

Preconditioning By Incomplete Block Cyclic Reduction

By Garry Rodrigue* and Donald Wolitzer

Abstract. Iterative methods for solving linear systems arising from the discretization of elliptic/parabolic partial differential equations require the use of preconditioners to gain increased rates of convergence. Preconditioners arising from incomplete factorizations have been shown to be very effective. However, the recursiveness of these methods can offset these gains somewhat on a vector processor. In this paper, an incomplete factorization based on block cyclic reduction is developed. It is shown that under block diagonal dominance conditions the off-diagonal terms decay quadratically, yielding more effective algorithms.

Introduction. Iterative methods are frequently used for solving the linear systems arising from the differencing of 2-dimensional elliptic or parabolic partial differential equations, [19]. These systems are often very large and take on the structure

$$(1.1) \quad Ax = \begin{bmatrix} A_1 & B_1' & & & & \\ B_1 & A_2 & B_2' & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & B_{L-1}' & \\ & & & & B_{L-1} & A_L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_L \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{bmatrix} = y,$$

where A_i is tridiagonal and symmetric ($i = 1, 2, \dots, L$) and B_i is tridiagonal ($i = 1, 2, \dots, L - 1$). Each block is order K . x_i and y_i are vectors of length K ($i = 1, \dots, L$). K and L refer to the number of discrete x and y coordinates respectively on an orthogonal numerical grid, [16]. The matrix A is assumed symmetric and positive definite

Most iterative methods for solving $Ax = y$ involve approximating the matrix A by a matrix M (called a preconditioner or a splitting matrix), and at each iteration, solving a system of the form $Mz = d$. For example, first order iterative methods such as the Jacobi, Gauss-Seidel, or SOR iterative methods (cf. [21]) can be put in a general form

$$(1.2) \quad Mx^{(k+1)} = Nx^{(k)} + y,$$

where $x^{(0)}$ is given and $A = M - N$. Second order iterative methods are generally regarded as accelerations of first order methods and can be put in general form as

$$(1.3) \quad \begin{aligned} Mz^{(k)} &= y - Ax^{(k)}, \\ x^{(k+1)} &= x^{(k-1)} + \omega_{k+1} (\alpha_k z^{(k)} + x^{(k)} - x^{(k-1)}), \end{aligned}$$

Received September 23, 1982; revised April 25, 1983.

1980 *Mathematics Subject Classification*. Primary 65F10, 65N20.

* Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

where ω_{k+1} , α_k are scalars and $x^{(1)}$ and $x^{(0)}$ are given. The Chebyshev semi-iterative method and Richardson second order method are of this form (cf. [21]) as well as the generalized conjugate gradient method (cf. Concus et al., [2]). As can be seen from (1.2) and (1.3), with the exception of inverting M , iterative methods are highly parallel algorithms. This is where the problem lies. The matrix M can range from a simple diagonal matrix (e.g., Jacobi's method), where inverting M is a single vector division, to a complicated product of lower and upper triangular matrices (e.g. point-SSOR) where inverting M is a highly recursive process, cf. Young [21]. Unfortunately, the more recursive the inversion process, the faster the iterates $x^{(k)}$ converge to the solution of the system (1.1) and the less vectorizable the method.

Lately, taking M to be an incomplete Cholesky factorization of A has proven to be a very powerful choice for a preconditioner. Intuitively, the construction of M in this case is very straightforward. Since A is symmetric and positive definite, it admits a Cholesky factorization, $A = LDL^t$, where $L = (l_{ij})$ is a lower triangular matrix and $D = (d_{ij})$ is a diagonal matrix. These matrices are given by the recurrence formula:

$$(1.4) \quad \begin{aligned} &\text{For } i = 1, \dots, N, \\ &l_{ji} = a_{ji} - \sum_{m=1}^{i-1} l_{jm} l_{im} d_{mm}, \quad j = i, i+1, \dots, N, \\ &d_{ii} = l/l_{ii}, \end{aligned}$$

where $N = KL$. For an incomplete Cholesky factorization of A , the computation (1.4) is performed with the constraint that $l_{ji} = 0$, where the pair (j, i) is not in some matrix sparsity pattern that is to be maintained. A common practice is for the new lower triangular matrix L to have the same sparsity pattern as the original matrix A . Using this form of $M = LDL^t$ as a preconditioner has its origins in Varga [20]. However, the real power of this preconditioner came when Meijerink and van der Vorst, [15], coupled it with the generalized conjugate gradient method of Concus, Golub, and O'Leary, [2]. Kershaw, [10], demonstrated the superiority of this method over the conventional relaxation methods such as S.O.R. Since then, several papers have appeared in the literature dealing with different aspects of this method, e.g., Manteuffel, [14], Axelsson, [1], Gustafson, [6], Ehrel et al. [4] and to describe the contribution of each of them would indeed add considerable length to this paper. Consequently, we have elected to omit this discussion.

Resolving the incomplete Cholesky factorization in (1.3) by a back-substitution algorithm is a recursive process and can be very troublesome for efficient vector computing. Greenbaum and Rodrigue, [5], Jordan, [8], and Kershaw, [9], describe implementations of the back-substitution algorithm on vector-processors but these methods do not yield vectors with lengths large enough for efficient pipelining. This is due mainly to the dependence of the back-substitution process on the way the original factorization was set up in the first place i.e. the Gaussian Elimination method (1.4). Thus, to achieve a more vectorizable method, the entire factorization and substitution process must be re-examined.

Lambiotte and Voigt, [11], and Madsen and Rodrigue, [12], have shown how the odd-even and cyclic reduction algorithms are highly parallel methods for solving scalar tridiagonal systems on vector-processors. Heller, [7], extended these algorithms to block-tridiagonal systems such as (1.1). In this paper, we show how the

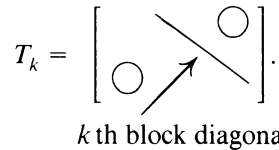
idea of incompleteness can be extended to these algorithms in such a way as to yield preconditioning systems that have a high potential for vector- as well as multi-processor computation. In addition, we capitalize on some of the results by Heller concerning the decay of the off-diagonal blocks in the odd-even or cyclic reduction process to yield an even higher degree of parallelism.

