of T. For numerical computation scaling may be required in (9) to prevent underflow or overflow, or it may be desirable to work with the ratios u_{i+1}/u_i and v_{i+1}/v_i considered below. If only a few of the main diagonals of T^{-1} are required and not u and v explicitly, the diagonals can be computed conveniently from the Cholesky factors of T.

Several papers have characterized the elements of inverses of diagonally dominant matrices. In [15] results are proved for tridiagonal matrices and in [5] they are extended to matrices of larger bandwidth. It is known that the elements of $(T^{-1})_{ij}$ are bounded in an exponentially decaying manner along each row or column. Specifically, there exist $\rho < 1$ and a constant C_0 such that

$$(T^{-1})_{ij} \leqslant C_0 \rho^{|i-j|}.$$

This result does not imply that the elements actually decay along each row; it merely provides a bound. With Hypothesis (H2), however, one can prove the following:

LEMMA 3. *Under Hypothesis* (H2) *the sequence $\{u_i\}_{i=1}^m$ is strictly increasing and the sequence $\{v_i\}_{i=1}^m$ is strictly decreasing.*

Proof. It is clear that $u_2 = a_1/b_1 > 1 = u_1$. The proof continues by induction, using formulas of Lemma 2. Since $u_{i-1} > u_{i-2}$, one has from (9) that

$$u_i > u_{i-1}\left(\frac{a_{i-1} - b_{i-2}}{b_{i-1}}\right) > u_{i-1},$$

because $a_{i-1} - b_{i-1} - b_{i-2} > 0$. To prove that the v_i are decreasing we need to modify the formulas of Lemma 2 slightly, using the ones for u to simplify those for v. Note that

$$a_i u_i - b_{i-1}u_{i-1} - b_i u_{i+1} = 0$$

and

$$(a_{i+1}u_{i+1} - b_i u_i)v_{i+1} = 1 + b_{i+1}u_{i+1}v_{i+2}.$$

Thus

$$v_i = \frac{a_{i+1}u_{i+1}v_{i+1} - b_{i+1}u_{i+1}v_{i+2}}{b_i u_{i+1}} = \frac{a_{i+1}}{b_i}\, v_{i+1} - \frac{b_{i+1}}{b_i}\, v_{i+2}, \quad \text{for } i \leqslant m-2,$$

and

$$v_{m-1} = \frac{a_m}{b_{m-1}}\, v_m.$$

Clearly $v_{m-1} > v_m$, and by induction $v_i > v_{i+1}$ $((a_{i+1} - b_{i+1})/b_i) > v_{i+1}$.

Note that we can prove the same result if we suppose only that T is diagonally dominant with $a_1 > b_1$ and $a_m > b_{m-1}$.

One can characterize the decay of the element along a row away from the diagonal. Let $\bar{\alpha}_i$ and $\bar{\beta}_i$ be such that $u_i = \bar{\alpha}_{i-1}u_{i-1}$, $v_i = v_{i-1}/\bar{\beta}_{i-1}$, $i \geqslant 2$. We have

$$\bar{\alpha}_i = \frac{a_i}{b_i} - \frac{b_{i-1}}{b_i}\frac{1}{\bar{\alpha}_{i-1}}, \quad \bar{\alpha}_1 = \frac{a_1}{b_1},$$

$$\bar{\beta}_i = \frac{a_{i+1}}{b_i} - \frac{b_{i+1}}{b_i}\frac{1}{\bar{\beta}_{i+1}}, \quad \bar{\beta}_{m-1} = \frac{a_m}{b_{m-1}}.$$

In the general case we do not know the solution of the recurrences (which are simply the recurrences for computing the elements of T^{-1}), but the previous discussion gives us the bounds

$$\bar{\alpha}_i > \frac{a_i - b_{i-1}}{b_i} > 1,$$

$$\bar{\beta}_i > \frac{a_{i+1} - b_{i+1}}{b_i} > 1.$$

In particular, we have, for $i > j$,

$$(T^{-1})_{ij} = \frac{1}{\bar{\alpha}_{i-1}\cdots\bar{\alpha}_j}(T^{-1})_{ii} \leqslant \frac{(T^{-1})_{ii}}{\prod_{k=j}^{i-1}\left(\frac{a_k - b_{k-1}}{b_k}\right)}.$$

If $1/\rho = \min_{k\geqslant 2}((a_k - b_{k-1})/b_k)$ we find, for $i > j$,

$$(T^{-1})_{ij} \leqslant (T^{-1})_{ii}\rho^{i-j}, \quad \rho < 1.$$

This latter bound is not very sharp. For example, for the matrix T with $a_i = 4$, $i = 1, \ldots, m$ and $b_i = 1$, $i = 1, \ldots, m-1$, which will be of interest later, we get $\rho = 1/3$. But for this case

$$\bar{\alpha}_1 = 4, \quad \bar{\alpha}_i = 4 - \frac{1}{\bar{\alpha}_{i-1}}, \quad i \geqslant 2.$$

The $\bar{\alpha}_i$ form a decreasing sequence that converges very quickly towards $2 + \sqrt{3} \approx 3.732$, which corresponds to a reduction factor of $1/(2 + \sqrt{3}) \approx 0.2679$, which is considerably less than $1/3$. Of course if the a_i's and b_i's are constant, we could construct the inverse in another way from the eigenvalues and eigenvectors of T, which are known in this case.

It is of importance to observe that if T is strictly diagonally dominant the elements of the inverse decrease strictly away from the diagonal -- the stronger the diagonal dominance the faster the decay. This suggests the following means for approximating the inverse of T with a matrix of small bandwidth.

If A is any matrix, denote by $\mathbf{B}(A,p)$ the band matrix consisting of the $2p+1$ main diagonals of A. For a banded approximation $\tilde{T}_2(p)$ to the inverse of T we consider

$$\tilde{T}_2(p) = \mathbf{B}(T^{-1},p) \tag{10}$$

with p small, say 1 or 2.

3.3. Approximation from Cholesky factors. Another way of approximating T^{-1} is to use the Cholesky factorization of T,

$$T = U^T U,$$

with

$$U^T = \begin{pmatrix} \gamma_1 & & & \\ -\delta_1 & \gamma_2 & & 0 \\ & \ddots & \ddots & \\ & & -\delta_{m-2} & \gamma_{m-1} \\ & & & -\delta_{m-1} \quad \gamma_m \end{pmatrix}$$

a lower bidiagonal matrix. We have

$$\gamma_1^2 = a_1, \qquad \gamma_1\delta_1 = b_1,$$
$$\delta_{i-1}^2 + \gamma_i^2 = a_i, \quad \gamma_i\delta_i = b_i, \quad i \geqslant 2.$$

The δ_i's are positive, and the diagonal dominance of T implies

$$\delta_i < \gamma_i, \quad 1 \leqslant i \leqslant m-1.$$

The matrix U^{-T} is lower triangular and dense. We denote

$$U^{-T} = \begin{pmatrix} \frac{1}{\gamma_1} & & & & \\ \zeta_1 & \frac{1}{\gamma_2} & & 0 & \\ \eta_1 & \zeta_2 & \frac{1}{\gamma_3} & & \\ \vdots & \ddots & \ddots & \ddots & \\ & \cdots & \eta_{m-2} & \zeta_{m-1} & \frac{1}{\gamma_m} \end{pmatrix}.$$

It is easy to see that the elements of U^{-T} can be computed diagonal by diagonal, since

$$\zeta_i = \frac{\delta_i}{\gamma_i\gamma_{i+1}}, \quad 1 \leqslant i \leqslant m-1, \qquad \eta_i = \frac{\delta_i\zeta_{i+1}}{\gamma_i}, \quad 1 \leqslant i \leqslant m-2,$$

and so on. We note also that U^{-T} can be generated diagonal by diagonal by taking successive terms of its Neumann series in U^T.

We have the following result similar to the one for the inverse of T.

LEMMA 4. *For each row, the elements of U^{-T} decrease away from the diagonal.*

Proof. Since $\delta_i/\gamma_i < 1$ we have $\eta_{i-1} < \zeta_i < 1/\gamma_{i+1}$; the proof is the same for the other elements.

As an approximation for U^{-T} we can, therefore, take $\mathbf{B}(U^{-T},p)$ with p small. As an approximation for T^{-1} we can use correspondingly

$$\widetilde{T}_3(p) = \mathbf{B}(U^{-1},p)\mathbf{B}(U^{-T},p). \tag{11}$$

Note that $\widetilde{T}_3(p)$ is positive definite. For $p=1$, one has the tridiagonal matrix

$$\widetilde{T}_3(1) = \begin{pmatrix} \frac{1}{\gamma_1^2}+\zeta_1^2 & \frac{\zeta_1}{\gamma_2} & & \\ \frac{\zeta_1}{\gamma_2} & \frac{1}{\gamma_2^2}+\zeta_2^2 & \frac{\zeta_2}{\gamma_3} & \\ & \ddots & \ddots & \ddots \end{pmatrix}.$$

Unless the Cholesky decomposition is needed explicitly, it is necessary to compute only the square of the γ_i's to obtain $\widetilde{T}_3(1)$, because

$$\frac{\zeta_i}{\gamma_{i+1}} = \frac{b_i}{\gamma_i^2\gamma_{i+1}^2}, \quad \zeta_i^2 = \frac{a_{i+1}-\gamma_{i+1}^2}{\gamma_i^2\gamma_{i+1}^2}.$$

Thus one obtains $\widetilde{T}_3(1)$ directly from a_i, b_i, and γ_i^2.

