SANDIA REPORT
SAND87—0903 • UC—32
Unlimited Release
Printed July 1987

Two Classes of Preconditioners Computed Using Block Matrix Factorization Techniques

Lorraine S. Baca, Douglas E. Salane

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

When printing a copy of any digitized SAND Report, you are required to update the markings to current standards.

SF2900Q(8-81)

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

UC-32
SAND87-0903
Unlimited Release
Printed July 1987

Two Classes of Preconditioners Computed Using Block Matrix Factorization Techniques

Lorraine S. Baca and Douglas E. Salane
Numerical Mathematics Division 1422

Abstract

Two methods for computing preconditioners for nonsymmetric block tridiagonal systems of linear equations are investigated. Adaptable general purpose implementations are given for both methods.

1. Introduction

The numerical solution of two-dimensional partial differential equations (PDE) frequently requires the solution of large highly structured systems of linear algebraic equations. Block matrix factorization is a technique that can be used to compute effective preconditioners which improve the performance of many popular iterative methods for solving these systems ([1],[2],[3],[4] and [5]).

In this work we examine two classes of SSOR like preconditioners [11] for nonsymmetric systems that result from different implementations of block matrix factorization. The preconditioners in one class are similar to those first presented in [3]. These generally have low storage requirements and are not expensive to compute. We suggest a general technique for computing them that employs partial pivoting. The other class of preconditioners is computed using complete block factorization. These preconditioners can lead to very effective iterative methods but are more expensive to compute.

Both classes of preconditioners are parameterized by a positive integer parameter p. For smaller values of p, the preconditioners are generally less expensive to compute and store. For larger values of p the preconditioners require more storage and computation but result in more reliable and faster converging iterative methods. The following is an outline of the paper.

A discussion of how block matrix factorization can be used to obtain preconditioners is given in section 2. In section 3 the two classes of preconditioners are derived and a general procedure for computing each is presented. The effect of the parameter p is examined in section 4. In section 5 numerical examples are given that compare the performance of preconditioners from both classes in a simple iterative method.

2. Background and Motivation

In this section we describe the use of block matrix factorization to compute preconditioners.

We consider the problem

$$Ax = b \tag{2.1}$$

where the matrix A has the form

$$\begin{bmatrix} B_1 & -F_1 & & & & \\ -E_1 & B_2 & -F_2 & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & -E_{n-2} & B_{n-1} & -F_{n-1} \\ & & & & -E_{n-1} & B_n \end{bmatrix},$$

and the matrices B_i, E_i and F_i are n by n. The n^2 by n^2 matrix A is said to be a block tridiagonal matrix. In addition, we assume that B_i, E_i and F_i are banded matrices. Such systems arise frequently in solving two-dimensional PDEs when n grid points are used in each direction to discretize the problem. Throughout this work, F denotes the n^2 by n^2 matrix that consists of the blocks F_i of A with all other components set to zero. The n^2 by n^2 matrices B and E are defined similarly. We note that in many important problems, A is an m-matrix (i.e., off diagonal elements of A are nonpositive, A is invertible and $A^{-1} \geq 0$).

Assume that the diagonal blocks B_i are nonsingular. We define the reduced n by n diagonal blocks G_i as

$$\begin{aligned} G_1 &= B_1, \\ G_i &= B_i - E_{i-1}G_{i-1}^{-1}F_{i-1}, \quad i=2,\ldots,n. \end{aligned} \tag{2.2}$$

The reduced diagonal blocks are produced by the forward course of block Gaussian elimination applied to A. It is not difficult to show that if A is a diagonally dominant m-matrix, then the matrices in (2.2) are well-defined and are also diagonally dominant m-matrices (see [7] and [3], Lemma 1.). Throughout this work G will denote the n^2 by n^2 block diagonal matrix whose i-th diagonal block is G_i.

We define the matrix M as

$$M = (G - E)\, G^{-1}\, (G - F). \tag{2.3}$$

A straightforward calculation shows that $M^{-1}A = I$ where I denotes the identity matrix.

In order to solve (2.1), the scheme (2.2) could be implemented as follows. Given G_{i-1}, calculate the triangular factors L_{i-1} and U_{i-1} [6] such that

$$G_{i-1} = L_{i-1}U_{i-1} .$$

Use these factors to compute $G_{i-1}^{-1} F_{i-1}$ and complete the calculation of G_i as in (2.2). Save the triangular factors L_{i-1} and U_{i-1}. The solution of (2.1) is given by

$$x = (G - F)^{-1}\, G\, (G - E)^{-1}\, b . \tag{2.4}$$

Writing the vectors b^t, x^t and q^t as $(b_1^t, b_2^t, \ldots, b_n^t)$, $(x_1^t, \ldots, x_n^t)$ and $(q_1^t, \ldots, q_n^t)$, respectively, where b_i, x_i and q_i are n-vectors, (2.4) can be computed by calculating