2. Incomplete Block Odd-Even Reduction. We begin by first discussing block odd-even reduction in terms of the diagonal notation of Madsen et al. [13]. This notation will be used throughout the paper and maintains the following conventions: For a given block matrix $T = (T_{ij})$,

$$T = \sum_{k=-(n-1)}^{n-1} T_k, \quad T_k = (T_{ij}^{(k)}),$$

$$T_{ij}^{(k)} = \begin{cases} T_{ij}, & \text{if } j = i + k, \\ 0, & \text{otherwise,} \end{cases}$$

i.e., T_k has only a nonzero k th block diagonal. Figuratively,



Sometimes we will be using the scalar entries of a block matrix. To avoid confusion we will use lower - case letters to denote the scalar diagonals. That is, if $T = (t_{ij})$ is a scalar matrix, then

$$T = \sum_{k=-(n-1)}^{n-1} t_k,$$

where the t_k are scalar diagonal matrices defined as above. The important consequence of this diagonal representation is that the diagonals of a matrix can be mathematically followed during the course of an algorithm. This is largely due to the following relationship for matrix multiplication: if $T = VW$, then

$$(2.0) \quad T_j = \sum_{j=k+m} V_k W_m.$$

With this notation in mind, the block odd-even reduction algorithm can be described as follows:

BLOCK ODD - EVEN REDUCTION

Decomposition: Let $A^{(0)} = A$ and p an integer such that $2^p \leq L < 2^{p+1}$. Then for $q = 0, 1, \dots, p - 1$,

$$(2.1) \quad Q^{(q)} = I - A^{(q)}(A_0^{(q)})^{-1},$$

$$(2.2) \quad A^{(q+1)} = (I + Q^{(q)})A^{(q)}.$$

Forward Substitution:

$$(2.3) \quad \prod_{q=0}^{p-1} (I + Q^{(q)})A^{(0)}x = A^{(p)}x = \prod_{q=0}^{p-1} (I + Q^{(q)})y = y^{(p)}.$$

Backward Substitution:

$$x = [A^{(p)}]^{-1}y^{(p)}.$$

Remark 1. Note that

$$A^{(q)} = A_{2^q}^{(q)} + A_{-2^q}^{(q)} + A_0^{(q)} \quad \text{and} \quad A^p = A_0^p,$$

i.e., the nonzero off-diagonal blocks of $A^{(q)}$ are precisely $A_{\pm 2^q}^{(q)}$, and A^p is a block-diagonal matrix.

Remark 2. If

$$(2.4) \quad H^{(q)} = I - (A_0^{(q)})^{-1}A^{(q)},$$

then

$$A^{(q+1)} = A^{(q)}(I + H^{(q)}).$$

Heller [7], analyzed the behavior of the l_1 - and l_∞ -norms of the off-diagonal blocks in the odd-even reduction algorithm and his results are as follows:

THEOREM 2.1. *Suppose $A_0^{(0)}$ is a nonsingular $N \times N$ matrix and $\|H^{(0)}\|_\infty < 1$. Then for $q = 0, 1, \dots, p$, the block odd-even reduction algorithm yields*

- (i) $A_0^{(q)}$ is a nonsingular $N \times N$ matrix;
- (ii) $\|H^{(q+1)}\|_\infty < \|(H^{(q)})^2\|_\infty$;
- (iii) If $\|Q^{(0)}\|_1 < 1$, then $\|Q^{(q+1)}\|_1 < \|(Q^{(q)})^2\|_1$.

Theorem 2.1(iii) states that the off-diagonal blocks of $Q^{(q)}$ decay quadratically to zero. Hence, block odd-even reduction can be prematurely terminated to obtain the following algorithm: Let

$$A^{(m)}x = \prod_{i=0}^{m-1} (I + Q^{(i)})A^{(0)}x = \prod_{i=0}^{m-1} (I + Q^{(i)})y = y^{(m)},$$

and define $x^{(m)} = (A_0^{(m)})^{-1}y^{(m)}$. Then, cf. Heller [7],

COROLLARY 2.1. *If $A_0^{(0)}$ is nonsingular and $\|H^{(0)}\|_\infty < 1$, then*

$$\frac{\|x - x^{(m)}\|_\infty}{\|x\|_\infty} \leq \|H^{(m)}\|_\infty,$$

where x is the solution of (1.1).

Since our goal is to examine the behavior of odd-even reduction on matrices arising from elliptic partial differential equations, it is natural to study the effect of the odd-even reduction algorithm on M -matrices. We call a matrix $T = (t_{ij})$ with $t_{ii} > 0$ and $t_{ij} \leq 0$, $i \neq j$, $i = 1, \dots, n$ an L -matrix. Further, an L -matrix is a nonsingular M -matrix if it is inverse positive, i.e., $T^{-1} \geq 0$. A necessary and sufficient condition for inverse positivity of an L -matrix T is the existence of a vector $v > 0$ such that $Tv > 0$. A symmetric nonsingular M -matrix is a Stieltjes matrix.

THEOREM 2.2. *If $A^{(0)}$ is an $N \times N$ Stieltjes matrix, then the block odd-even reduction algorithm generates $A^{(q)}$, $0 \leq q \leq p$, that are also $N \times N$ Stieltjes matrices.*

Proof. The proof is by induction on q . Assume $A^{(q)}$ is a Stieltjes matrix. Then, setting certain off-diagonal entries of $A^{(q)}$ to zero still results in an M -matrix (cf. Varga [19, Theorem 3.12]). In particular, $A_0^{(q)}$ is a Stieltjes matrix, $(A_0^{(q)})^{-1} \geq 0$, and

$$Q^{(q)} = I - A^{(q)}(A_0^{(q)})^{-1} \geq 0.$$

Let $x > 0$ such that $A^{(q)}x > 0$. Then

$$(2.5) \quad A^{(q+1)}x = (I + Q^{(q)})A^{(q)}x > 0.$$

By (2.2),

$$\begin{aligned} A_0^{(q+1)} &= A_0^{(q)} - A_{-2q}^{(q)}(A_0^{(q)})^{-1}A_{2q}^{(q)} - A_{2q}^{(q)}(A_0^{(q)})^{-1}A_{-2q}^{(q)}, \\ A_{2q}^{(q+1)} &= -A_{2q}^{(q)}(A_0^{(q)})^{-1}A_{2q}^{(q)}, \end{aligned}$$

and

$$A_{-2q}^{(q+1)} = -A_{-2q}^{(q)}(A_0^{(q)})^{-1}A_{-2q}^{(q)}.$$

Hence, $A^{(q+1)}$ is an L -matrix and, by (2.5), an M -matrix. Symmetry follows from the fact that $(A_{2q}^{(q)})' = A_{-2q}^{(q)}$. \square

As can be easily seen, the block odd-even reduction algorithm causes numerical fill-in of the blocks almost immediately from the onset, and any matrix sparsity patterns in the original system (1.1) are completely destroyed. However, following the example of the incomplete Cholesky factorization, sparsity patterns can still be maintained if an incomplete version of the block odd-even reduction algorithm is used.

Sparsity patterns are formalized by using the concept of a graph. If $M = \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$, then a graph G is any subset of M . A given graph G induces an operator sp_G on the set of all $n \times n$ matrices, $A(n)$, as follows:

$$\text{sp}_G: A(n) \rightarrow A(n), \quad \text{sp}_G[A] = B = (b_{ij}),$$

where

$$b_{ij} = 0 \quad \text{if } (i, j) \notin G \quad \text{and} \quad b_{ij} = a_{ij} \quad \text{if } (i, j) \in G.$$

A graph G is said to be symmetric if $(i, j) \in G$ implies $(j, i) \in G$. Hence, if G is symmetric,

$$[\text{sp}_G(A)]' = \text{sp}_G[A'].$$

For matrices with block entries, $T = (T_{ij})$ where $T_{ij} \in A(n)$, define

$$\text{sp}_G[T] = (\text{sp}_G[T_{ij}]).$$

In the same way, if G is a symmetric graph,

$$[\text{sp}_G(T)]' = \text{sp}_G(T').$$

We assume in the remainder of the paper that a symmetric graph has been given and write sp as the induced operator sp_G . An incomplete version of the block odd-even reduction algorithm now follows.