Note that $\widetilde{T}_3(1)T$ is the five diagonal matrix

$$\widetilde{T}_3(1)T = \begin{pmatrix} 1 & -b_2u_1v_3 & b_2u_1v_2 & & \\ -b_2u_1v_3 & 1 & -b_3u_2v_4 & b_3u_2v_3 & \\ b_2u_1v_2 & -b_3u_2v_4 & 1 & -b_4u_3v_3 & b_4u_3v_4 \\ & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

Since the u_iv_j are expected to be small, $\widetilde{T}_3(1)$ can be expected to be a good approximation to T^{-1}.

3.4. Polynomial approximation. A classical way to obtain an approximation of T^{-1} is to use a polynomial expansion in powers of T. Let D_T be the diagonal of T and denote

$$\overline{T} = T - D_T.$$

Then

$$T^{-1} = (I + D_T^{-1}\overline{T})^{-1}D_T^{-1}.$$

Since T is strictly diagonally dominant, the corresponding Jacobi iteration is convergent, which implies that the eigenvalues of the Jacobi iteration matrix $-D_T^{-1}\overline{T}$ (which are real) are contained in $(-1,+1)$ (see for example [12], [21]). Thus one can write

$$(I + D_T^{-1}\overline{T})^{-1} = \sum_{k=0}^{\infty}(-1)^k(D_T^{-1}\overline{T})^k,$$

the series being convergent.

The powers of $D_T^{-1}\overline{T}$ contain more and more nonzero diagonals as k increases. As an approximate inverse we can take simply the first few terms, which are the sparsest ones (Taking only the first term gives the diagonal approximation $\widetilde{T}_1$ of §3.1.). It is well known, however, that if the eigenvalues of $D_T^{-1}\overline{T}$ are not close enough to zero, the truncated series could be a poor approximation. Better polynomial approximations can be found (cf. [14]).

Let $S = D_T^{-1}\overline{T}$, and suppose we want to find a polynomial P of degree less than or equal to ν that minimizes $\|(I+S)^{-1} - P(S)\|_2$. Since S is similar to a symmetric matrix there exists a unitary matrix Q such that

$$S = Q\Theta Q^T,$$

where Θ is a diagonal matrix whose elements are the eigenvalues of S.

We have

$$P(S) = QP(\Theta)Q^T,$$

so that

$$\|(I+S)^{-1} - P(S)\|_2 = \|(I+\Theta)^{-1} - P(\Theta)\|_2 \leqslant C_1 \max_i \left\| \frac{1}{1+\theta_i} - P(\theta_i) \right\|,$$

where C_1 is constant and θ_i, $1 \leqslant i \leqslant m$, are the eigenvalues of S. To minimize the right-hand side (the minimum, of course, need not minimize also the left-hand side) we must find the polynomial approximation of $1/(1+x)$ on the set of eigenvalues θ_i of S. Instead we could solve the simpler problem of finding

$$\min \max_{\theta \in [\theta_1, \theta_m]} \left\| \frac{1}{1+\theta} - P(\theta) \right\|,$$

where θ_1 (respectively, θ_m) is the smallest (respectively, largest) eigenvalue of S. The solution to this problem is given by Chebyshev polynomials.

In general, however, even the extremal eigenvalues θ_1 and θ_m are not known; all one knows is that $-1 < \theta_1 \leqslant \theta_m < 1$ holds. Since $1/(1+x)$ is discontinuous at $x = -1$, we could simply compute P to yield

$$\min_P \max_{\theta \in [0,1]} \left\| \frac{1}{1+\theta} - P(\theta) \right\|.$$

This should give a good result for the eigenvalues between 0 and 1, but a poor one for the smaller eigenvalues. For a first degree polynomial we obtain

$$P(\Theta) \approx 0.9412 - 0.4706\,\Theta.$$

As will be seen later, it is possible to obtain a better approximation when additional information about the eigenvalues is available. In general, we shall be considering tridiagonal polynomial approximations $\widetilde{T}$ to T^{-1} of the form

$$\widetilde{T}_4(\alpha,\beta) = \alpha D_T^{-1} + \beta D_T^{-1}\overline{T}D_T^{-1}, \tag{12}$$

where the coefficients α and β are real.

3.5. Comparison of approximations for the model problem. We now compare the above approximations for the model problem, for which in (7) $a_i = 4$, $i = 1, \ldots, m$, and $b_i = 1$, $i = 1, \ldots, m-1$. The case $m = 10$ is considered. The upper triangular part of the inverse T^{-1} as computed in double precision

FORTRAN on an IBM 3081 by MATLAB [19] to four places is

$$\begin{bmatrix} 0.2679 & 0.0718 & 0.0192 & 0.0052 & 0.0014 & 0.0004 & \cdots \\ & 0.2872 & 0.0770 & 0.0206 & 0.0055 & 0.0015 & \cdots \\ & & 0.2886 & 0.0773 & 0.0207 & 0.0056 & \cdots \\ & & & 0.2887 & 0.0773 & 0.0207 & \cdots \\ & & & & 0.2887 & 0.0773 & \cdots \\ & & & & & 0.2887 & \cdots \\ & & & & & \cdots & \cdots \end{bmatrix},$$

illustrating the rapid decay away from the diagonal.

For the different approximations $\tilde{T}_i$ to T^{-1} we get the following results (using MATLAB), as summarized in Table 1 and Figure 1. The last entry

TABLE 1

Values of $\|\tilde{T}_i - T^{-1}\|_2$ *for the model problem,* $m = 10$.

Approximation to T^{-1}		$\|\tilde{T}_i - T^{-1}\|_2$
Diagonal (§3.1)	$\tilde{T}_1$	0.2305
Banded from exact inverse (§3.2)	$\tilde{T}_2(1)$	0.0456
Banded from exact inverse (§3.2)	$\tilde{T}_2(2)$	0.0104
From Cholesky factors (§3.3)	$\tilde{T}_3(1)$	0.0569
From Cholesky factors (§3.3)	$\tilde{T}_3(2)$	0.0134
Polynomial (§3.4)	$\tilde{T}_4(1,-1)$	0.1106
Polynomial (§3.4)	$\tilde{T}_4(.9412,-.4706)$	0.1888
Polynomial (§3.4)	$\tilde{T}_4(1.1429,-1.1429)$	0.0577

	.5	.6	.7	.8	.9	1.0	1.1	1.2	1.3	1.4
$\tilde{T}_1$	2,8	7	9		3	7		1	3	2,8
$\tilde{T}_2(1)$				3,9	1,5,9	1,7,8	3,4			
$\tilde{T}_2(2)$					6,7,8,9	0,0,1,3,3,4				
$\tilde{T}_3(1)$				8,8	3,3	0,0,7,8	3,3			
$\tilde{T}_3(2)$					7,7,7,9	0,0,1,3,3,3				
$\tilde{T}_4$ (1,−1)			7,7	2,2,9,9	6,6,9,9					
$\tilde{T}_4$ (.9412,−.4706)		1,6	4	2	1,7	2,4,6,6				
$\tilde{T}_4$ (1.1429,−1.1429)				8,8	4,4	2,2,9,9	4,4			

FIG. 1. *Tabular display of eigenvalue distributions of* $\tilde{T}_i T$.

$\tilde{T}_4(1.1429,-1.1429)$ corresponds to the min max polynomial over $\theta\in[-0.5,0.5]$, which interval is approximately the one bounded by the extremal eigenvalues $\theta_1 \approx -0.4797$, $\theta_m \approx 0.4797$ of S for this problem. Thus one should expect this polynomial to give a better approximation than the other two. Values of $\|\tilde{T}_i - T^{-1}\|_2$ to four places are given in Table 1. Figure 1 depicts the eigenvalues of $\tilde{T}_i T$ in a format that permits a rough comparison of their distributions: The eigenvalues are rounded to two decimal places, and the least significant digit is entered in the column corresponding to the first digit(s).

It is evident that for this model problem the banded approximations from the exact inverse and the approximations from Cholesky factors can give better approximations to T^{-1} than the polynomial expansions, in the sense of clustering about 1 of the eigenvalues of $\tilde{T}_i T$ and smallness of $\|\tilde{T}_i - T^{-1}\|_2$. It would be of interest to know if the same results would hold for matrices T of larger bandwidth.

4. Block preconditionings for the two-dimensional case. Based on the approximate inverses of §3, we define the corresponding block preconditionings for the two-dimensional problem. For this case the D_i are tridiagonal, and our goal is to keep the Δ_i, $i \geq 2$, in (4b) tridiagonal, or possibly of slightly greater bandwidth. For the preconditionings discussed below, only the Cholesky factors L_i of the Δ_i are actually stored for computational purposes, corresponding to (6).

4.1. The block preconditionings.

4.1.1. BDIA. The diagonal approximation $\tilde{T}_1$ in (8) is used; Λ_{i-1} is diagonal with

$$(\Lambda_{i-1})_{jj} = \frac{1}{(\Delta_{i-1})_{jj}}.$$

The Δ_i's are tridiagonal matrices at each stage differing from D_i only in their diagonal elements.

4.1.2. INV(1). The banded approximation $\tilde{T}_2(1)$ in (10) from the exact inverse is used,

$$\Lambda_{i-1} = \mathbf{B}(\Delta_i^{-1}, 1).$$

Each of the Δ_i's is tridiagonal. At each stage we compute two vectors u and v and use them to obtain the three main diagonals of Δ_i^{-1}. We then compute and store the Cholesky factors of Δ_i. $2N$ words of storage are needed for M, as in BDIA. We do not consider here keeping more diagonals in the approximation to Δ_i^{-1} for this case, as the particularly simple expression in Theorem 2 becomes more complex if the Δ_i's have more than 3 diagonals.