$$\begin{aligned} q_1 &= b_1, \\ q_j &= b_j + E_{j-1}\, q_{j-1}, \quad j=2,\ldots,n, \end{aligned} \tag{2.5a}$$

and solving

$$(L_n U_n)\, x_n = q_n \,,$$

(2.5b)

$$(L_j U_j)\, x_j = q_j + F_j\, x_{j+1},\ j=n-1,\ldots,1 \,.$$

Although (2.2) and (2.4) provide a direct solution method that fully exploits problem structure, it requires a considerable amount of computation and storage. Note that the G_i are usually dense matrices even if the B_i are banded. Thus storing $L_i U_i$ for $i=1,2,\ldots,n$ requires n^3 words of storage. Also, calculating G_i requires $O(n^3)$ floating point operations, and the total cost of (2.2) is $O(n^4)$ operations. For many problems these computational costs can be avoided. Observe that obtaining the final solution by (2.5) requires $O(n^3)$ operations.

The basic idea behind incomplete block factorization schemes is to approximate the G_i with matrices $\tilde{G}_i$ that are inexpensive to compute and require less storage. Instead of (2.3) we could form the matrix

(2.6a) $$\tilde{M} = (\tilde{G} - E)\, \tilde{G}^{-1}\, (\tilde{G} - F)$$

where $\tilde{G}$ is an n^2 by n^2 block diagonal matrix whose diagonal blocks are the $\tilde{G}_i$. In this case

(2.6b) $$x_0 = \tilde{M}^{-1} b$$

is only an approximation to the solution of (2.1). Also, $\tilde{M}^{-1}A \neq I$. However, if $\tilde{G}_i$ is a good approximation to G_i, we would expect that $\tilde{M}^{-1}$ is a good approximation to the inverse of A, and also that $|\rho(\tilde{M}^{-1}A)-1|$ is small where ρ denotes the spectral radius. The matrix $\tilde{M}^{-1}$ can be viewed as a preconditioner for the system (2.1).

$\tilde{M}$ can be used to improve the performance of various iterative methods. If A is symmetric and positive definite, incomplete Cholesky factorization can be used to compute the $\tilde{G}_i$. The resulting matrix $\tilde{M}$ is symmetric and can be used as a preconditioner for the conjugate gradient method [5]. For

nonsymmetric problems $\tilde{M}$ can be used in an iterative method based on splitting [10]. We recall several fundamental facts regarding these methods.

Given the matrix A, consider the splitting

$$A = \tilde{M} - N$$

where $\tilde{M}$ is a nonsingular matrix. The matrix N is referred to as the defect matrix. The splitting suggests the iteration

$$\begin{aligned} x_{k+1} &= x_k + \tilde{M}^{-1} r_k, \\ r_k &= b - Ax_k . \end{aligned} \tag{2.7}$$

If $\epsilon_k = x_k - x$ where x denotes the solution of (2.1), and x_0 is the starting point for the iteration, then

$$\epsilon_{k+1} = (I - \tilde{M}^{-1}A)^k \epsilon_0 . \tag{2.8}$$

Thus convergence requires that $\rho(I-\tilde{M}^{-1}A) < 1$. The better $\tilde{M}^{-1}$ approximates A^{-1}, the faster (2.7) converges.

One possible way to approximate G_i is to simply ignore the term $E_{i-1}G_{i-1}^{-1}F_{i-1}$ and set $\tilde{G}_i = B_i$. For this choice (2.6a) is

$$M_{SSOR} = (B - E)^{-1} B (B - F)^{-1} \tag{2.9}$$

which is the SSOR matrix [11] with relaxation factor set to unity. Since the B_i in (2.9) are all banded matrices, the basic iterative step (2.7) is computed using very little additional storage and, approximately n^2p operations where p is the number of super- and sub-diagonals of B. Procedures based on incomplete block factorization that attempt to compute a better approximation for the G_i can be viewed as more highly implicit methods than the SSOR method.

Diagonal dominance plays an important role in these methods and this is illustrated in the following simple example. Consider the usual second order centered difference discretization of the two-dimensional Laplace equation, $\Delta u = 0$. The discretized system can be written in the form (2.1) where $E_i = I$, $F_i = I$ and B_i is a tridiagonal matrix with 4 on the diagonal, -1 on the super- and sub-diagonals. Although B_i^{-1} is dense, components in a given row decrease by about an order of magnitude every two columns from the main diagonal. In this example the G_i in (2.2) are also diagonally dominant and components of the G_i^{-1} decrease in magnitude away from the main diagonal. Thus, one is led to consider modifications to (2.2) that approximate G_i by ignoring outer super- and sub-diagonals.

3. Approximations for Reduced Diagonal Blocks

In this section we describe two methods for computing approximations to the blocks G_i in (2.4). The two methods lead to two different classes of preconditioners. The preconditioners in each class are parameterized by an integer parameter p. For smaller values of p, preconditioners in the first class have low storage requirements; those in the second class have low storage and computational requirements.