INCOMPLETE BLOCK ODD - EVEN REDUCTION

Decomposition: Let $\tilde{A}^{(0)} = A$ and p an integer such that $2^p \leq L < 2^{p+1}$. Then for $q = 0, 1, \dots, p - 1$,

Step 1. This is just the normal Cholesky $\tilde{L}^{(q)}\tilde{D}^{(q)}(\tilde{L}^{(q)})^t$ factorization of $\tilde{A}_0^{(q)}$;

Step 2. Define

$$\tilde{C}_+^{(q)} = \text{sp}\left[\tilde{A}_{2^q}^{(q)}(\tilde{L}^{(q)})^{-t}(\tilde{D}^{(q)})^{-1}\right], \quad \tilde{C}_-^{(q)} = \text{sp}\left[\tilde{A}_{-2^q}^{(q)}(\tilde{L}^{(q)})^{(-t)}(\tilde{D}^{(q)})^{-1}\right].$$

Step 3. Define

$$\tilde{A}^{(q+1)} = \tilde{A}_{-2^q}^{(q+1)} + \tilde{A}_0^{(q+1)} + \tilde{A}_{2^q}^{(q+1)},$$

where

$$\begin{aligned} \tilde{A}_0^{(q+1)} &= \tilde{A}_0^{(q)} - \text{sp}\left[\tilde{C}_+^{(q)}\tilde{D}^{(q)}(\tilde{C}_+^{(q)})^t\right] - \text{sp}\left[(\tilde{C}_-^{(q)})^t\tilde{D}^{(q)}(\tilde{C}_-^{(q)})^t\right], \\ \tilde{A}_{-2^q}^{(q+1)} &= -\text{sp}\left[(\tilde{C}_-^{(q)})\tilde{D}^{(q)}(\tilde{C}_-^{(q)})^t\right], \quad \tilde{A}_{2^q}^{(q+1)} = -\text{sp}\left[\tilde{C}_+^{(q)}\tilde{D}^{(q)}(\tilde{C}_+^{(q)})^t\right]. \end{aligned}$$

Forward Substitution: Let $y^{(0)} = y$. Then for $q = 0, 1, \dots, p - 1$,

$$y^{(q+1)} = \left[I + \tilde{C}_+^{(q)}(\tilde{L}^{(q)})^{-1} + (\tilde{C}_-^{(q)})(\tilde{L}^{(q)})^{-1} \right] y^{(q)}.$$

Backward Substitution:

$$x_I = (\tilde{A}_0^{(p)})^{-1} y^{(p)}.$$

x_I is called the *incomplete solution* of (1.1).

For the incomplete block odd-even reduction algorithm we get

LEMMA 2.1. *Suppose A is an $N \times N$ Stieltjes matrix. Then the incomplete block odd-even reduction algorithm generates $\tilde{A}^{(q)}$, $1 \leq q \leq p$, that are also $N \times N$ Stieltjes matrices.*

Proof. Using induction on q , let us assume $\tilde{A}^{(q)}$ is a Stieltjes matrix. Then $\tilde{A}_0^{(q)}$ is a Stieltjes matrix, $\tilde{A}_{\pm 2^q}^{(q)} \leq 0$, and $(\tilde{A}_{-2^q}^{(q)})^t = \tilde{A}_{2^q}^{(q)}$. Further, $(\tilde{L}^{(q)})^{-1} \geq 0$ so that $\tilde{C}_{\pm}^{(q)} \leq 0$. Thus

$$\tilde{A}_{\pm 2^{q+1}}^{(q+1)} = -\text{sp}\left[\tilde{C}_{\pm}^{(q)}\tilde{D}^{(q)}(\tilde{C}_{\mp}^{(q)})^t\right] \leq 0, \quad (\tilde{A}_{2^q}^{(q+1)})^t = \tilde{A}_{-2^q}^{(q+1)},$$

and $\tilde{A}_0^{(q+1)}$ is a symmetric L -matrix. Hence, $\tilde{A}^{(q+1)}$ is a symmetric L -matrix. It now remains to show that $\tilde{A}^{(q+1)}$ is inverse-positive. Since $\tilde{A}^{(q)}$ is an M -matrix, there exists a vector $s > 0$ such that $\tilde{A}^{(q)}s > 0$. Define

$$T = \left[2I - \tilde{A}^{(q)}(\tilde{A}_0^{(q)})^{-1} \right] \tilde{A}^{(q)}$$

so that $Ts > 0$. Further,

$$T_0 = \tilde{A}_0^{(q)} - \tilde{A}_{-2^q}^{(q)}(\tilde{A}_0^{(q)})^{-1}\tilde{A}_{2^q}^{(q)} - \tilde{A}_{2^q}^{(q)}(\tilde{A}_0^{(q)})^{-1}\tilde{A}_{-2^q}^{(q)}.$$

Since

$$\tilde{A}_{\pm 2^q}^{(q)}(\tilde{A}_0^{(q)})^{-1}\tilde{A}_{\mp 2^q}^{(q)} \leq -\text{sp}\left[\tilde{C}_{\pm}^{(q)}\tilde{D}^{(q)}(\tilde{C}_{\mp}^{(q)})^t\right] \leq 0,$$

we obtain $\tilde{A}_0^{(q+1)}s \geq T_0s \geq 0$. Similarly, $T_{\pm 2^{q+1}} \leq \tilde{A}_{\pm 2^{q+1}}^{(q+1)} \leq 0$. Thus

$$\begin{aligned} \tilde{A}^{(q+1)}s &= \tilde{A}_0^{(q+1)}s + \tilde{A}_{2^q}^{(q+1)}s + \tilde{A}_{-2^q}^{(q+1)}s \\ &\geq T_0s + T_{2^q}s + T_{-2^q}s = Ts > 0, \end{aligned}$$

and the proof follows. \square

Using the identities

$$\tilde{Q}^{(q)} = I - \tilde{A}^{(q)}(\tilde{A}_0^{(q)})^{-1} \quad \text{and} \quad \tilde{H}^{(q)} = I - (\tilde{A}_0^{(q)})^{-1}\tilde{A}^{(q)},$$

similar to the ones in (2.1) and (2.4), it is now possible to establish a theorem similar to Theorem 2.1 for the incomplete algorithm.

THEOREM 2.3. *Let A be an $N \times N$ Stieltjes matrix. Then for the incomplete block odd-even reduction algorithm,*

- (i) if $\|\tilde{H}^{(0)}\|_\infty < 1$, then $\|\tilde{H}^{(q+1)}\|_\infty \leq \|(\tilde{H}^{(q)})^2\|_\infty$, $0 \leq q \leq p - 1$,
- (ii) if $\|\tilde{Q}^{(0)}\|_1 < 1$, then $\|\tilde{Q}^{(q+1)}\|_1 \leq \|(\tilde{Q}^{(q)})^2\|_1$, $0 \leq q \leq p - 1$.