4.1.3. CHOL(p). We use $\tilde{T}_3(p)$ from (11),

$$\Lambda_{i-1} = \mathbf{B}(U_i^{-1}, p)\mathbf{B}(U_i^{-T}, p),$$

where $\Delta_{i-1} = U_{i-1}^T U_{i-1}$, with U_{i-1} an upper triangular matrix. At each stage we compute U_{i-1}, which is (except for $i = 2$) a matrix with $p+1$ nonzero main diagonals. The first $p+1$ diagonals of U_i^{-1} can be computed diagonalwise starting from the main diagonal. Since Λ_{i-1} is a symmetric matrix of bandwidth $2p+1$, approximately $2m_1 + (p+1)\sum_{i=2}^{n} m_i$ words of storage are needed for Δ_i.

CHOL(p) is a special case of the following method proposed by Underwood [20] in a slightly different setting.

4.1.4. UND(p,q). For this case

$$\Lambda_{i-1} = \mathbf{B}(\mathbf{B}(U_{i-1}^{-1}, q-1)\mathbf{B}(U_{i-1}^{-T}, q-1), 2p-1),$$

with $q \geq p$. One computes the q main diagonals of U_{i-1}^{-T}, but then stores only the $2p-1$ main diagonals of the product to form Λ_{i-1}. More information about U_{i-1}^{-T} is used in UND(p,q) than in CHOL(p). The storage needed is $2m_1 + p\sum_{i=2}^{n} m_i$. Note that UND($p$,$p$) $\equiv$ CHOL($p-1$).

4.1.5. POL(α,β). We use the polynomial approximation $\tilde{T}_4(\alpha,\beta)$ defined in §3.4,

$$\Delta_{i-1} = D_{T,i-1} + \overline{T}_{i-1},$$

$$\Lambda_{i-1} = \alpha D_{T,i-1}^{-1} + \beta D_{T,i-1}^{-1}\overline{T}_{i-1}D_{T,i-1}^{-1}.$$

Each Δ_i is tridiagonal. Different values of α and β are used. The storage requirements are the same as for BDIA and INV(1).

4.2. Properties of the Δ_i. Now we study the properties of the Δ_i in order to prove that all of the methods described above can be carried out (i.e., we prove that the Δ_i satisfy hypothesis (H2) placed on T).

THEOREM 3. *Under Hypothesis* (H1) *each* Δ_i *computed by* BDIA, INV(1), CHOL(p), UND(p,q), *and* POL(α,β) *with* $\beta \leq 0$, $0 < \alpha \leq 1$, $\beta + \alpha \geq 0$ *is strictly diagonally dominant with positive diagonal elements and nonpositive off diagonal elements.*

Proof. This can be proved by induction using the same technique as in Lemma 1. As the proof is essentially the same for all cases, we carry it out only for CHOL(p).

Let

$$B = \begin{pmatrix} B_1 & -C^T \\ -C & B_2 \end{pmatrix}$$

be a positive definite M-matrix satisfying (H1) with B_1 and B_2 square, and let $B_1 = L_{B_1}L_{B_1}^T$ be the Cholesky decomposition of B_1. Denote

$$L_{B_1}^{-1} = \tilde{L}_{B_1}^{-1} + R_{B_1},$$

where $\tilde{L}_{B_1}^{-1}$ contains the $p+1$ diagonals of $L_{B_1}^{-1}$ that are kept for the approximation, and R_{B_1} contains the remaining diagonals. Under hypothesis (H1) both $\tilde{L}_{B_1}^{-1}$ and R_{B_1} are nonnegative. From the remark following Lemma 1 we know that $B_2 - CL_{B_1}^{-T}L_{B_1}^{-1}C^T$ is strictly diagonally dominant. We have

$$B_2 - C\tilde{L}_{B_1}^{-T}\tilde{L}_{B_1}^{-1}C^T = B_2 - CL_{B_1}^{-T}L_{B_1}^{-1}C^T + C(\tilde{L}_{B_1}^{-T}R_{B_1} + R_{B_1}^T\tilde{L}_{B_1}^{-1} + R_{B_1}^TR_{B_1})C^T.$$

The last matrix on the right is nonnegative, which implies that $B_2 - C\tilde{L}_{B_1}^{-T}\tilde{L}_{B_1}^{-1}C^T$ is at least as strongly diagonally dominant as is $B_2 - CL_{B_1}^{-T}L_{B_1}^{-1}C^T$. The desired result for CHOL(p) then follows by induction, taking $B_1 = D_1$, the first diagonal block of A.

4.3. Modified block preconditionings. It is known that point incomplete Cholesky decomposition can be modified to yield a better approximation to A in some cases. The modified decomposition is obtained from R by altering the diagonal elements of the Cholesky factors so that the row sums of M are equal to the corresponding row sums of A (i.e., the row sums of R are zero). This gives an improvement of the condition number of $M^{-1}A$ for natural ordering of the unknowns and for A diagonally dominant [7], [13].

As noted in §3, the remainder R for the block incomplete Cholesky preconditioning is a block diagonal matrix whose elements are

$$R_1 = 0,$$
$$R_i = \Delta_i - D_i + A_i\Delta_{i-1}^{-1}A_i^T = A_i(\Delta_{i-1}^{-1} - \Lambda_{i-1})A_i^T, \quad 2 \leqslant i \leqslant n.$$

Thus the row sum of Δ_{i-1}^{-1} must be available if R_i is to have row sum zero.

4.3.1. MINV(1). For the case of INV(1), Δ_{i-1}^{-1} itself is readily available, thus it is easy to define MINV(1), the modified form of INV(1): Compute Δ_{i-1}^{-1} at each stage from the two vectors u and v. Form the product $R_i = A_i[\Delta_{i-1}^{-1} - \mathbf{B}(\Delta_{i-1}^{-1},1)]A_i^T$, which is a matrix with positive elements except for the 3 main diagonals, which are zero. Then subtract from $D_i - A_i\mathbf{B}(\Delta_{i-1}^{-1},1)A_i^T$ the diagonal matrix made up of the row sums of R_i, to yield the modified Δ_i corresponding to a remainder with a zero row sum.

We note that it follows from Hypothesis (H1) that the remainder matrix is nonpositive definite, hence the eigenvalues of $M^{-1}A$ are greater than or equal to 1 for MINV(1).

THEOREM 4. *Under Hypothesis* (H1) *each* Δ_i *given by* MINV(1) *is a strictly diagonally dominant matrix with positive diagonal elements and negative off diagonal elements.*

Proof. Consider

$$\begin{pmatrix} B_1 & -C^T \\ -C & B_2 \end{pmatrix}.$$

Let $S_2 = C[B_1^{-1} - \mathbf{B}(B_1^{-1},1)]C^T$ and let R_2 be the diagonal matrix of row sums of S_2. Since $B_1^{-1} \geqslant 0$, the elements of S_2 and hence of R_2 are positive. Note that $B_2 - C\mathbf{B}(B_1^{-1},1)C^T - R_2$ has the same row sums as $B_2 - C[\mathbf{B}(B_1^{-1},1)]C^T - S_2 = B_2 - CB_1^{-1}C^T$. This, together with the positivity of the elements, shows that $B_2 - C\mathbf{B}(B_1^{-1},1)C^T - R_2$ is diagonally dominant.

4.3.2. MUND(p,q). For the other block preconditionings the row sums of R_i can be calculated easily, but not quite so directly. However, in UND(p,q) with $q > p$ a part of the remainder is immediately available and can be subtracted from the diagonal. Recall that

$$\Delta_{i-1} = L_{i-1}L_{i-1}^T,$$

$$R_i = A_i[\Delta_{i-1}^{-1} - \mathbf{B}(\mathbf{B}(L_{i-1}^{-T},q-1)\mathbf{B}(L_{i-1}^{-1},q-1),2p-1)].$$

Denote by $\tilde{L}_{i-1}^{-1} = \mathbf{B}(L_{i-1}^{-1},q-1)$ the q diagonals of the inverse of L_{i-1} that are computed, and by Q_{i-1} the diagonals that are not computed

$$L_{i-1}^{-1} = \tilde{L}_{i-1}^{-1} + Q_{i-1}.$$

Then

$$R_i = A_i[\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1} + \tilde{L}_{i-1}^{-T}Q_{i-1} + Q_{i-1}^T\tilde{L}_{i-1}^{-1} + Q_{i-1}^TQ_{i-1} - \mathbf{B}(\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1},2p-1)]A_i^T.$$

We can obtain $\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1} - \mathbf{B}(\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1},2p-1)$, since it is made up of the diagonals of the product that are not kept in the algorithm. Thus, instead of discarding these diagonals we could subtract their row sums from the main diagonal. This constitutes the algorithm MUND(p,q): Compute q diagonals of L_{i-1}^{-1}. Form the product $\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1}$. Use the $2p-1$ main diagonals to form $D_i - A_i\mathbf{B}(\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1},2p-1)A_i^T$. Let S_{i-1} be the matrix made up of the $q-p$ outer diagonals of $\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1}$. Compute the row sums of $A_iS_{i-1}A_i^T$ and subtract them from the diagonal of $D_i - A_i\mathbf{B}(\tilde{L}_{i-1}^{-T}\tilde{L}_{i-1}^{-1},2p-1)A_i^T$ to obtain Δ_i.

THEOREM 5. *Under Hypothesis* (H1) *each* Δ_i *given by* MUND(p,q) *is a strictly diagonally dominant matrix with positive diagonal elements and negative off-diagonal elements.*

Proof. Along the same lines as for Theorem 4.

4.4. Higher dimensions. One can develop block incomplete Cholesky factorizations for three dimensional problems similarly, using, for example, incomplete instead of complete factorizations L_i for the Δ_i. It is planned to investigate these preconditionings in a subsequent study.