Method 1 is the procedure for computing the first class of preconditioners. In order to simplify the description of the method we assume that B_i, E_i and F_i have no more than p super- and sub-diagonals. The notation e_s is used to denote a vector whose only nonzero component is the s-th component which is 1.

Method 1

1. Select an integer p, $1 \le p \le n$.

   Set $G_1 = B_1$.

   Repeat steps 2, 3 and 4 for $i=2,3,\ldots,n$.

   2. Factor G_{i-1} to obtain L_{i-1} and U_{i-1}.

3. Compute $X = G_{i-1}^{-1} F_{i-1}$ by solving

$$(L_{i-1}U_{i-1})\, Xe_s = F_{i-1}e_s$$

for $s=1,2,\ldots,n$. Set $G_i = B_i - E_{i-1}X$.

4. Form the truncated factors $\tilde{L}_{i-1}$ and $\tilde{U}_{i-1}$ by discarding all but the diagonal and p sub- and super-diagonals of L_{i-1} and U_{i-1}, respectively.

5. Compute $\tilde{L}_n$ and $\tilde{U}_n$ for G_n.

In order to solve (2.1), let $\tilde{G}_i = \tilde{L}_i\, \tilde{U}_i$ be an approximation to the i-th reduced block. Use the matrices $\tilde{G}_i$ to form the matrix $\tilde{M}$ as in (2.6a). $\tilde{M}$ can then be used in iteration (2.7) to obtain the solution to (2.1). It can also be used to precondition (2.1).

The LINPACK routine SGEFA [6] is used in our implementation of Method 1 to compute L_i and U_i in step 2. It employs partial pivoting for stability. If at some stage in constructing L_i and U_i SGEFA chooses a nondiagonal pivot, it interchanges the current row and selected pivot row. When SGEFA terminates, the lower triangle of the matrix contains L_i and the upper triangle contains U_i.

In step 4 all but the diagonal and p super- and sub-diagonals are discarded to form $\tilde{L}_i$ and $\tilde{U}_i$. Motivation for this was provided in section 2 when A is a diagonally dominant matrix. For many problems row interchanges performed by SGEFA bring elements of larger magnitudes into the diagonal positions of U_i. Also, these interchanges tend to reduce the magnitudes of elements in the outer super- and sub-diagonals of both factors.

Iteration (2.7) requires the vector $\tilde{M}^{-1}r$. To compute this vector our implementation of (2.7), which is used in section 5, employs (2.5a) and (2.5b) where the truncated factors $\tilde{L}_i$ and $\tilde{U}_i$ are used in (2.5b). We have modified LINPACK routine SGESL [6] so it can use the truncated factors to

obtain a solution. The modified routine exploits the banded structure of $\tilde{L}_i$ and $\tilde{U}_i$ to reduce the total cost of solving the systems in (2.5b) to approximately $(3p+1)n^2$ operations. The unmodified version of SGESL is used in step 3 to compute Xe_s.

Method 1 offers no savings in computation over the direct method (2.2), but total storage for all the matrices $\tilde{L}_i$ and $\tilde{U}_i$ is only $(2p+1)n^2$. Note that the method computes the same blocks G_i that are produced in (2.2). Errors arise when the truncated factors of G_i are used to form $\tilde{G}$. One would expect the inverse of $\tilde{M}$ to be a good approximation to A^{-1} and this is certainly evidenced in the numerical tests presented in section 5.

The next method, Method 2, usually requires considerably less computation than Method 1. For the remainder of this section, it is assumed that $\tilde{G}_i$ is computed by Method 2. For a matrix Λ, the notation $[\Lambda]_p$ denotes the banded matrix formed from the diagonal, and p sub- and p super-diagonals of Λ.

<u>Method 2</u>

1. Select an integer p, $1 \le p \le n$. Set $\tilde{G}_1$ to the diagonal and p super- and sub-diagonals of B_1.

   Repeat steps 2, 3 and 4 for i=2,...,n.

   2. Factor $\tilde{G}_{i-1}$ to obtain $\tilde{L}_{i-1}$ and $\tilde{U}_{i-1}$.

   3. Compute $X = (\tilde{G}_{i-1})^{-1} F_{i-1}$ by solving $(\tilde{L}_{i-1}\tilde{U}_{i-1}) X = F_{i-1}$.

   4. Compute $\tilde{G}_i = B_i - [E_{i-1}X]_p$.

5. Compute $\tilde{L}_n$ and $\tilde{U}_n$.

As in Method 1 the matrices $\tilde{G}_i$ can be used to form $\tilde{M}$ by (2.6a). $\tilde{M}$ can then be used in (2.7) to solve (2.1) or as a preconditioner for (2.1).