Proof. The proof is by induction on q . Assume (i) holds up to q . According to the previous lemma, $\tilde{A}^{(q)}$ is a Stieltjes matrix. Define

$$T^{(q+1)} = \tilde{A}^{(q)}(I + \tilde{H}^{(q)}) \quad \text{and} \quad H^{(q+1)} = I - (T_0^{(q+1)})^{-1}T^{(q+1)}.$$

Then, by Theorem 2.1,

$$\|H^{(q+1)}\|_\infty \leq \|(\tilde{H}^{(q)})^2\|_\infty.$$

Further,

$$\begin{aligned} T_{\pm 2^{q+1}}^{(q+1)} &= -\tilde{A}_{\pm 2^q}^{(q)}(\tilde{A}_0^{(q)})^{-1}\tilde{A}_{\pm 2^q}^{(q)} \\ &= -\left[\tilde{A}_{\pm 2^q}^{(q)}(\tilde{L}^{(q)})^{-1}(\tilde{D}^{(q)})^{-1}\right]\tilde{D}^{(q)}\left[(\tilde{D}^{(q)})^{-1}(\tilde{L}^{(q)})^{-1}\tilde{A}_{\pm 2^q}^{(q)}\right] \\ &\leq \tilde{A}_{\pm 2^{q+1}}^{(q+1)} \leq 0, \end{aligned}$$

and

$$T_0^{(q+1)} = \tilde{A}_0^{(q)} - \tilde{A}_{-2^q}^{(q)}(\tilde{A}_0^{(q)})^{-1}\tilde{A}_{2^q}^{(q)} - \tilde{A}_{2^q}^{(q)}(\tilde{A}_0^{(q)})^{-1}\tilde{A}_{-2^q}^{(q)} \leq \tilde{A}_0^{(q+1)}.$$

Since both $\tilde{A}_0^{(q+1)}$ and $T_0^{(q+1)}$ are nonsingular M -matrices, then (cf. Varga [19, p. 87])

$$(T_0^{(q+1)})^{-1} \geq (\tilde{A}_0^{(q+1)})^{-1} \geq 0$$

and

$$(T_0^{(q+1)})^{-1}(T_{\pm 2^{q+1}}^{(q+1)}) \leq (\tilde{A}_0^{(q+1)})^{-1}(\tilde{A}_{\pm 2^{q+1}}^{(q+1)}) \leq 0$$

yielding

$$\|\tilde{H}^{(q+1)}\|_\infty \leq \|H^{(q+1)}\|_\infty,$$

and (i) follows. The proof of (ii) follows in a similar manner. \square

COROLLARY. *Suppose A is an $N \times N$ M -matrix with $\|H^{(0)}\|_\infty < 1$. Let*

$$y^{(m)} = \prod_{q=0}^{m-1} \left[I + \tilde{C}_+^{(q)}(\tilde{L}^{(q)})^{-1} + (\tilde{C}_-^{(q)})(\tilde{L}^{(q)})^{-1} \right] y \quad \text{and} \quad x^{(m)} = [\tilde{A}_0^{(m)}]^{-1}y^{(m)}.$$

If x_I is the incomplete solution of (1.1), then

$$\frac{\|x_I - x^{(m)}\|_\infty}{\|x_I\|_\infty} \leq \|\tilde{H}^{(m)}\|_\infty.$$

Proof. The proof follows from the fact that

$$\tilde{A}_0^{(m)}(x_I - x^{(m)}) = -[\tilde{A}_{-2^m}^{(m)} + \tilde{A}_{2^m}^{(m)}]x_I. \quad \square$$

3. Incomplete Block Cyclic Reduction. The algorithm of cyclic reduction is a variation of the odd-even reduction algorithm where only selected block rows of the matrix are eliminated. The basic idea is as follows:

Consider an even block equation of (1.1)

$$B_{2i-1}x_{2i-1} + A_{2i}x_{2i} + B_{2i}^t x_{2i+1} = y_{2i}.$$

We eliminate the odd-indexed unknowns from the even-indexed equation by multiplying equation $(2i - 1)$ by $(-B_{2i-1}A_{2i-1}^{-1})$ and equation $(2i + 1)$ by $(-B_{2i}^t A_{2i+1}^{-1})$ and adding these to equation $2i$. The resulting equation is

$$\begin{aligned} &(-B_{2i-1}A_{2i-1}^{-1}B_{2i-2})x_{2i-2} + (A_{2i} - B_{2i-1}A_{2i-1}^{-1}B_{2i-1}^t - B_{2i}^t A_{2i+1}^{-1}B_{2i}^t)x_{2i} \\ &- (B_{2i}^t A_{2i+1}^{-1}B_{2i+1}^t)x_{2i+2} = y_{2i} - B_{2i-1}A_{2i-1}^{-1}y_{2i-1} - B_{2i}^t A_{2i+1}^{-1}y_{2i+1}. \end{aligned}$$

If we do this to all of the even equations, they make up a new block-tridiagonal system with half of the original number of blocks. Once this new system has been solved, the solution of the original system may be obtained by back-substitution into the odd equations. To solve this new system, the same approach may be applied to its even indexed equations (the indices of these were originally multiples of four). The process may be continued until just one block remains.

The complete algorithm can be expressed as follows: (Here, the matrix subscript denotes the subblock position of (1.1) rather than the diagonals as before.)

BLOCK CYCLIC REDUCTION (Version 1)

Decomposition: Let $B_i^{(0)} = B_i, A_i^{(0)} = A_i (i = 1, \dots, L)$, and p be an integer such that $2^p \leq L < 2^{p+1}$. Then for each $q = 0, 1, 2, \dots, p - 1$,

- (1) $C_{2i-1}^{(q)} = B_{2i-1}^{(q)}(A_{2i-1}^{(q)})^{-1}, 1 \leq (2i - 1) \leq (L \text{ DIV } 2^q)$,
- (2) $C_{2i}^{(q)} = (B_{2i}^{(q)})^t(A_{2i+1}^{(q)})^{-1}, 2 \leq 2i \leq (L \text{ DIV } 2^q)$,
- (3) $A_i^{(q+1)} = A_{2i}^{(q)} - C_{2i-1}^{(q)}(B_{2i-1}^{(q)})^t - C_{2i}^{(q)}B_{2i}^{(q)}, 1 \leq i \leq (L \text{ DIV } 2^{q+1})$,
- (4) $B_i^{(q+1)} = -C_{2i+1}^{(q)}B_{2i}^{(q)}, 1 \leq i \leq (L \text{ DIV } 2^{q+1} - 1)$,

where $(a \text{ DIV } b) = c, c$ the largest integer $\leq a/b$.