5. Numerical experiments. In this section we present the results of numerical experiments on two-dimensional test problems comparing the preconditionings introduced in the previous sections and some other, commonly used, point and block preconditionings. The other preconditionings include: the point incomplete Cholesky decomposition IC(p,q) introduced by Meijerink and van der Vorst [16], [17], in which p bands adjacent to the main diagonal and q outer bands are kept in the factorization; its modified version MIC(p,q), of which the simplest MIC(1,1), first introduced by Dupont, Kendall, and Rachford for five diagonal matrices [7], is denoted here by DKR (and is used without parameters); symmetric successive overrelaxation (SSOR) and its block version BSSOR (which in our case is line SSOR); and for a few cases 1-line Jacobi preconditioning (LJAC). In addition, results will be given for some problems for the point Jacobi preconditioning DIAG, for which M is a diagonal matrix whose diagonal elements are those of A, and for conjugate gradients without preconditioning ($M = I$, the identity matrix).

For a five diagonal matrix the work per iteration and storage for each of the methods is given in Table 2. (For simplicity, the technique of [8] for reducing the work requirements of the conjugate gradient method is not incorporated.) The work is represented by number of floating point multiplies; about the same number of additions are required also.

Table 2 does not include the overhead operations required to construct M. If one carries out many iterations or solves several systems with different right-hand sides, then this overhead can usually be neglected. Specific cases are discussed in §5.1. Also not included in Table 2 is the work that might be required for evaluating iteration termination criteria.

It should be noted that the work requirements for the preconditionings depend on the manner in which the computer programs are written. Generally we have organized our programs with a preference toward multiplication over division; for example, in INV(1) we use Varga's implementation of Gauss elimination

TABLE 2

Work per iteration and storage for the preconditionings.

Preconditioning M	Mults.	Storage
I	10N	0
DIAG	11N	0
IC(1,1), DKR	16N	N
SSOR	17N	0
IC(1,2), MIC(1,2)	18N	3N
IC(1,3), MIC(1,3)	20N	4N
IC(2,4)	24N	6N
BSSOR	18N	2N
BDIA, INV(1), MINV(1), POL(α,β)	18N	2N
CHOL(p), UND($p+1,q$), MUND($p+1,q$)	$(4p+14)$N	$(p+1)$N

for tridiagonal matrices, which stores the reciprocals of the diagonals [21]. If a division is carried out, as in DIAG when it is desired neither to scale the matrix in advance nor to store the reciprocals of the diagonal, then, as is customary, a division is counted as equivalent to a multiply. In CHOL(p: $p>1$), UND(p,q), and MUND(p,q) routines from LINPACK [6] are used, but the operation counts entered in Table 2 are made to correspond to the manner in which we implement the other preconditionings. Thus the entries in Table 2, though basically consistent, should be considered as approximate. They are used in subsequent tables to convert observed number of iterations to computational work.

Our implementation of the conjugate gradient algorithm requires 4 N-vectors of storage, plus 3 N-vectors for the matrix A and 1 N-vector for the right-hand side. If it is not necessary to save the right-hand side, then 1 N-vector of storage could be eliminated. The additional storage required by each of the preconditionings is given in the last column of Table 2.

5.1. First test problem. The first test problem is the model problem

$$-\Delta u = f \qquad \text{in } \Omega \text{ the unit square } (0,1)\times(0,1)$$

with

$$u\Big|_{\partial\Omega} = 0.$$

We use the standard five point stencil on a square mesh with $h = (n+1)^{-1}$, $N = n^2$, and natural ordering to obtain the corresponding linear algebraic system (2). The experimental results are given for different values of h and different stopping criteria. An estimate of the condition number of $M^{-1}A$ is given for each of the preconditionings, as obtained from the conjugate gradient algorithm (cf. [3]), and for small dimension ($n = 10$) the complete spectrum of $M^{-1}A$ is visualized.

The computations were carried out in double precision FORTRAN on an IBM 3081. Unless otherwise noted the solution of the linear system is smooth

(the right-hand side b in (2) corresponds to the solution $\xi_i(\xi_i-1)\eta_j(\eta_j-1)\exp(\xi_i\eta_j)$ at a point (ξ_i,η_j)), and the starting vector has random elements in [-1,1]. As the number of additions is roughly the same as the number of multiplications, we indicate only the work required for the multiplications. The divisions that may appear to be needed by some methods are not indicated, since they can be removed with alternative coding. In Table 3 are given the number of iterations

TABLE 3

Number of iterations and total work per point for $\|r^k\|_\infty/\|r^0\|_\infty \leq 10^{-6}$. *Test problem* 1, $N = 2500$.

M	# iterations	work/N
I	109	1090
DIAG	109	1199
IC(1,1)	33	528
IC(1,2)	21	378
IC(1,3)	17	340
IC(2,4)	12	288
DKR	23	368
MIC(1,2)	17	306
MIC(1,3)	14	280
SSOR $\omega = 1$	40	680
SSOR $\omega = 1.7$	21	357
LJAC	80	1040
BSSOR $\omega = 1$	28	504
BSSOR $\omega = 1.7$	16	288
BDIA	22	396
POL(1,-1)	18	324
POL(0.9412,-0.4706)	21	378
POL(1.143,-1.143)	17	306
INV(1)	15	270
MINV(1)	11	198
CHOL(1)	16	288
CHOL(2)	12	264
CHOL(3)	9	234
CHOL(4)	8	240
CHOL(5)	7	238
UND(2,3)	15	270
UND(2,4)	15	270
UND(3,4)	11	242
UND(3,5)	11	242
UND(4,5)	9	234
UND(4,6)	9	234
UND(5,6)	7	210
MUND(2,3)	12	216
MUND(2,4)	10	180
MUND(2,5)	9	162
MUND(3,4)	10	220
MUND(3,5)	8	176
MUND(3,6)	8	176
MUND(4,5)	8	208
MUND(4,6)	7	182
MUND(5,6)	7	210

and the corresponding total work per point required to achieve the stopping criterion $\|r^k\|_\infty / \|r^0\|_\infty \leq 10^{-6}$, for the case $N = 2500$. The value $\omega = 1.7$ for SSOR and BSSOR is the observed optimal for each case to the nearest 0.1 for minimizing the number of iterations required for convergence.

From Table 3, the following observations can be made.

(i) For the patterns chosen, the larger the number of diagonals in the incomplete Cholesky decomposition, the fewer the number of iterations required for convergence, as observed in [17] for the point preconditionings.

(ii) The modified versions of the preconditionings give better results (for this problem and ordering of the mesh points).

(iii) In general, there is a trade off between storage and execution speed, but if a low storage point-preconditioning is desired, DKR seems a good choice. SSOR can give good results, but suitable parameter values are needed.

(iv) For methods of comparable storage the block methods give better results than point methods, both in terms of number of iterations and work requirements.

(v) For CHOL(p) it is not effective to go to values of p larger than $p = 3$, and, as observed also in [2], for UND(p,q) to values of q beyond $q = p+1$. It is better to use the additional information given by UND(p,q) for larger q to obtain a modified version of the factorization for $q = p+1$.

(vi) The best polynomial, as expected, is POL(1.1429,-1.1429).

(vii) For this problem the best all-around preconditioning appears to be MINV(1), because it has very low storage requirements and gives almost the best work count -- approximately half of IC(1,2) and two thirds of MIC(1,2), which require more storage.

Table 4 gives a comparison of some of the methods for solving the test problem to only moderate accuracy $\|r^k\|_\infty / \|r^0\|_\infty \leq 10^{-4}$, comparable to discretization error. The conclusions drawn for the smaller residuals in Table 3 are in general unchanged.

TABLE 4

Number of iterations and total work per point for $\|r^k\|_\infty / \|r^0\|_\infty \leq 10^{-4}$. *Test problem* 1, $N = 2500$.

M	# iterations	work/N
I	63	630
IC(1,1)	20	320
IC(2,4)	7	168
DKR	16	256
SSOR $\omega = 1.7$	13	221
BSSOR $\omega = 1.7$	10	180
INV(1)	9	162
MINV(1)	7	126
CHOL(1)	9	162
CHOL(5)	4	136

In Table 5 are given the values of the smallest and largest eigenvalues of $M^{-1}A$, as estimated by the conjugate gradient algorithm, as well as the corresponding condition numbers. It is seen that a considerable reduction in the condition number can be achieved with some of the modified preconditionings, with only a low cost in storage.

TABLE 5

Extremal eigenvalues and condition number of $M^{-1}A$. *Test problem* 1, $N = 2500$.

M	$\lambda_{\min}(M^{-1}A)$	$\lambda_{\max}(M^{-1}A)$	$\kappa(M^{-1}A)$
I	0.0076	7.992	1053
IC(1,1)	0.0128	1.206	94.0
IC(1,2)	0.033	1.179	35.6
IC(1,3)	0.049	1.131	23.2
IC(2,4)	0.091	1.138	12.5
DKR	1.003	15.36	15.3
MIC(1,2)	1.003	8.83	8.3
MIC(1,3)	1.006	6.19	6.15
SSOR $\omega = 1.$	0.0075	1.	132.5
SSOR $\omega = 1.7$	0.040	1.	25.1
LJAC	0.0038	1.99	527.
BSSOR $\omega = 1.$	0.0150	1.	66.8
BSSOR $\omega = 1.7$	0.074	1.	13.5
BDIA	0.024	1.023	42.6
POL(1,-1)	0.035	1.	28.7
POL(0.9412,-0.4706)	0.027	1.002	37.2
POL(1.143,-1.143)	0.043	1.023	23.8
INV(1)	0.059	1.073	18.2
MINV(1)	1.006	4.261	4.24
CHOL(1)	0.050	1.050	20.8
CHOL(2)	0.090	1.065	11.8
CHOL(3)	0.142	1.076	7.56
CHOL(4)	0.204	1.078	5.29
CHOL(5)	0.272	1.078	3.97
UND(2,3)	0.058	1.07	18.5
UND(2,4)	0.059	1.073	18.2
UND(2,5)	0.059	1.073	18.2
UND(3.4)	0.104	1.086	10.5
UND(3,5)	0.106	1.089	10.2
UND(4,5)	0.162	1.091	6.75
UND(4,6)	0.166	1.096	6.59
UND(5,6)	0.228	1.088	4.78
MUND(2,3)	0.102	1.242	12.2
MUND(2,4)	0.202	1.564	7.74
MUND(2,5)	0.380	2.024	5.33
MUND(3,4)	0.164	1.242	7.58
MUND(3,5)	0.291	1.518	5.22
MUND(3,6)	0.483	1.887	3.91
MUND(4,5)	0.234	1.221	5.21
MUND(4,6)	0.375	1.449	3.87
MUND(5,6)	0.309	1.197	3.88