In step 2 $\tilde{L}_{i-1}$ and $\tilde{U}_{i-1}$ are just the factors of a banded matrix with p super- and sub-diagonals. Partial pivoting is used in constructing these factors and due to possible row interchanges performed in selecting pivots, $\tilde{U}_{i-1}$ can have up to 2p nonzero super-diagonals. Thus storage for $\tilde{G}_{i-1}$ in factored form requires (3p+1)n words.

In step 3 the factors are used to obtain X by solving a matrix equation. This is the computationally intensive step in method 2. The following is an outline of our implementation of this step.

Recall that $\tilde{U}_{i-1}$ is constructed by applying a sequence of row interchanges and elementary row operations to $\tilde{G}_{i-1}$. More specifically,

$$\tilde{U}_{i-1} = \left(\prod_{k=1}^{n-1} R_k \, P_k\right) \tilde{G}_{i-1}$$

where R_k and P_k are n by n matrices. Multiplication by P_k interchanges rows, and multiplication by R_k performs an elementary row operation. R_k is a matrix with one on the main diagonal, and all other entries are zero with the exception of the p positions below the main diagonal in the k-th column. These entries are the multipliers found in the k-th column of $\tilde{L}_{i-1}$.

In order to compute X, the matrix

$$T = \left(\prod_{k=1}^{n-1} R_k \, P_k\right) F_{i-1}$$

is formed, and then the system

$$\tilde{U}_{i-1} \, X = T$$

is solved. Both calculations can be implemented by making calls to the basic linear algebra subroutine SAXPY [6][1] with vectors of length n.

[1]Recall that given vectors x and y and the scalar a, SAXPY forms ax+y.

We have found this implementation performs well on both scalar and vector machines. Since n is the number of grid points in one direction, n is large enough in most problems so that excessive loop overhead is avoided, and vector performance rates are achieved on vector machines.

In step 4 only the components on the diagonal and the p super- and sub-diagonals of $E_{i-1}X$ are computed.

Our current implementation requires approximately $(3p+1)n^3$ operations to compute all the $\tilde{G}_i$ in factored form. If pivots were restricted to the diagonal, this would be reduced to about $(2p+1)n^3$ operations. If, in addition, the banded structure of F_i were exploited in forming T, the operation count could be reduced further. For example, if F_i has a symmetric structure and a bandwidth no greater than 2p+1, the operation count can be reduced to about $(1.5p+1)n^3$.

Both methods employ partial pivoting for stability. We feel this is essential in a method that will be used to solve a wide range of problems. It also results in codes that are easier to use. For example, in solving multicomponent PDE systems it eliminates the need for a particular variable ordering that ensures diagonal pivoting is stable.

4. Analysis of the Defect Matrix and Convergence

Methods 1 and 2 calculate a matrix $\tilde{M}$ that approximates A. Both methods require the selection of a parameter p that determines the accuracy of the approximation. In either method the defect matrix is $N = \tilde{M} - A$. In this section we analyze the behavior of the defect matrix and prove the convergence of Method 2.

The following is a well-known convergence result for (2.7). Assume that A is an invertible matrix such that $A^{-1} \geq 0$. If in the splitting of A $\tilde{M}$ is nonsingular, $\tilde{M}^{-1} \geq 0$ and $N \geq 0$, then

$$\rho\,(\tilde{M}^{-1}N) \quad < \quad 1,$$

and it follows from (2.8) that iteration (2.7) converges [10, Theorem 3.13]. Obviously, a splitting where components of N are small in magnitude is desirable. Throughout this section, we assume that A in (2.1) is a diagonally dominant m-matrix; therefore, $A^{-1} \geq 0$.

In this section a superscript p is used to indicate that the matrix depends on the value of p used in Method 1 or Method 2. For example, G_i^p is used to denote an approximation to G_i. It will be pointed out whether G_i^p is computed by Method 1 or Method 2. Regardless of the method used, it is assumed that in constructing G_i^p pivots are taken on the diagonal. It will also be assumed, unless otherwise indicated, that $p \geq p_0$ where p_0 is the largest half bandwidth of the matrices E_i, F_i and B_i.

The defect matrix N^p is the block diagonal matrix

$$
\begin{aligned}
N_1^p &= G_1^p - B_1, \\
N_i^p &= G_i^p - B_i + E_{i-1}(G_{i-1}^p)^{-1} F_{i-1}, \quad i=2,\ldots,n .
\end{aligned}
\tag{4.1}
$$

In Method 1,

$$G_i^p = L_i^p U_i^p$$

where L_i^p and U_i^p are the truncated factors of

$$B_i - E_{i-1}G_{i-1}^{-1}F_{i-1} .$$

In Method 2 G_i^p is the band matrix computed in step 4. Note that if $p = n$, N^p is the zero matrix for both methods. We require several preliminary results in order to describe the behavior of N^p. Throughout this section if Λ is a matrix, $\Lambda_{i,j}$ denotes the i,j-th element.