Forward Substitution: Let $y_i^{(0)} = y_i (i = 1, \dots, L)$. Then for each $q = 0, 1, 2, \dots, p - 1$,

$$y_i^{(q+1)} = y_{2i}^{(q)} - C_{2i-1}^{(q)}y_{2i-1}^{(q)} - C_{2i}^{(q)}y_{2i+1}^{(q)}, \quad 1 \leq i \leq (L \text{ DIV } 2^{q+1}).$$

Backward Substitution: Let $x_1^{(p)} = (A^{(p)})^{-1}y_1^{(p)}$. Then for each $q = p - 1, p - 2, \dots, 1, 0$

$$\begin{aligned} x_{2i}^{(q)} &= x_i^{(q+1)}, \quad 1 \leq i \leq (L \text{ DIV } 2^{q+1}), \\ x_{2i-1}^{(q)} &= (A_{2i-1}^{(q)})^{-1}y_{2i-1}^{(q)} - (C_{2i-2}^{(q)})^t x_{2i-2}^{(q)} \\ &\quad - (C_{2i-1}^{(q)})^t x_{2i}^{(q)}, \quad 1 \leq (2i - 1) \leq (L \text{ DIV } 2^q). \end{aligned}$$

Formally, block cyclic reduction can also be viewed with the use of odd-even permutations $P^{(q)}$. Such permutations take an ordering of blocks $1, 2, \dots$ into the ordering $1, 3, 5, \dots, 2, 4, \dots$, cf. Rodrigue et al., [17].

BLOCK CYCLIC REDUCTION (Version 2)

Decomposition: For $q = 0, 1, \dots, p - 1$, let (we now resume the diagonal notation)

$$P^{(q+1)}A^{(q)}(P^{(q+1)})^t = \begin{bmatrix} G^{(q)} & | & (E^{(q)})^t \\ \hline E^{(q)} & | & F^{(q)} \end{bmatrix},$$

$$C^{(q)} = E^{(q)}(G^{(q)})^{-1}, \quad A^{(q+1)} = F^{(q)} - C^{(q)}(E^{(q)})^t.$$

Forward Substitution: Let $y^{(0)} = y$. For $q = 0, 1, \dots, p - 1$

$$P^{(q+1)}y^{(q)} = \begin{bmatrix} y_{\text{od}}^{(q)} \\ y_{\text{ev}}^{(q)} \end{bmatrix}, \quad y^{(q+1)} = y_{\text{ev}}^{(q)} - C^{(q)}y_{\text{od}}^{(q)}.$$

Backward Substitution: Let $x^{(p)} = (A^{(p)})^{-1}y^{(p)}$. For $q = p - 1, p - 2, \dots, 0$

$$x_{\text{od}}^{(q)} = (G^{(q)})^{-1} [y_{\text{od}}^{(q)} - (E^{(q)})^t x^{(q+1)}].$$

Let $x^{(q)}$ be such that

$$P^{(q+1)}x^{(q)} = \begin{bmatrix} x_{\text{od}}^{(q)} \\ x^{(q+1)} \end{bmatrix}.$$

For $q = 0, 1, \dots, p - 1$, let

$$H^{(q)} = I - (A_0^{(q)})^{-1}A^{(q)}, \quad Q^{(q)} = I - A^{(q)}(A_0^{(q)})^{-1}.$$

Since cyclic reduction is nothing more than odd-even reduction on selected block rows of the matrix, we get immediately from Theorem 2.1,

THEOREM 3.1 *Suppose $A_0^{(0)}$ is invertible and $\|H^{(0)}\|_\infty < 1$. Then for $q = 0, 1, \dots, p - 1$, the block cyclic reduction algorithm yields*

- (i) $A_0^{(q)}$ is invertible, i.e., cyclic reduction is well-defined;
- (ii) $\|H^{(q+1)}\|_\infty \leq \|(H^{(q)})^2\|_\infty$;
- (iii) If $\|Q^{(0)}\|_1 < 1$, then $\|Q^{(q+1)}\|_1 < \|Q^{(q)^2}\|_1$.

As in incomplete odd-even reduction, it is also possible to define an incomplete block cyclic reduction algorithm.

INCOMPLETE BLOCK CYCLIC REDUCTION

Decomposition: For $q = 0, \dots, p - 1$

$$(3.9) \quad P^{(q+1)}\tilde{A}^{(q)}(P^{(q+1)})^t = \begin{bmatrix} G^{(q)} & | & (E^{(q)})^t \\ \hline E^{(q)} & | & F^{(q)} \end{bmatrix},$$

$$(3.9a) \quad \tilde{G}^{(q)} = \tilde{L}^{(q)}\tilde{D}^{(q)}(\tilde{L}^{(q)})^t \quad (\text{Cholesky decomposition}),$$

$$(3.9b) \quad C_+^{(q)} = \tilde{E}_0^{(q)}(\tilde{L}^{(q)})^{-t}(\tilde{D}^{(q)})^{-1},$$

$$(3.9c) \quad C_-^{(q)} = \tilde{E}_1^{(q)}(\tilde{L}^{(q)})^{-t}(\tilde{D}^{(q)})^{-1},$$

$$(3.9d) \quad \tilde{C}_+^{(q)} = \text{sp}[C_+^{(q)}],$$

$$(3.9e) \quad \tilde{C}_-^{(q)} = \text{sp}[C_-^{(q)}],$$

$$(3.9f) \quad \tilde{A}^{(q+1)} = \tilde{A}_-^{(q+1)} + \tilde{A}_0^{(q+1)} + \tilde{A}_+^{(q+1)},$$

$$(3.9g) \quad \tilde{A}_0^{(q+1)} = \tilde{F}^{(q)} - \text{sp}[\tilde{C}_+^{(q)}\tilde{D}^{(q)}(\tilde{C}_+^{(q)})^t] - \text{sp}[\tilde{C}_-^{(q)}\tilde{D}^{(q)}(\tilde{C}_-^{(q)})^t],$$

$$(3.9h) \quad \tilde{A}_-^{(q+1)} = -\text{sp}[\tilde{C}_+^{(q)}\tilde{D}^{(q)}(\tilde{C}_+^{(q)})^t],$$

$$(3.9i) \quad \tilde{A}_+^{(q+1)} = -\text{sp}[\tilde{C}_-^{(q)}\tilde{D}^{(q)}(\tilde{C}_-^{(q)})^t].$$

Forward Substitution: For $q = 0, 1, 2, \dots, p - 1$

$$y^{(q)} = \begin{bmatrix} y_{\text{od}}^{(q)} \\ y_{\text{ev}}^{(q)} \end{bmatrix},$$

$$(3.10) \quad y^{(q+1)} = y^{(q)} - \bar{C}_+^{(q)}(L^{(q)})^{-1}y_{\text{ev}}^{(q)} - \bar{C}_-^{(q)}(L^{(q)})^{-1}y_{\text{ev}}^{(q)}$$

where $\bar{C}_\pm = \tilde{C}_\pm$ or C_\pm as before.