In Table 6 are given the estimated condition numbers $\kappa(M^{-1}A)$ for different values of $n = (1/h) - 1$. The quantity α is the estimated value, from the $n = 25$ and $n = 50$ data, of the exponent corresponding to the assumed asymptotic relationship $\kappa(M^{-1}A) \sim Ch^{-\alpha}$, where C is a constant. It is known theoretically that for $M = I$ and $M = \text{IC}(1,1)$ there holds $\kappa(M^{-1}A) = O(h^{-2})$ and that for $M = \text{DKR}$, $\kappa(M^{-1}A) = O(h^{-1})$. The values of α obtained from the numerical

TABLE 6

Estimated condition number for different mesh sizes and exponent α of asymptotic dependence on $h = 1/(n+1)$. Test problem 1.

M	$\kappa(M^{-1}A)$				α
	$n=10$	$n=20$	$n=25$	$n=50$	
I	48.37	178.1	273.3	1053	2.00
IC(1,1)	5.10	16.59	25.	94	1.97
IC(1,2)	2.38	6.67	9.8	35.6	1.91
IC(1,3)	1.80	4.56	6.6	23.2	1.87
IC(2,4)	1.32	2.75	3.8	12.5	1.77
DKR	3.04	5.93	7.4	15.3	1.08
MIC(1,2)	1.84	3.36	4.2	8.3	1.01
MIC(1,3)	1.49	2.56	3.15	6.1	0.98
SSOR $\omega = 1.$	6.88	23.12	35.	132	1.97
LJAC	24.68	89.5	137.	527	2.00
BSSOR $\omega = 1.$	3.93	12.04	18.	66.7	1.94
BDIA	2.76	7.9	11.7	42.5	1.91
POL(1,-1)	2.09	5.52	8.	28.6	1.89
POL(0.9412,-0.4706)	2.5	7.	10.3	37.1	1.90
POL(1.143,-1.143)	1.86	4.7	6.7	23.8	1.88
INV(1)	1.61	3.74	5.3	18.2	1.83
MINV(1)	1.3	1.94	2.31	4.23	0.90
CHOL(1)	1.73	4.18	6.	20.8	1.85
CHOL(2)	1.32	2.65	3.65	11.85	1.75
CHOL(3)	1.14	1.93	2.53	7.54	1.62
CHOL(4)	1.06	1.55	1.95	5.28	1.48
CHOL(5)	1.026	1.34	1.61	3.98	1.34
UND(2,3)	1.63	3.8	5.4	18.52	1.83
UND(2,4)	1.62	3.75	5.33	18.24	1.83
UND(3,4)	1.26	2.42	3.3	10.47	1.71
UND(3,5)	1.25	2.39	3.24	10.24	1.71
UND(4,5)	1.12	1.8	2.33	6.73	1.57
UND(4,6)	1.11	1.77	2.28	6.54	1.56
UND(5,6)	1.05	1.47	1.82	4.8	1.44
MUND(2,3)	1.39	2.76	3.79	12.95	1.82
MUND(2,4)	1.29	2.1	2.72	7.74	1.55
MUND(2,5)	1.28	1.89	2.26	5.33	1.27
MUND(3,4)	1.18	1.97	2.58	7.55	1.59
MUND(3,5)	1.15	1.67	2.04	5.22	1.39
MUND(3,6)	1.14	1.6	1.85	3.9	1.11
MUND(4,5)	1.09	1.57	1.96	5.22	1.45
MUND(4,6)	1.07	1.43	1.68	3.8	1.21
MUND(5,6)	1.04	1.35	1.62	3.9	1.30

experiments are in accord with these relationships. We see that all the point incomplete decompositions IC(p,q) seem to be $O(h^{-2})$, although the more diagonals that are taken the slower is the convergence to this asymptotic behavior. The MIC methods are $O(h^{-1})$.

For the block methods INV and CHOL the limiting value of α seems to be two, and for MINV one. The observed values of α for the range of h considered are smaller for the block methods than for the point methods with the same storage. It is difficult to assess from the results the order of the MUND methods; we believe that they are somewhere between 1 and 2, closer to 1 if more diagonals are used to form M. Finally, Table 6 shows that even for smaller values of n block methods give better reduction of the condition number than point methods.

It is well known that the rate of convergence of the conjugate gradient method depends not only on the condition number but on the distribution of the interior eigenvalues as well. It is therefore of interest to compare the eigenvalue spectra for the different methods. These are compared for $n = 10$ in Figs. 2-4. Each eigenvalue is designated by a vertical bar drawn at the appropriate abscissa value. This representation depicts in an easily observable manner the separation and clustering of the eigenvalues.

The spectra for all of the methods shown in Fig. 2 are on the same scale for easy comparison. From the figure it is seen that for the block methods the eigenvalues are more clustered than for the point ones having the same storage requirements. (The relatively greater clustering for block SSOR over point SSOR is a well-known property, cf. [9].) The values $\omega = 1.7$ and $\omega = 1.5$ are to the nearest 0.1 those for which the condition numbers for SSOR and BSSOR, respectively, are smallest. The point modified methods, for which the eigenvalue range is different than for the other methods, are shown separately in Fig. 3. Fig. 4 shows on the same scale four methods with comparable storage: IC(1,1) and DKR, with one vector of storage, and INV(1) and MINV(1) with two. Spectra for block SSOR preconditioning for the values $\omega = 1.0(0.1)1.9$ can be found in [2], and enlargements showing the fine structure of the spectra of Figs. 2-4 are in an Appendix to [2], available separately from the authors.

Table 7 gives the number of iterations required to solve the test problem for different convergence criteria. For these cases the initial approximation was $x^0 \equiv 0$, and the solution was the same smooth vector as for Tables 4 and 5 with $N = 2500$.

From these results, it appears that, at least for the test problem with a smooth solution, the relative norm of the residual gives a good stopping criterion.

In Table 8 we give results for $N = 2500$ for the same smooth solution as for previous tables, with two different choices of the starting vector, $x^0 \equiv 0$ and x^0 consisting of random numbers in [-1,1]. The stopping criterion is $\|r^k\|_\infty / \|r^0\|_\infty \leqslant 10^{-6}$. The initial approximation x^0 random appears to give better results. This feature will be developed in a subsequent study.

From the tables one can conclude that for this test problem block methods give better results than point ones. The most promising block method is MINV(1). Since the setup time for constructing M was not included in the tables, it is of interest to consider it, as it can be of importance if only one problem is to be solved or only a few iterations taken. Table 9 gives the effect of including the setup time for three of the preconditionings for the $N = 2500$ test problem. Times are in CPU seconds for an IBM 3081 computer. Even if the setup times are

included, MINV(1) still gives considerable improvement for this problem.

The effects of Neumann boundary conditions were examined as well in [2], where it was found that the relative merits of the different preconditionings are about the same as for this test problem.

TABLE 7

Number of iterations for different convergence criteria. Test problem 1, $x^0 \equiv 0$.

M	Number of iterations			
	$\frac{\|r^k\|_\infty}{\|r^0\|_\infty} \leq 10^{-6}$	$\|x-x^k\|_\infty \leq 10^{-6}$	$\|x-x^k\|_2 \leq 10^{-6}$	$\|x-x^k\|_A \leq 10^{-6}$
I	117	99	114	110
IC(1,1)	38	31	36	35
IC(1,2)	26	22	26	24
IC(1,3)	21	19	22	20
IC(2,4)	16	14	16	15
DKR	25	18	22	21
MIC(1,2)	18	14	17	16
MIC(1,3)	18	16	18	17
SSOR $\omega = 1.$	44	37	43	41
SSOR $\omega = 1.7$	22	17	20	19
BSSOR $\omega = 1.$	36	28	34	32
BSSOR $\omega = 1.7$	18	15	18	16
BDIA	27	24	28	26
POL(1,-1)	23	20	24	22
INV(1)	19	16	19	18
MINV(1)	13	9	11	11
CHOL(1)	20	18	21	19
CHOL(2)	15	13	16	14
CHOL(3)	12	11	13	12
CHOL(4)	10	9	10	10
CHOL(5)	9	8	9	8
UND(2,3)	19	16	19	18
UND(3,4)	14	13	15	14
UND(4,5)	12	10	12	11
UND(5,6)	9	8	10	9
MUND(2,3)	15	14	16	15
MUND(2,4)	13	11	13	12
MUND(2,5)	12	9	11	10
MUND(3,4)	12	11	13	12
MUND(3,5)	11	9	11	10
MUND(4,5)	10	9	10	10
MUND(4,6)	9	8	9	9
MUND(5,6)	9	8	9	8

TABLE 8

Number of iterations for $\|r^k\|_\infty / \|r^0\|_\infty \leq 10^{-6}$ *for different starting vectors. Test problem* 1.