Preliminary Results

Lemma 4.1 Let G be an n by n diagonally dominant m-matrix and let L and U be triangular factors for G. Assume that pivots are taken from the diagonal in constructing L and U. Then

$$L_{j,k} \le 0 \text{ if } j \ne k, \tag{4.2a}$$

$$U_{j,k} \le 0 \text{ if } j \ne k, \tag{4.2b}$$

$$U_{j,j} \ge 0 \tag{4.2c}$$

for $j=1,2,\ldots,n$ and $k = 1,2,\ldots,n$.

Proof: Inequalities (4.2b) and (4.2c) hold since a basic step of Gaussian elimination on a diagonally dominant m-matrix results in a diagonally dominant m-matrix [7]. Let $\bar{U}_{j,k}$ and $\bar{U}_{j,j}$ be elements in the reduced matrix after j-1 steps of Gaussian elimination. If $j \ne k$, then $\bar{U}_{j,k} \le 0$ and $\bar{U}_{j,j} > 0$. Since $L_{k,j} = \bar{U}_{k,j}/\bar{U}_{j,j}$, (4.2a) follows.■

In the next lemma, $G^p = L^p U^p$ where L and U are the triangular factors of G and $L^p = [L]_p$ and $U^p = [U]_p$.

Lemma 4.2 Let E, F and G be n by n matrices and assume that $E \ge 0$, $F \ge 0$ and that G is a diagonally dominant m-matrix. If p_1 and p_2 are positive integers less than or equal n and $p_1 < p_2$, then

$$0 \le E(G^{p_1})^{-1}F \le E(G^{p_2})^{-1}F \quad . \tag{4.3}$$

Proof: For $1 \le p \le n$,

$$E(G^p)^{-1}F = E(U^p)^{-1}\,(L^p)^{-1}\,F \quad . \tag{4.4}$$

For a given unit vector e_s, consider the solution of the system

$$L^p x = Fe_s \ , \tag{4.5}$$

which is

$$x_1 = (1/L^p_{1,1})\ F_{1,s} \ ,$$

$$x_i = F_{i,s} - \sum_{k=1}^{i-1} L^p_{i,k}\ x_k, \ i=2,\ldots,n \ .$$

Since G is an m-matrix, Lemma (4.1) shows that $L^p_{i,k} \le 0$ if $i \ne k$. Since $L^p_{1,1} = 1$ and $F \ge 0$, it follows that $x_1 \ge 0$. Also, if $p_1 < p_2$, then the definition of L^p implies that

$$x^{p_1} \le x^{p_2}$$

where x^{p_1} and x^{p_2} denote the corresponding solutions of (4.5). From the preceding inequality, we have

$$(L^{p_1})^{-1}\ F \le (L^{p_2})^{-1}\ F \ , \tag{4.6a}$$

and similar arguments show that

$$E\ (U^{p_1})^{-1} \le E\ (U^{p_2})^{-1} \ . \tag{4.6b}$$

Thus, inequality (4.3) follows from (4.4) and (4.6).■

Note that if $E \ne 0$ and $F \ne 0$, then the inequalities in (4.6) are strict if $L_{i,j} \ne 0$ for $i \ge j$ and $U_{i,j} \ne 0$ for $i \le j$. Under these conditions, it follows that the inequalities in (4.3) are also strict.

<u>Lemma 4.3</u> Assume that G is an n by n diagonally dominant m-matrix, and L and U are its triangular factors. Then for positive integers $p_1 < p_2 \le n$,

$$L^{p_1} U^{p_1} \le [L^{p_2} U^{p_2}]_{p_1} .$$

Proof. The proof is a straightforward consequence of Lemma 4.1.■

Behavior of N^p in Method 1

Let A in (2.1) be a diagonally dominant m-matrix and assume the G_i^p are computed by Method 1. For integers p_1 and p_2, $1 \le p_1 < p_2 \le n$, it follows from Lemma 4.2 that

$$B_i - E_{i-1} (G_{i-1}^{p_2})^{-1} F_{i-1} \le B_i - E_{i-1}(G_{i-1}^{p_1})^{-1} F_{i-1}. \tag{4.7}$$

This inequality leads to the following theorem that describes the behavior of N_p for Method 1.