Backward Substitution: Let

$$x^{(p)} = (\tilde{A}^{(p)})^{-1}y^{(p)}.$$

Then for $q = p - 1, p - 2, \dots, 1, 0$,

$$(3.11) \quad x_{\text{od}}^{(q)} = (\tilde{L}^{(q)})^{-t}[(\tilde{D}^{(q)})^{-1}(\tilde{L}^{(q)})^{-1}y_{\text{ev}}^{(q)} - (\bar{C}_+^{(q)})^t x^{(q+1)} - (\bar{C}_-^{(q)})^t x^{(q+1)}],$$

let $x^{(q)}$ be such that

$$P^{(q+1)}x^{(q)} = \begin{bmatrix} x_{\text{od}}^{(q)} \\ x^{(q+1)} \end{bmatrix}.$$

Let

$$\tilde{H}^{(q)} = I - (\tilde{A}_0^{(q)})^{-1}\tilde{A}^{(q)}, \quad \tilde{Q}^{(q)} = I - \tilde{A}^{(q)}(\tilde{A}_0^{(q)})^{-1}.$$

As stated before, incomplete block cyclic reduction is incomplete block odd-even reduction on selected block rows of the matrix. Hence we get

THEOREM 3.2. *Suppose A is a Stieltjes matrix and $\|H^{(0)}\|_\infty < 1$. Then for $q = 0, 1, \dots, p - 1$, the incomplete block cyclic reduction algorithm yields*

- (i) $\tilde{A}^{(q)}$ is a Stieltjes matrix;
- (ii) $\|\tilde{H}^{(q+1)}\|_\infty \leq \|(\tilde{H}^{(q)})^2\|_\infty$;
- (iii) If $\|\tilde{Q}^{(0)}\|_1 < 1$, then $\|\tilde{Q}^{(q+1)}\|_1 \leq \|(\tilde{Q}^{(q)})^2\|_1$.

If the decomposition and forward-substitution phases are terminated at $q = m \leq p - 1$ and the back-substitution begins with

$$x^{(m)} = (\tilde{A}_0^{(m)})^{-1}y^{(m)},$$

then we get

COROLLARY. *If A is a Stieltjes matrix and $\|H^{(0)}\|_\infty < 1$, then*

$$\frac{\|x_I - x_I^{(m)}\|_\infty}{\|x_I\|_\infty} \leq \|\tilde{H}^{(m)}\|_\infty,$$

where x_I is the incomplete solution and $x_I^{(m)}$ is the incomplete solution determined by terminating the process at $q = m \leq p - 1$.

4. Point vs. Block Incomplete Cholesky Decomposition. A few words on the difference between the usual incomplete Cholesky decomposition and that given in the previous section are worth mentioning. Cyclic reduction is equivalent to block Gaussian Elimination without (block) pivoting on a permuted system $(PAP')(Px) = Py$ where P is a permutation matrix taking the original ordering of blocks $1, 2, \dots, L$ into $1, 3, 5, \dots, 2, 6, 10, \dots, 4, 12, 20, \dots, 2^p$. To illustrate, for $L = 7$

$$(4.1) \quad PAP' = \begin{bmatrix} A_1 & & & & B'_1 & & & & \\ & A_3 & & & B_2 & & & & B'_3 \\ & & A_5 & & & B'_5 & & & B_4 \\ & & & A_7 & & B_6 & & & \\ B_1 & B'_2 & & & A_2 & & & & \\ & & B_5 & B'_6 & & A_6 & & & \\ & B_3 & B'_4 & & & & & & A_4 \end{bmatrix}.$$

Further, the block Cholesky factorization of $PAP' = LDL'$ yields a block lower triangular matrix L with the block structure

$$(4.2) \quad L = \begin{bmatrix} \times & & & & & & & & \\ & \times & & & & & & & \bigcirc \\ & & \times & & & & & & \\ & & & \times & & & & & \\ \times & \times & & & \times & & & & \\ & & \times & \times & & & \times & & \\ & \times & \times & & \times & \times & \times & & \end{bmatrix}.$$

As mentioned earlier in Section 1, the standard incomplete Cholesky decomposition commonly found in the literature generates a factorization LDL' where L retains the sparsity pattern of (4.1). However, in the block incomplete cyclic reduction algorithm, the block sparsity pattern of the block Cholesky factor in (4.2) is maintained.

5. Applications. We consider certain finite difference approximations of the second order elliptic partial differential equations in two dimensions.

$$(5.1) \quad -\nabla\Lambda(x, y)\nabla u + p(x, y)u = -\frac{\partial\Lambda\partial u}{\partial x^2} - \frac{\partial\Lambda\partial u}{\partial y^2} + p(x, y)u = f(x, y)$$

with $\Lambda > 0, p \geq 0, (x, y)$ in some two-dimensional region R . Associated boundary conditions are

$$\alpha(x, y) + \beta(x, y) \frac{\partial u}{\partial n} = \alpha(x, y),$$

where (x, y) are in Γ , the sufficiently smooth boundary of R , and $\partial u/\partial n$ refers to outward pointing normal on Γ .

To approximate the solution of (5.1), a network of straight lines with mesh spaces h_x and h_y , parallel to each of the coordinate axes, is superimposed over the region. Then at each regular mesh point (ih_x, jh_y) ,

$$\begin{aligned} (-\nabla\Lambda\nabla u)_{ij} &= \frac{h_y}{h_x} [\Lambda_{i+1/2,j}(u_{ij} - u_{i+1,j}) + \Lambda_{i-1/2,j}(u_{ij} - u_{i-1,j})] \\ &\quad + \frac{h_x}{h_y} [\Lambda_{i,j+1/2}(u_{ij} - u_{i,j+1}) + \Lambda_{i,j-1/2}(u_{ij} - u_{i,j-1})]. \end{aligned}$$

This yields a set of inhomogeneous, linear, simultaneous, symmetric equations which can be expressed in matrix notation $Ax = y$ where x and y consist of the unknown approximate solution u_{ij} at (ih_x, jh_y) and the known boundary values, respectively. If we assume a columnwise ordering of the unknowns, the coefficient matrix A is a sparse, structured, real $N \times N$ matrix (N is the number of unknown mesh points) with the following properties: (i) symmetric and (ii) positive definite. We will use the generalized conjugate gradient method of Concus et al., [2], to solve this system. A description of the algorithm is given in Appendix I. All computations are performed on a CRAY-1 vector-processor.

For our first set of computations, we compare the generalized conjugate gradient methods for preconditioners given by an incomplete Cholesky factorization and an incomplete block cyclic reduction. For the calculations performed here, the incomplete Cholesky factorization provides a preconditioner of the form $M = LL^T$ where

$$L = \begin{bmatrix} E_1 & & & & & \\ & F_2 & & & & \\ & & E_2 & & & \\ & & & \ddots & & \\ & & & & F_L & \\ & & & & & E_L \end{bmatrix},$$

where the E_i are lower bidiagonal matrices and the F_i are tridiagonal matrices. This factorization is denoted by IC(1, 1), [18].