M	# of iterations	
	$x^0 = 0$	x^0 random
I	117	109
IC(1,1)	38	33
IC(1,2)	26	21
IC(1,3)	21	17
IC(2,4)	16	12
DKR	25	23
MIC(1,2)	18	17
MIC(1,3)	18	14
SSOR $\omega = 1.$	44	40
SSOR $\omega = 1.7$	22	21
BSSOR $\omega = 1.$	36	28
BSSOR $\omega = 1.7$	18	16
BDIA	27	22
POL(1,-1)	23	18
INV(1)	19	15
MINV	13	11
CHOL(1)	20	16
CHOL(2)	15	12
CHOL(3)	12	9
CHOL(4)	10	8
CHOL(5)	9	7
UND(2,3)	19	15
UND(3,4)	14	11
UND(4,5)	12	9
UND(5,6)	9	7
MUND(2,3)	15	12
MUND(2,4)	13	10
MUND(2,5)	12	9
MUND(3,4)	12	10
MUND(3,5)	11	8
MUND(4,5)	10	8
MUND(4,6)	9	7
MUND(5,6)	9	7

TABLE 9

Total time including setup in CPU *seconds for* $\|r^k\|_\infty / \|r^0\|_\infty \leq 10^{-6}$. *Test problem* 1.

M	total time
IC(1,1)	1.37
INV(1)	0.963
MINV(1)	0.723

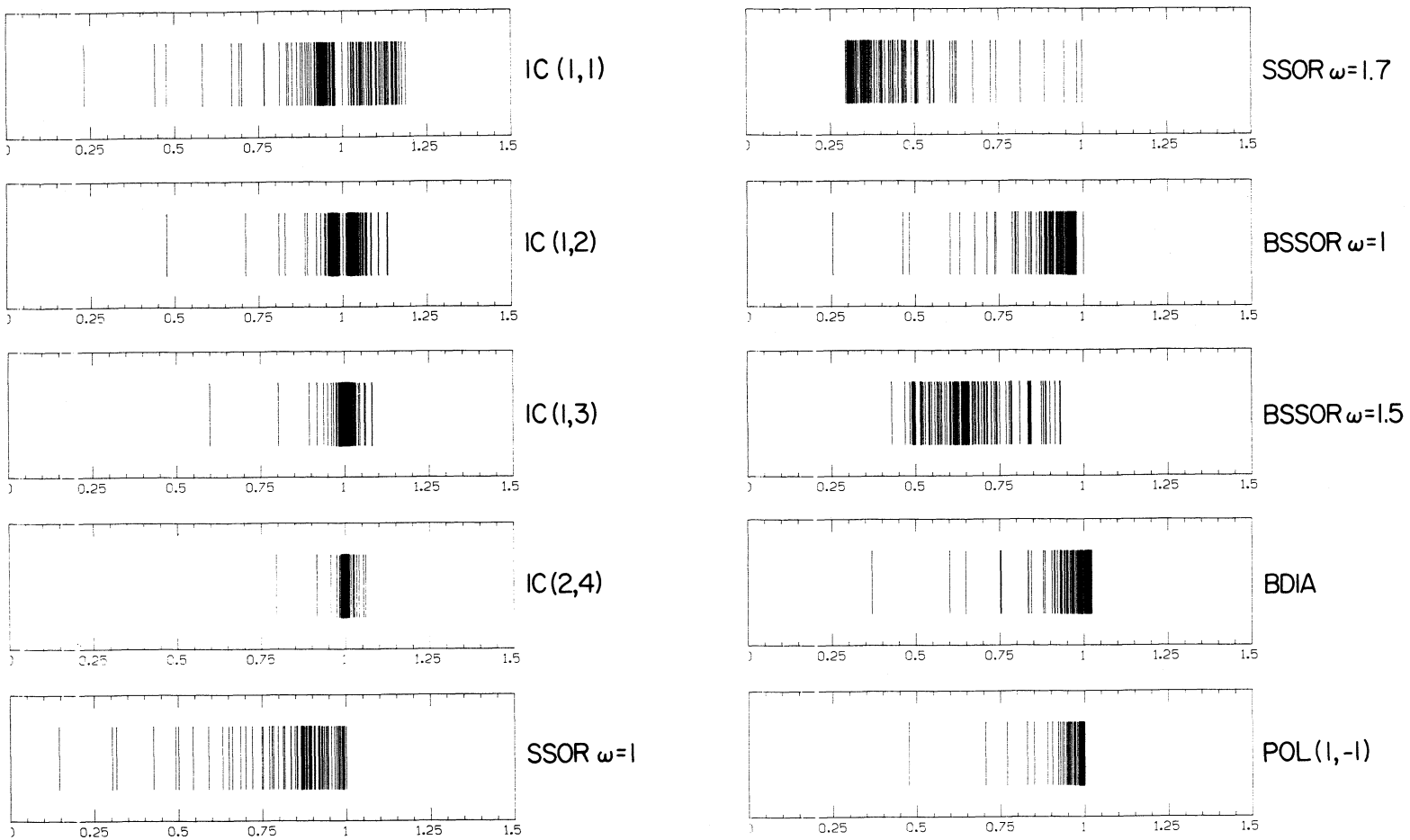


FIG. 2. *Spectra of $M^{-1}A$ for different preconditionings M. Test problem* 1. $N = 100$.

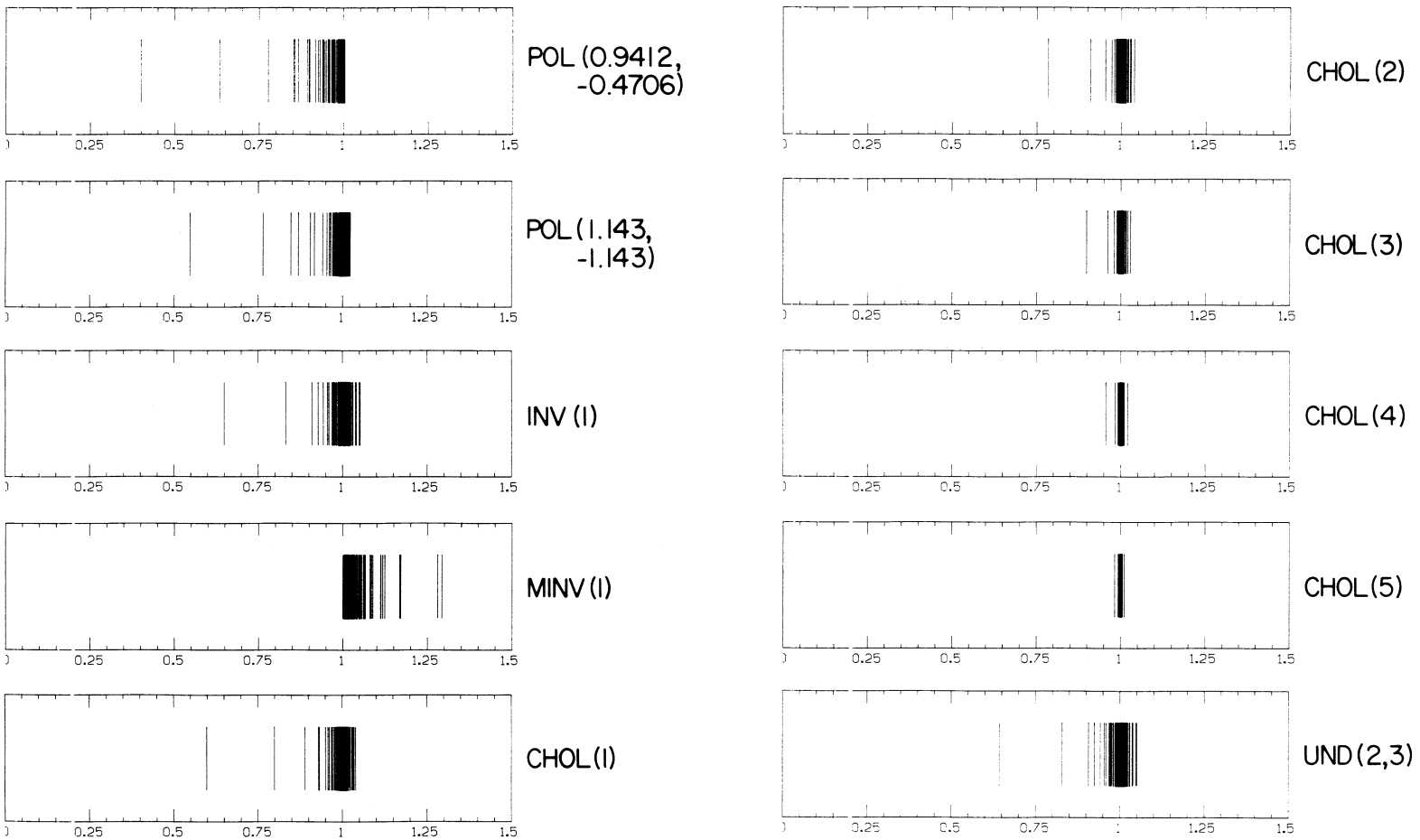


FIG. 2. (*cont.*)