Theorem 4.1 Let N^p be defined by (4.1). If p_1 and p_2 are integers such that $1 \le p_1 < p_2 \le n$, then

$$[N^{p_1}]_{p_1} \le [N^{p_2}]_{p_1} \le 0 .$$

Proof. Since G_i is a diagonally dominant m-matrix, Lemma 4.3 shows $G_i^{p_1} \le [G_i^{p_2}]_{p_1}$. This inequality and inequality (4.7) show that

$$G_i^{p_1} - B_i + E_{i-1}(G_{i-1}^{p_1})^{-1} F_{i-1} \le [G_i^{p_2}]_{p_1} - B_i + E_{i-1}(G_{i-1}^{p_2})^{-1} F_{i-1} ,$$

and

$$[N_i^{p_1}]_{p_1} \le [N_i^{p_2}]_{p_1} ;$$

hence, the first inequality is true. The second follows since N_i^n is the zero matrix.■

Theorem 4.1 shows that for Method 1, if $p_1 < p_2$ entries in the p_1 super- and sub-diagonals of N^{p_1} are negative. Furthermore, entries in the p_1

super- and sub-diagonals of N^{p_2} have magnitudes less or equal to those of N^{p_1}. However, in N^{p_1} super- and sub-digaonals more than p_1 columns away from the main diagonal can have positive entries. In addition, in N^{p_2} entries in sub- and super-diagonals more than p_2 columns away from the main diagonal can be larger than those in N^{p_1}. This occurs, for example, if Method 1 is used to compute the approximate reduced diagonal blocks for the diffusion problem in section 5.

Behavior of N^p in Method 2

Let G_i^p denote the reduced blocks computed in Method 2. The diagonal blocks of N^p are N_1^p, which is the zero matrix, and

$$N_i^p = -\,[E_{i-1}(G_{i-1}^p)^{-1}\,F_{i-1}]_p + E_{i-1}(G_{i-1}^p)^{-1}\,F_{i-1},\quad i=2,\ldots,n\ . \tag{4.9}$$

It is not difficult to show that the G_i^p computed by Method 2 are diagonally dominant m-matrices. Recall that we have assumed the half bandwidth of each B_i is no greater than p_0 and $p \geq p_0$. Since $G_1^p = B_1$, it is a diagonally dominant m-matrix. Now, $G_2^p = [\Lambda]_p$ where $\Lambda = B_2 - E_1(G_1^p)^{-1}F_1$. Since Λ is obtained by Gaussian elimination on a diagonally dominant m-matrix, it is also a diagonally dominant m-matrix. Moreover, so is $[\Lambda]_p$ since discarding outer super- and sub-diagonals of a matrix preserves the diagonally dominant m-matrix property. Hence, G_2^p is a diagonally dominant m-matrix. This argument can be repeated to show that all the matrices G_i^p are diagonally dominant m-matrices.

The following lemma provides several results needed to describe the behavior of N^p for Method 2. It is also required for the convergence of Method 2 which is discussed later in this section.

Lemma 4.4 Let the G_i^p be computed as in Method 2. If p_1 and p_2 are integers such that $p_0 \leq p_1 < p_2 \leq n$, then for $i=1,2,\ldots,n$

(4.10a) $$G_i^{p_2} \le G_i^{p_1} \quad ,$$

(4.10b) $$0 \le (G_i^{p_1})^{-1} \le (G_i^{p_2})^{-1} \quad ,$$

(4.10c) $$0 \le E_i \, (G_i^{p_1})^{-1} \, F_i \le E_i \, (G_i^{p_2})^{-1} \, F_i \quad .$$

<u>Proof.</u> The proof is by induction. Since $G_1^p = B_1$ for any $p \ge p_0$, the three inequalities hold for $i = 1$. Assume that (4.10a) holds for some i. Since G_i^p is an m-matrix, (4.10b) follows. Since $E_i \ge 0$ and $F_i \ge 0$, (4.10c) follows. From (4.10c), we have that

$$B_{i+1} - [E_i(G_i^{p_1})^{-1} F_i]_{p_1} \ge B_{i+1} - [E_i(G_i^{p_2})^{-1} F]_{p_2} \; ;$$

hence, (4.10a) holds for i+1. In addition, (4.10b) and (4.10c) are true for i+1.■

Note that (4.9) is just the matrix $E_i(G_i^p)^{-1} F_i$ with the diagonal and the p super- and sub-diagonals set to zero. If $p_1 < p_2$, then the diagonal and p_2 sub- and super-diagonals of N^{p_2} are zero while only the diagonal and p_1 sub- and super-diagonals of N^{p_1} are zero. However, (4.10c) and (4.9) show that for the outer nonzero diagonals of N^{p_1} and N^{p_2}, elements of N^{p_2} are greater than those of N^{p_1}.

The preceding discussion indicates that preconditioners computed by Method 2 may not be effective for certain problems unless a large value of p is used. For problems that are strongly diagonally dominant, the magnitudes of elements on the outer sub- and super-diagonals of N^p are small relative to elements on the diagonal of M^p. However, in less diagonally dominant problems, this will not be the case unless p is chosen large enough so that these elements are 0.

It is very interesting to note the difference in the defect matrices that result from Method 1 and Method 2. Method 1 is much closer to a direct method for computing the reduced diagonal blocks (2.2) and provides more accurate approximations. Yet, the defect matrix for Method 1 need not be nonnegative. This is atypical of most suggested incomplete block factorization methods which usually result in a regular splitting.