The solution of the residual correction $Mz_i = r_i$ (cf. (3) in Appendix I) is given by

Forward Substitution:

$$s_1 = E_1^{-1}r_1.$$

For $i = 2, \dots, L$,

$$s_i = E_i^{-1}(r_i - F_i s_{i+1}).$$

Backward Substitution:

$$z_L = (E_L)^{-1}s_L.$$

For $i = L - 1, \dots, 1$,

$$z_i = (E_i^{-t})(s_i - F_i^t z_{i-1}).$$

For the computations here, the inversion of both E_i and E_i^t were performed by a scalar first order recursion. Although on vector machines other than the CRAY, algorithms such as recursive doubling, cf. Dubois and Rodrigue [3], may be more suitable.

Incomplete block cyclic reduction consists of two phases: the decomposition and the forward/backward substitution. The decomposition is performed only once and is accomplished by the algorithm given in (3.9a-i). The implementation of this algorithm for calculations done on a vector-processor is given in Appendix II where the sparsity pattern "sp" is the tridiagonal pattern. In this way, a solution of the equation $Mz_i = r_i$ by incomplete block cyclic reduction in the conjugate gradient method is provided.

obtained by stopping the incomplete block cyclic reduction process at the m th cycle, then by Theorem 3.2, the iteration count may not be severely affected. To test this, Problem 2 was rerun. As a consequence of Theorem 3.4 in Heller, [7], the matrix does in fact satisfy Theorem 3.2. Table 3 records the results. For comparison purposes, the time in parentheses is the time taken from Table 2 for the IC(1, 1) algorithm.

Comments. As one can see, only two cycles of reduction are necessary to achieve the minimum number of iterations. In doing so, the IC(1, 1) is now twice as expensive as the early termination IBCR method. If the algorithm were executed in a multi-processing environment, an even greater speed-up would be obtained since each level of cyclic reduction involves data-movement through the memory-processor interface.

7. Acknowledgements. The authors would like to acknowledge helpful discussions with David Kershaw and Alex Friedman. We also thank the U.S. Department of Energy, Office of Basic Energy Sciences, Applied Mathematical Sciences Division, for support.

TABLE 1. *5 pt. Laplace equation on $N \times N$ grid*

\sqrt{N}	Itns		Time	
	IC(1, 1)	IBCR	IC(1, 1)	IBCR
10	15	10	.008	.016
30	31	24	.15	.13
50	47	39	.67	.43
75	67	55	2.1	1.2
100	87	72	4.9	2.8

TABLE 2. *5 pt. random matrix*

\sqrt{N}	Itns		Time	
	IC(1, 1)	IBCR	IC(1, 1)	IBCR
10	22	16	.01	.02
30	48	40	.24	.2
50	75	61	1.05	.66
75	107	91	3.4	1.8
100	135	114	7.6	4.4

TABLE 3. *Random matrix with early termination of cyclic reduction*

\sqrt{N}	q	Itns.	Time	
30	1	77	.176	(1.24)
	2	42	.136	
	3	40	.168	
	4	40	.2	
50	1	128	.6	(1.05)
	2	68	.43	
	3	62	.49	
	4	61	.57	
	5	61	.66	
100	1	249	4.6	(7.6)
	2	133	3.2	
	3	114	3.2	
	4	114	3.6	
	5	114	4.0	
	6	114	4.4	

Appendix I

GENERALIZED CONJUGATE-GRADIENT ALGORITHM for solving $Ax = y$:
 Let M be a symmetric positive-definite matrix and x_0 an arbitrary initial vector.

Let $r_0 = y - Ax_0$ and $p_0 = M^{-1}r_0$. Then for $i = 0, 1, 2, \dots$

- (1) $x_{i+1} = x_i + \alpha_i p_i$, where $\alpha_i = (r_i, M^{-1}r_i) / (p_i, Ap_i)$,
- (2) $r_{i+1} = r_i - \alpha_i Ap_i$,
- (3) $p_{i+1} = M^{-1}r_{i+1} + \beta_i p_i$, where $\beta_i = (r_{i+1}, M^{-1}r_{i+1}) / (r_i, M^{-1}r_i)$.

Appendix II

INCOMPLETE BLOCK CYCLIC REDUCTION

Decomposition: Let $A^{(0)} = A$. Then for $q = 0, 1, \dots, p - 1$ let

$$\tilde{G}^{(q)} = \tilde{L}^{(q)} \tilde{D}^{(q)} (\tilde{L}^{(q)})^t,$$

where

$$\tilde{D}^{(q)} = d_0^{(q)}, \quad \tilde{L}^{(q)} = I + l_{-1}^{(q)},$$

and

$$(1) \quad (\tilde{L}^{(q)})^{-1} = \sum_{i=0}^{K-1} (-l_{-1}^{(q)})^i.$$

Since we are using the tridiagonal sparsity pattern,

$$\tilde{C}_+^{(q)} = c_{+,-1}^{(q)} + c_{+,0}^{(q)} + c_{+,1}^{(q)}.$$

Further,

$$(2) \quad \tilde{E}_0^{(q)} = e_{-1}^{(q)} + e_0^{(q)} + e_{+1}^{(q)},$$

so that the scalar diagonals of $\tilde{C}_+^{(q)}$ are constructed by performing (3.9b) with (2) and a truncated series in (1), i.e.,

$$\begin{aligned} c_{+,-1}^{(q)} &= e_{-1}^{(q)}(d_0^{(q)})^{-1}, \\ c_{+,0} &= e_0^{(q)}(d_0^{(q)})^{-1} - e_{-1}(l_{-1}^{(q)})^t(d_0^{(q)})^{-1} = e_0^{(q)}(d_0^{(q)})^{-1} - c_{+,-1}(l_{-1}^{(q)})^t, \\ c_{+,+1} &= e_1^{(q)}(d_0^{(q)})^{-1} - e_0^{(q)}(l_{-1}^{(q)})^t(d_0^{(q)})^{-1} + e_1^{(q)}((l_{-1}^{(q)})^t)^2(d_0^{(q)})^{-1} \\ &= e_1^{(q)}(d_0^{(q)})^{-1} + c_{+,0}^{(q)}(l_{-1}^{(q)})^t. \end{aligned}$$

A similar algorithm can be used for $\tilde{C}_-^{(q)}$.