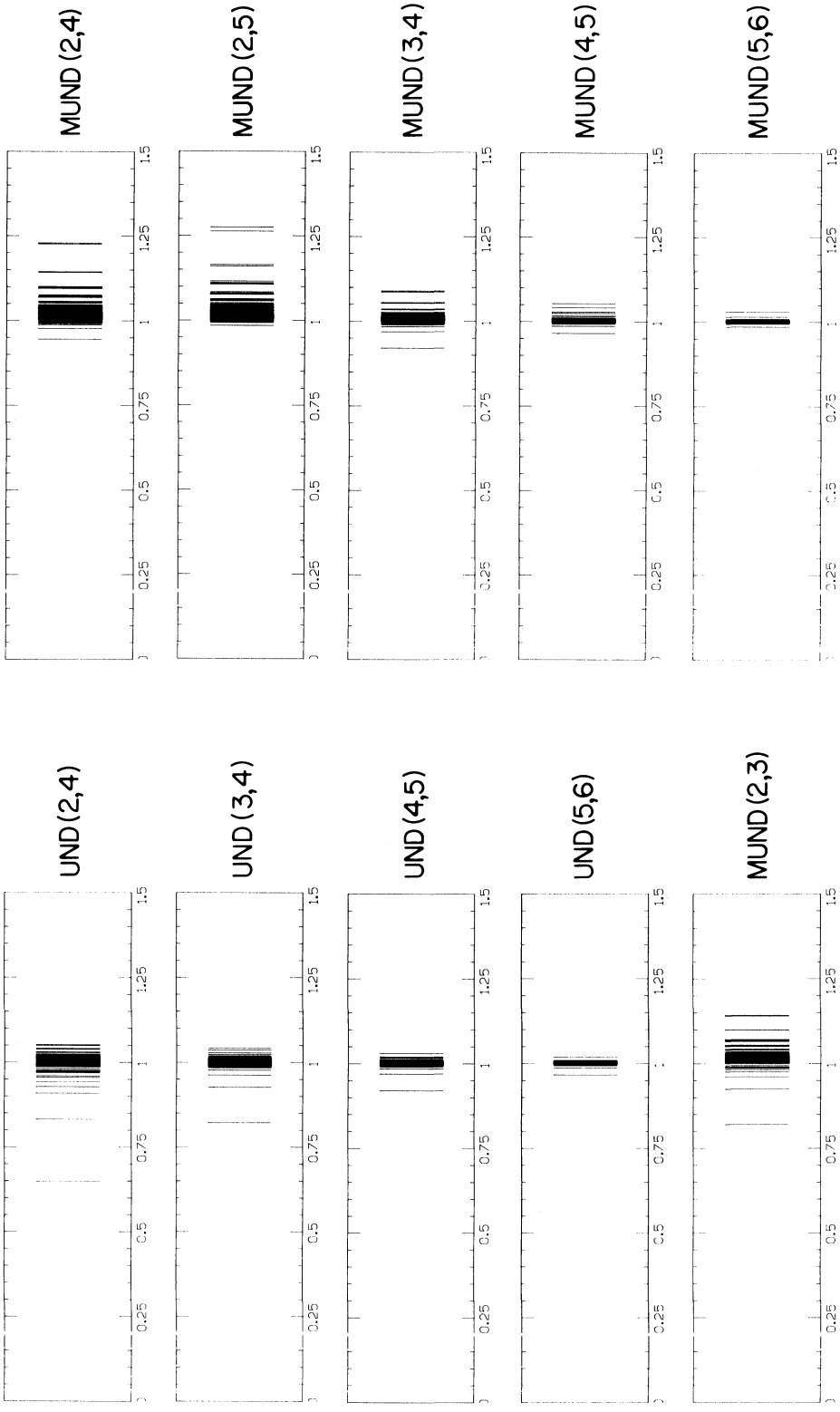


FIG. 2. (*cont.*)

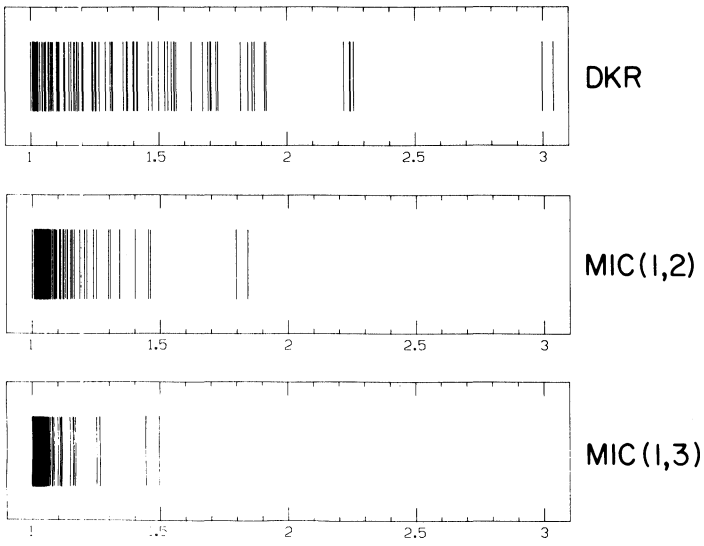


FIG. 3. *Spectra of $M^{-1}A$ for modified preconditionings. Test problem* 1. $N = 100$.

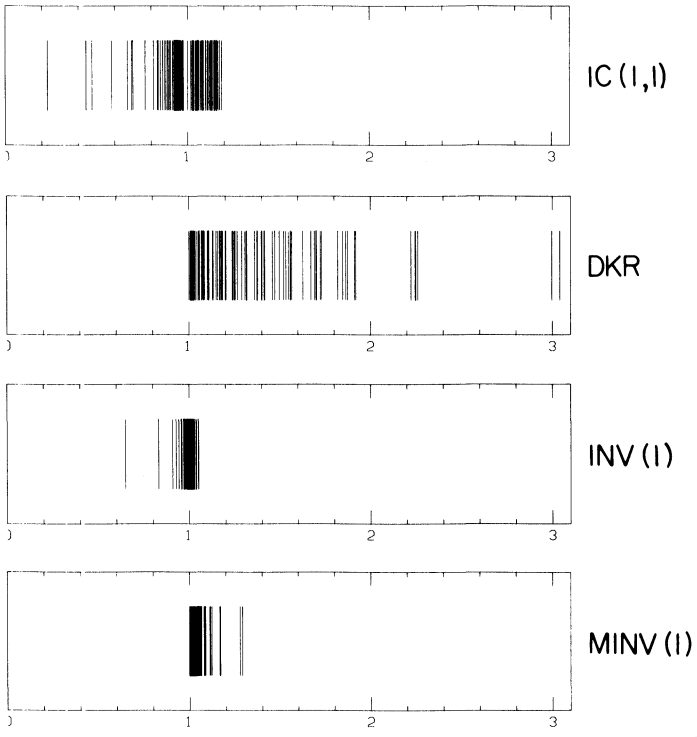


FIG. 4. *Spectra of $M^{-1}A$ for four preconditionings with comparable, minimal storage. Test problem* 1. $N = 100$.

5.2. Second test problem. We solve the linear system obtained by the standard five point discretization of the problem

$$-\frac{\partial}{\partial\xi_1}\left(\lambda(\xi_1,\xi_2)\frac{\partial u}{\partial\xi_1}\right)-\frac{\partial}{\partial\xi_2}\left(\lambda(\xi_1,\xi_2)\frac{\partial u}{\partial\xi_2}\right)=f \quad \text{in } \Omega=(0,1)\times(0,1),$$

$$u=0 \quad \text{on } \partial\Omega,$$

for the discontinuous λ depicted in Fig. 5. The solution is the same smooth one as for the first test problem, the starting vector is random, and the stopping criterion is $\|r^k\|_\infty/\|r^0\|_\infty \leqslant 10^{-6}$.

Table 10 gives the results for the number of iterations, the work required, and an estimate of the condition number as obtained from the conjugate gradient parameters. The values $\omega=1.6$ for SSOR and $\omega=1.5$ for BSSOR are the observed optimal ones to the nearest 0.1.

The very large condition numbers for most of the entries result from the small first eigenvalue, which is isolated from the others. Thus the number of iterations does not change much, for example from IC(1,1), which has a small isolated eigenvalue, to DKR, which has all eigenvalues greater than one. It is the distribution of the other eigenvalues that is important. In terms of work per point, block methods give better results than point ones. Again MINV(1) seems a good compromise between efficiency and storage. This example shows that block methods can be effective for problems with coefficients having large jump discontinuities.

5.3. Third test problem. This example, which is frequently used in the literature, was presented in [21]. The problem is to solve

$$-\frac{\partial}{\partial\xi_1}\left(\lambda_1\frac{\partial u}{\partial\xi_1}\right)-\frac{\partial}{\partial\xi_2}\left(\lambda_2\frac{\partial u}{\partial\xi_2}\right)+\sigma u=0 \quad \text{in } \Omega=(0,2.1)\times(0,2.1),$$

$$\left.\frac{\partial u}{\partial n}\right|_{\partial\Omega}=0.$$

The domain is shown in Fig. 6 and depicts the values of the coefficients, which are discontinuous. The solution is $u\equiv 0$.

We take $h=1/42$, x^0 a vector with random elements in [-1,1], and stopping criterion $\|x^k\|_\infty \leqslant 10^{-6}$. The results are given in Table 11. The values $\omega=1.7$ for SSOR and $\omega=1.5$ for BSSOR are the observed optimal ones to the nearest 0.1.

Table 11 indicates that for this problem the larger the number of diagonals retained, the lower the work required for convergence. This holds both for point and block methods. Generally, the block methods are slightly better.

In order to compare our methods with those presented by Meijerink and Van der Vorst [17] for this problem, we give the results in Table 12 for convergence criterion $\|r^k\|_2 \leqslant 10^{-6}$. For the IC methods, we obtain about the same results as in [17], within a few iterations. (The distribution from which the starting vectors were drawn is different—our random numbers are between -1 and 1, while theirs are between 0 and 1.)

To compare point and block methods with the same storage, one can take, for example, IC(1,2) or MIC(1,2) and CHOL(2). It is clear that the block method is better. The situation is the same if more diagonals are taken. To get down to 16

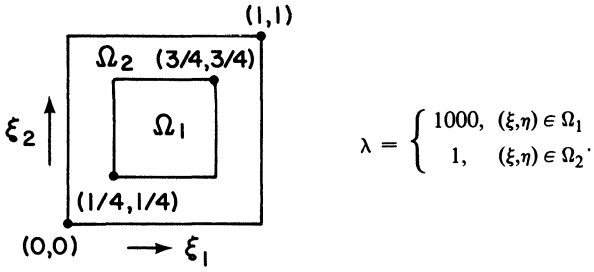
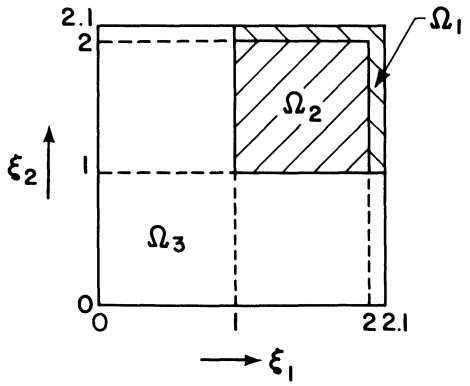


FIG. 5. *Test problem* 2.