Convergence of Method 2

Since the defect matrix that results in Method 1 can have both positive and negative components, the classical convergence theory for regular splittings [10] does not apply. We do not analyze the convergence of this method here. Method 2, however, yields a regular splitting. Convergence follows from general convergence results presented in [3, Theorem 1 and Corollary 1]. For the convenience of the reader, we restate these results here as Theorem 4.2 using our notation. Recall that the matrix A is assumed to have the block tridiagonal form (2.1).

Theorem 4.2 Consider the splitting $A = \tilde{M} - N$ where $\tilde{M} = (\tilde{G} - E)\,\tilde{G}^{-1}(\tilde{G} - F)$ and $N = \tilde{G} + E\,\tilde{G}^{-1}F - B$. If $A^{-1} \geq 0$, $\tilde{G}^{-1} \geq 0$ and $N \geq 0$, then $\rho(\tilde{M}^{-1}N) < 1$ and iteration (2.7) converges.

The next theorem provides a convergence result for Method 2.

Theorem 4.3 Assume that A is a diagonally dominant m-matrix and let p be an integer, $p_0 \leq p \leq n$. Assume that G_i^p are computed by Method 2 and are used to form the matrices G^p, M^p and N^p. Then for the splitting $A = M^p - N^p$, $\rho((M^p)^{-1}N^p) < 1$ and iteration (2.7) converges with $\tilde{M} = M^p$.

Proof. Equality (4.9) and inequality (4.10c) of Lemma 4.4 show that $N_i^p \geq 0$ for $i = 1,2,\ldots,n$ and $p_0 \leq p \leq n$, hence, $N^p \geq 0$. By inequality (4.10b) of Lemma 4.4 $(G_i^p)^{-1} \geq 0$, for $i=1,2,\ldots,n$ and $p_0 \leq p \leq n$, thus, $(G^p)^{-1} \geq 0$. Since $A^{-1} \geq 0$, Theorem (4.3) follows by applying Theorem (4.2) with $N = N^p$ and $\tilde{G} = G^p$.■

5. Numerical Experiments

In this section we examine the performance of (2.7) on two test problems where $\tilde{M}$ is computed using Method 1 (M1) and Method 2 (M2). We also present results obtained using M_{SSOR} (2.10) for purposes of comparison.

The test problems are the following simple two-dimensional diffusion problem (Laplace equation) and a two-dimensional diffusion-convection problem.

1. Diffusion problem: $\Delta u = 0$ if $(x,y) \in \Omega$ where $\Omega = (0,1) \times (0,1)$ and $u = 1$ on $\partial\Omega$.

2. Diffusion-convection problem: $-\epsilon\Delta u + \alpha u_x + \beta u_y = S(x,y)$ if $(x,y) \in \Omega$, $u = 0$ on $\partial\Omega$.

For problem 2, it was assumed that the solution was

$$\sin(\pi x)\sin(\pi y)\, e^{\frac{1}{\sqrt{\epsilon}}} \left(e^{\frac{x-1}{\sqrt{\epsilon}}} + e^{-y} + e^{y-1}\right) .$$

$S(x,y)$ was obtained by applying the partial differential operator to the solution. In the experiments $\epsilon = .01$, $\alpha = .2$ and $\beta = .2$.

Problem 2 was chosen because it requires the solution of a nonsymmetric linear system. In order to handle convection in this problem, an upwind discretization scheme was used. Problem 1 is commonly used as a test problem and was discretized using the usual centered second order differences.

M1 and M2 were used to solve problems 1 and 2 with 15 and 31 interior grid points in both the x and y directions. The grid spacing in each direction is $(n+1)^{-1}$. The tables in Figures 5.1 and 5.2 give the number of iterations of (2.7) required to solve each problem for a given value of the parameter p. The number of iterations required by SSOR is also given.

Problem 1

n=15	(value of p) 1	2	3	6	
M1	16	10	7	4	
M2	23	14	9	5	SSOR = 81 (iterations)

n=31	1	2	3	6	
M1	49	27	18	9	
M2	79	44	28	12	SSOR = 294

Figure 5.1

Problem 2

n=15	(value of p) 1	2	3	6	
M1	14	9	6	4	
M2	19	12	8	4	SSOR = 63

n=31	1	2	3	6	
M1	39	23	15	7	
M2	61	36	23	10	SSOR = 201

Figure 5.2

For each test an initial guess was obtained using (2.6a) and (2.6b) where $\tilde{G}$ was computed using the appropriate method - M1, M2 or SSOR. An iteration was terminated when the inequalities

$$(r_k^t \tilde{M}^{-1} r_k)^{1/2} \le 10^{-5}$$

and

$$\max_i \; | (\tilde{M}^{-1} r_k)^t e_i | \le 10^{-5} | x_k^t e_i |$$

where $x_k^t e_i \neq 0$ were both satisfied. The method for determining an initial approximation and the stopping criteria are similar to those used in [3].