Again, since we are using the tridiagonal sparsity pattern,

$$\tilde{A}_0^{(q+1)} = a_{0,-1}^{(q+1)} + a_{0,0}^{(q+1)} + a_{0,1}^{(q+1)}.$$

Now,

$$\tilde{F}^{(q)} = f_{-1}^{(q)} + f_0^{(q)} - f_1^{(q)},$$

so that an algorithm for (3.9g) is

$$\begin{aligned} a_{0,0}^{(q+1)} &= f_0^{(q)} - c_{+,-1}^{(q)}d_0^{(q)}(c_{+,-1}^{(q)})^t - c_{+1}^{(q)}d_0^{(q)}(c_{+1}^{(q)})^t - c_{+0}^{(q)}d_0^{(q)}c_{+0}, \\ &\quad - c_{-,-1}^{(q)}d_0^{(q)}(c_{-,-1}^{(q)})^t - c_{-1}^{(q)}d_0^{(q)}(c_{-1}^{(q)})^t - c_{-0}^{(q)}d_0^{(q)}c_{-0}^{(q)}, \\ a_{0,-1}^{(q+1)} &= f_{-1}^{(q)} - c_{+,-1}^{(q)}d_0^{(q)}c_{+0}^{(q)} - c_{+0}^{(q)}d_0^{(q)}(c_{+1}^{(q)})^t \\ &\quad - c_{-,-1}^{(q)}d_0^{(q)}c_{-0}^{(q)} - c_{-0}^{(q)}d_0^{(q)}(c_{-1}^{(q)})^t, \\ a_{0,1}^{(q+1)} &= f_1^{(q)} - c_{+1}^{(q)}d_0^{(q)}c_{+0}^{(q)} - c_{+0}^{(q)}d_0^{(q)}(c_{+,-1}^{(q)})^t \\ &\quad - c_{-1}^{(q)}d_0^{(q)}c_{-0}^{(q)} - c_{-0}^{(q)}d_0^{(q)}(c_{-,-1}^{(q)})^t. \end{aligned}$$

Similarly, an algorithm for (3.9h)

$$\begin{aligned} a_{-1,0}^{(q+1)} &= -c_{+0}^{(q)}d_0^{(q)}(c_{-0}^{(q)}) - c_{+,-1}^{(q)}d_0^{(q)}(c_{-,-1}^{(q)})^t - c_{+1}^{(q)}d_0^{(q)}(c_{-1}^{(q)})^t, \\ a_{-1,-1}^{(q+1)} &= -c_{+,-1}^{(q)}d_0^{(q)}c_{-0}^{(q)} - c_{+0}^{(q)}d_0^{(q)}(c_{-1}^{(q)})^t, \\ a_{-1,1}^{(q+1)} &= -c_{+1}^{(q)}d_0^{(q)}c_{-0}^{(q)} - c_{+0}^{(q)}d_0^{(q)}(c_{-,-1}^{(q)})^t, \end{aligned}$$

and an algorithm for (3.9i)

$$\begin{aligned} a_{+1,0}^{(q+1)} &= -c_{-1}^{(q)}d_0^{(q)}(c_{+1}^{(q)})^t - c_{-,-1}^{(q)}d_0^{(q)}(c_{+,-1}^{(q)})^t - c_{-0}^{(q)}d_0^{(q)}c_{+0}^{(q)}, \\ a_{+1,-1}^{(q+1)} &= -c_{-,-1}^{(q)}d_0^{(q)}c_{+0}^{(q)} - c_{-0}^{(q)}d_0^{(q)}(c_{+1}^{(q)})^t, \\ a_{+1,+1}^{(q+1)} &= -c_{-1}^{(q)}d_0^{(q)}c_{+0}^{(q)} - c_{-0}^{(q)}d_0^{(q)}(c_{+,-1}^{(q)})^t. \end{aligned}$$

The forward and backward substitutions are given by (3.10) and (3.11) where the inversion of the $\tilde{L}^{(q)}$'s is carried out by a first order scalar recursion.

Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, California 94550

Department of Mathematics
Wellesley College
Wellesley, Massachusetts 02181

1. O. AXELSSON, "A generalized SSOR method," *BIT*, v. 13, 1972, pp. 443–467.
2. P. CONCUS, G. GOLUB & 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.
3. P. DUBOIS & G. RODRIGUE, *An Analysis of the Recursive Doubling Algorithm*, Proceedings of High Speed Computer and Algorithm Organization, Academic Press, New York, 1977.
4. J. EHREL, A. LICHNEWSKY & F. THOMASSET, *Parallelism in Finite Element Computation*, Proceedings of I.B.M. Symposium on Vector Computers and Scientific Computation, Rome, 1982.
5. A. GREENBAUM & G. RODRIGUE, *The Incomplete Cholesky Conjugate Gradient Method for the STAR (5-Point Operator)*, LLNL Report UCID-17574, 1977.
6. I. GUSTAFSSON, "A class of first order factorization methods," *BIT*, v. 18, 1978, pp. 142–156.
7. D. HELLER, "Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems," *SIAM J. Numer. Anal.*, v. 13, no. 4, Sept. 1976, pp. 484–496.
8. T. JORDAN, "A guide to parallel computation and some CRAY-1 experiences," *Parallel Computations* (G. Rodrigue, ed.), Academic Press, New York, 1982.
9. D. KERSHAW, "Solution of single tridiagonal linear systems and vectorization of the ICCG algorithm on the CRAY-1," *Parallel Computations* (G. Rodrigue, ed.), Academic Press, New York, 1982.
10. D. KERSHAW, "The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations," *J. Comput. Phys.*, v. 26, 1978, pp. 43–65.
11. J. J. LAMBIOTTE & R. G. VOIGT, *The Solution of Tridiagonal Linear Systems on the CDC-STAR-100 Computer*, ICASE Report, NASA Langley Research Center, Hampton, Virginia, 1974.
12. N. K. MADSEN & G. H. RODRIGUE, *A Comparison of Direct Methods for Tridiagonal Systems on the CDC-STAR-100*, Report UCRL-76993, Lawrence Livermore Laboratory, July 1975.
13. N. MADSEN, G. RODRIGUE & J. KARUSH, "Matrix multiplication by diagonals on a vector parallel processor," *J. Inform. Process.*, v. 5, No. 2, June 1976.
14. T. A. MANTEUFFEL, "The shifted incomplete-Cholesky factorization," *Sparse Matrix Proceedings* (Iain S. Duff and G. W. Stewart, eds.), SIAM, Philadelphia, Pa., 1978.
15. J. A. MEIJERINCK & H. A. VAN DER VORST, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix," *Math. Comp.*, v. 31, 1977, pp. 148–162.
16. G. RODRIGUE, C. HENDRICKSON & M. PRATT, "The numerical solution of the Lagrangian diffusion equation on a vector processor," *Parallel Computation* (G. Rodrigue, ed.), Academic Press, New York, 1982.
17. G. RODRIGUE, N. K. MADSEN & J. T. KARUSH, "Odd-even reduction for banded linear systems," *J. Assoc. Comput. Mach.*, v. 26, no. 1, January 1979, pp. 72–81.
18. H. A. VAN DER VORST & J. M. VAN KATS, *Manteuffel's Algorithm with Preconditioning for the Iterative Solution of Certain Sparse Linear Systems with a Non-Symmetric Matrix*, Technical Report TR-11, ACCU-Reeks No. 29, Academic Computer Centre, University of Utrecht, The Netherlands, Aug. 1979.
19. R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1962.
20. R. S. VARGA, "Factorization and normalized iterative methods," *Boundary Problems in Differential Equations* (R. E. Langer, ed.), Univ. of Wisconsin Press, Madison, 1960, pp. 121–142.
21. D. YOUNG, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1972.