TABLE 10

Number of iterations, total work per point, and estimated condition number of $M^{-1}A$. *Test problem* 2, $N = 2500$, $\|r^k\|_\infty / \|r^0\|_\infty \leqslant 10^{-6}$.

M	# iterations	work/N	$\kappa(M^{-1}A)$
DIAG	137	1507	
IC(1,1)	47	752	46770
IC(1,2)	30	540	17062
IC(1,3)	25	500	11102
IC(2,4)	18	432	5668
DKR	32	512	40
MIC(1,2)	23	414	26
MIC(1,3)	20	400	24
SSOR $\omega = 1.$	55	935	66162
SSOR $\omega = 1.6$	36	612	16620
BSSOR $\omega = 1.$	41	738	33929
BSSOR $\omega = 1.5$	23	414	14777
BDIA	34	612	21489
POL(1,-1)	28	504	14182
INV(1)	22	396	8790
MINV(1)	17	306	20
CHOL(1)	24	432	10288
CHOL(2)	18	396	5531
CHOL(3)	14	364	3307
CHOL(4)	12	360	2154
CHOL(5)	10	340	1490
UND(2,3)	22	396	8946
UND(3,4)	17	374	4762
UND(4,5)	14	364	2876
UND(5,6)	12	360	1899
MUND(2,3)	19	342	5825
MUND(2,4)	17	306	3472
MUND(2,5)	16	288	2135
MUND(3,4)	15	330	3355
MUND(3,5)	14	308	2135
MUND(3,6)	14	308	1379
MUND(4,5)	12	312	2136
MUND(4,6)	12	312	1416
MUND(5,6)	11	330	1451
LJAC	111	1443	



	λ_1	λ_2	σ
Ω_1	1.	1.	0.02
Ω_2	2.	2.	0.03
Ω_3	3.	3.	0.05

FIG. 6. *Test problem* 3.

TABLE 11

Number of iterations and total work per point for $\|x^k\|_\infty \leq 10^{-6}$. *Test problem* 3, $N = 1849$.

M	# iterations	work/N
IC(1,1)	74	1184
IC(1,2)	47	846
IC(1,3)	38	760
IC(2,4)	29	696
DKR	53	848
MIC(1,2)	36	648
MIC(1,3)	29	580
SSOR $\omega = 1.$	88	1496
SSOR $\omega = 1.7$	52	884
BSSOR $\omega = 1.$	65	1170
BSSOR $\omega = 1.5$	46	828
BDIA	52	936
POL(1,-1)	43	774
INV(1)	34	612
MINV(1)	25	450
CHOL(1)	36	648
CHOL(2)	28	616
CHOL(3)	22	572
CHOL(4)	19	570
CHOL(5)	16	544
UND(2,3)	34	612
UND(3,4)	26	572
UND(4,5)	21	546
UND(5,6)	18	540
MUND(2,3)	28	504
MUND(2,4)	25	450
MUND(2,5)	23	414
MUND(3,4)	23	506
MUND(3,5)	21	462
MUND(4,5)	19	494
MUND(4,6)	18	468
MUND(5,6)	17	510

TABLE 12

Number of iterations and total work per point for $\|r^k\|_2 \leq 10^{-6}$. *Test problem* 3, $N = 1849$.

M	# iterations	work/N
IC(1,1)	79	1264
IC(1,2)	49	882
IC(1,3)	39	780
IC(2,4)	30	720
DKR	66	1056
MIC(1,2)	43	774
MIC(1,3)	35	700
SSOR $\omega = 1.$	94	1598
SSOR ω_{opt}	56	952
BSSOR $\omega = 1.$	68	1224
BSSOR ω_{opt}	48	864
BDIA	55	990
POL(1,-1)	45	810
INV(1)	36	648
MINV(1)	29	522
CHOL(1)	38	684
CHOL(2)	29	638
CHOL(3)	23	598
CHOL(4)	20	600
CHOL(5)	17	578
UND(2,3)	36	648
UND(3,4)	28	616
UND(4,5)	22	572
UND(5,6)	19	570
MUND(2,3)	30	540
MUND(2,4)	26	468
MUND(2,5)	24	432
MUND(3,4)	24	528
MUND(3,5)	22	484
MUND(4,5)	20	520
MUND(4,6)	19	494
MUND(5,6)	17	510

iterations with point preconditioning IC(5,7) is used in [17], but approximately the same goal can be achieved with only six instead of 12 vectors of storage using the block preconditioning CHOL(5).

6. Concluding remarks. The above examples show that, for linear problems coming from finite-difference approximations of elliptic partial differential equations, the block preconditionings we have introduced can give better results for two-dimensional problems than the corresponding point ones currently in use. The results are better also than for block SSOR preconditioning. Generally, for natural ordering of the unknowns, the modified methods give better results for our test problems than unmodified ones. Particularly attractive is the preconditioning INV(1)—and its modified form MINV(1)—because of the low storage require-

ments and rapid convergence. The results for three dimensional problems await further study. It would be of interest to explore the behavior of our block preconditioning methods on more general problems such as the ones arising from finite element approximation with node orderings leading to a block tridiagonal matrix.

7. Acknowledgment. We are pleased to acknowledge that much of this work has been stimulated by the paper of R. R. Underwood [20] and our personal association with him.

REFERENCES

[1] E. ASPLUND, *Inverse of matrices* $\{a_{ij}\}$ *which satisfy* $a_{ij} = 0$ *for* $j > i + p$, Math. Scand., 7 (1959), pp. 57-60.

[2] P. CONCUS, G. H. GOLUB, AND G. MEURANT, *Block preconditioning for the conjugate gradient method,* Report LBL-14865, Lawrence Berkeley Lab., Univ. of California, 1982.

[3] P. CONCUS, G. H. GOLUB, AND D. P. O'LEARY, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations,* in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 309-332.

[4] R. W. COTTLE, *Manifestations of the Schur complement,* Linear Algebra Appl., 8 (1974), pp. 120-211.

[5] S. DEMKO, *Inverses of band matrices and local convergence of spline projections,* SIAM J. Numer. Anal., 14 (1977), pp. 616-619.

[6] J. J. DONGARRA, C. B. MOLER, J. R. BUNCH, AND G. W. STEWART, LINPACK *Users' Guide,* Society for Industrial and Applied Mathematics, Philadelphia, 1979.

[7] T. DUPONT, R. P. KENDALL, AND H. RACHFORD, *An approximate factorization procedure for solving self adjoint elliptic difference equations,* SIAM J. Numer. Anal., 5 (1968), pp. 559-573.

[8] S. EISENSTAT, *Efficient implementation of a class of preconditioned conjugate gradient methods,* SIAM J. Sci. Statist. Comput., 2 (1981), pp. 1-4.

[9] L. W. EHRLICH, *The block symmetric successive overrelaxation method,* SIAM J. Appl. Math., 12 (1964), pp. 807-826.

[10] D. K. FADDEEV, *Properties of the inverse of a Hessenberg matrix,* in Numerical Methods and Computational Issues, 5 (1981), V. P. Ilin and V. N. Kublanovskaya, eds. (in Russian).

[11] A. GEORGE AND J. W. H. LIU, *Algorithms for matrix partitioning and the numerical solution of finite element systems,* SIAM J. Numer. Anal., 15 (1978), pp. 297-327.

[12] G. H. GOLUB AND G. MEURANT, *Résolution numérique des grands systèmes linéaires,* Collection de la Direction des Etudes et Recherches de l'Electricité de France, vol. 49, Eyrolles, Paris, 1983.

[13] I. GUSTAFSSON, *A class of first order factorization methods,* BIT, 18 (1978), pp. 142-156.

[14] O. G. JOHNSON, C. A. MICCHELLI, AND G. PAUL, *Polynomial preconditioners for conjugate gradient calculations,* SIAM J. Numer. Anal., 20 (1983), pp. 362-376.

[15] D. KERSHAW, *Inequalities on the elements of the inverse of a certain tridiagonal matrix,* Math. Comp., 24 (1970), pp. 155-158.

[16] J. A. MEIJERINK AND H. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix,* Math. Comp., 31 (1977), pp. 148-162.

[17] J. A. MEIJERINK AND H. VAN DER VORST, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems,* J. Comput. Phys., 44 (1981), pp. 134-155.

[18] G. MEURANT, *The Fourier/tridiagonal method for the Poisson equation from the point of view of block Cholesky factorization,* Report LBID-764, Lawrence Berkeley Lab., Univ. of California, 1983.

[19] C. MOLER, MATLAB *Users' Guide,* Dept. of Computer Science, Univ. of New Mexico, Albuquerque, NM, 1981.

[20] R. R. UNDERWOOD, *An approximate factorization procedure based on the block Cholesky decomposition and its use with the conjugate gradient method,* Report NEDO-11386, General Electric Co., Nuclear Energy Div., San Jose, CA, 1976.

[21] R. S. VARGA, *Matrix Iterative Analysis,* Prentice Hall, Englewood Cliffs, NJ, 1962.

[22] D. M. YOUNG, *Iterative Solution of Large Linear Systems,* Academic Press, New York, 1971.