In each problem and for all values of n and p, M1 requires the least number of iterations. The savings are most pronounced in problems 1 and 2 for the case n = 31 where the problems are more highly ill-conditioned. Note that both M1 and M2 require fewer iterations when larger values of p are used to compute $\tilde{M}$. The results obtained for the diffusion problem using M2 with p = 1 are identical to those in [3].

A significant portion of the costs of solving a problem with M1 or M2 is the cost of computing $\tilde{M}$. Since M1 requires $O(n^4)$ operations to compute $\tilde{M}$, it is the most expensive method for solving the two test problems despite the fact it requires the least number of iterations. Recall that M2 requires at least $(1.5p+1)n^3$ operations to compute $\tilde{M}$. In the SSOR method, iterations of (2.7) are the main costs.

The method of choice for solving either of the test problems is M2 with p = 1 or p = 2. This method requires the least total work to solve problem 1 or 2 in either the case that n = 15 or n = 31. For example, consider problem 1 in the case n = 31. SSOR requires approximately $1180n^2$ operations to solve the problem. M2 requires approximately $424n^2$ operations if p = 1, and $494n^2$ if p = 2. It should be noted that the diagonal blocks B_i are strongly diagonally dominant in both problems; therefore, SSOR performs reasonably well.

Figure 5.3 shows the bounds on the spectral radius of $\tilde{M}^{-1}A$ where A is the coefficient matrix in problem 1 for n = 31. The two columns give the

results obtained when $\tilde{M}$ is computed by M1 or M2 using several different values of p. The bounds on $\rho(A)$ are $.019 \le \rho(A) \le 7.98$.

p	M1	M2
1	[.22, 1.18]	[.14, 1.07]
2	[.37, 1.18]	[.24, 1.08]
3	[.51, 1.16]	[.36, 1.09]
6	[.80, 1.10]	[.67, 1.07]

Figure (5.3)

M1 does a slightly better job of reducing the spectral radius of $\tilde{M}^{-1}A$ for all values of p. For p = 2 and 3 there is a slight increase in the upper bound on the spectral radius when M is computed by M2. Note that for M1, the upper bound on the spectral radius does not increase as in M2.

6. References

1. O. Axelsson, "A Survey of Preconditioned Iterative Methods for Linear Systems of Algebraic Equations," Rept. 8428, Department of Mathematics, Catholic University, Toernoorveld, Netherlands, 1984.
2. O. Axelsson, "A General Incomplete Block-Matrix Factorization Method," Lin. Alg. Appls., 74 (1986), pp. 179-190.
3. O. Axelsson, S. Brinkkemper and V. P. Il'in, "On Some Versions of Incomplete Block-Matrix Factorization Methods," Lin. Alg. Appls., 58 (1984), pp. 3-15.
4. S. Brinkkemper, "A Class of Incomplete Block-Factorization Methods," Rept. 71, Department of Informatics, Nymejen University, Toernoorveld, The Netherlands, 1985.

5. P. Concus, G. H. Golub and G. Meurant, "Block Preconditioning for the Conjugate Gradient Method," SIAM J. Sci. Stat. Comput., 6 (1985), pp. 220-252.

6. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, LINPACK Users' Guide, SIAM, Philadelphia, 1979.

7. Ky Fan, "Note on M Matrices," Quart. J. Math Oxford (2), 11 (1960), pp. 43-49.

8. D. S. Kershaw, The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations, J. of Comput. Phys. 26, (1978), pp. 43-65.

9. J. A. Meijerink and H. A. 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.

10. R. S. Varga, Matrix Iterative Analysis, Prentice Hall, Englewood Cliffs, NJ, 1962.

11. D. M. Young, Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.

Distribution:

0200	G. J. Simmons
1400	E. H. Barsis
1410	P. J. Eicker
1412	G. S. Davidson
1413	G. G. Weigand
1413	R. E. Benner
1420	R. J. Thompson
1422	R. C. Allen, Jr.
1422	D. E. Amos
1422	L. S. Baca (10)
1422	P. B. Bailey
1422	L. A. Romero
1422	D. E. Salane (30)
1422	D. E. Womble
1511	N. E. Bixler
1512	M. R. Baer
1513	D. W. Larson
1520	C. W. Peterson
1533	M. E. Kipp
2112	H. A. Watts
2113	J. A. Wisniewski
2643	R. J. Hanson
2643	A. H. Treadway
2643	E. L. Walton
2645	L. A. Bertram
2645	W. H. Vandevender
6233	T. M. Gerlach
6400	D. J. McCloskey
6512	P. J. McDaniel
7112	F. R. Biggs
8024	P. W. Dean
8245	R. J. Kee, Jr.
3141	S. A. Landenberger (5)
3151	W. L. Garner (3) For DOE/OSTI
3154-1	C. H. Dalin (28) For DOE/OSTI