ACCELERATED ITERATIVE METHODS FOR THE SOLUTION OF TRIDIAGONAL SYSTEMS ON PARALLEL COMPUTERS

D. E. Heller

D. K. Stevenson

J. F. Traub

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania

December, 1974

Part of this work was performed under the auspices of the U. S. Atomic Energy Commission while J. F. Traub was a consultant at Lawrence Livermore Laboratory. The work was also supported by the National Science Foundation under Grant GJ-32111 and the Office of Naval Research under Contract N00014-67-A-0010, NR 044-422.

ABSTRACT

We consider iterative methods for the solution of tridiagonal systems and present a new iteration whose rate of convergence is comparable to that of the optimal two-cyclic Chebyshev iteration but which does not require the calculation of optimal parameters. The theory has a natural extension to block tridiagonal systems. Numerical experiments suggest that on a parallel computer this new algorithm is the best of the iterative algorithms we consider.

TABLE OF CONTENTS

1. Introduction.

In this paper we consider iterative methods for the solution of the tridiagonal system

$$A\underline{x} = \underline{c} \tag{1.1}$$

where for convenience A is taken to have unit diagonal:

$$A = \begin{pmatrix} 1 & b_1 & & & \\ a_2 & 1 & b_2 & & \\ & a_3 & 1 & b_3 & \\ & & \ddots & \ddots & \ddots \\ & & & a_{n-1} & 1 & b_{n-1} \\ & & & & a_n & 1 \end{pmatrix} . \tag{1.2}$$

We present a new iteration whose rate of convergence is comparable to that of the optimal two-cyclic Chebyshev iteration but which does not require the calculation of optimal parameters. We give a sufficient condition for this iteration to converge which is also a non-standard sufficient condition for the system (1.1) to have a solution. The theory has a natural extension to block tridiagonal systems. Numerical experiments suggest that on a vector computer (i.e., a parallel computer which processes vectors efficiently) this new algorithm is the best of the iterative algorithms we consider.

The new algorithm illustrates two general techniques of parallel algorithms: first, altering a sequential scalar algorithm to obtain a vector iteration which exploits the parallel capabilities of vector computers; and second, arranging the vector computations to accelerate the rate of convergence of an iterative method. We also discuss a model of vector computation which we use to estimate the performance of parallel algorithms.

The solution of tridiagonal systems is a ubiquitous problem in scientific computing and has received extensive treatment in the context of sequential computers. The advent of parallel computers such as Illiac IV, Control Data's STAR-100, and Texas Instruments' ASC has led to the development of new algorithms composed largely of efficient vector operations. The work of Stone [1973,1974] dealt primarily with direct methods while Traub [1973] considered iterative methods on parallel computers. Lambiotte and Voigt [1974] reported on the implementation of the methods presented in

these earlier papers with respect to a particular vector computer, the STAR-100. The work presented here is an extension and improvement of Traub's earlier work.

The next section contains an informal derivation of the new algorithm, which we call Accelerated Parallel Gauss (APG), together with some techniques useful in parallel algorithms. Section Three is devoted to an analysis of APG based on a matrix formulation. Several variations of APG are discussed in Section Four. Three of these variations are of interest if division is expensive compared to other arithmetic operations. Some empirical observations on the numerical behavior of APG are presented in Section Five, and Section Six presents an extension of the theory to block tridiagonal systems.

In Section Seven we review briefly the classical iterative methods which lend themselves naturally to a vector formulation. Section Eight contains a detailed comparison of the various iterative methods, both theoretically for a model problem and computationally for some numerical experiments.

2. A New Algorithm.

In this section we give a precise statement of the Accelerated Parallel Gauss algorithm, together with an informal development. This is intended to be more than an expository device. We feel that the two major techniques illustrated in this section -- vectorizing an "inherently" sequential algorithm and speeding up certain types of vector iterations -- may have wide applicability for parallel algorithms in general, and this is the perspective we take in the following discussion.

The classical Gaussian elimination algorithm for tridiagonal systems may be stated as follows.

1. (Factor $A = (I + L)D(I + R)$.)
   Let $d_1 = 1$.
   For $j = 2,3,\ldots,n$ let $d_j = 1 - \dfrac{a_j b_{j-1}}{d_{j-1}}$.

2. (Solve $(I + L)\underline{f} = \underline{c}$.)
   Set $\ell_j = \dfrac{a_j}{d_{j-1}}$ for all $j > 1$.
   Let $f_1 = c_1$.
   For $j = 2,3,\ldots,n$ let $f_j = c_j - \ell_j f_{j-1}$.

3. (Solve $(I + R)\underline{x} = D^{-1}\underline{f}$.)
   Set $r_j = \dfrac{b_j}{d_j}$ and $g_j = \dfrac{f_j}{d_j}$ for all j.
   Let $x_n = g_n$.
   For $j = n-1,n-2,\ldots,1$ let $x_j = g_j - r_j x_{j+1}$.

Since any calculation by this algorithm depends upon all previously calculated results, the classical algorithm is "inherently" sequential. Traub [1973] observed that this algorithm could be converted into an iterative algorithm by successively approximating the d's, then the f's, and finally the x's. For example

$$d_j^{(i)} = 1 - \frac{a_j b_{j-1}}{d_{j-1}^{(i-1)}}, \quad \text{for all } j > 1. \tag{2.1}$$

where the superscripts denote the iteration step. Since all d's are updated simultaneously, this iteration can be viewed as a vector iteration. This iterative algorithm requires many more operations than the classical Gaussian elimination, but it becomes interesting in the context of a computer which can perform operations on vectors of length n much faster than it can perform n scalar operations.

The next observation deals with a method of accelerating certain types of vector iterations, namely those for which the updated value of a component depends only upon the value of the component with next smaller index. From (2.1) it is obvious that $d_j^{(i)}$ depends only on the value of $d_{j-1}^{(i-1)}$ and on no other component of the vector $\underline{d}^{(i-1)}$. If we update all the even-subscripted d's and then update the odd-subscripted ones, always using the most recent values available, the result for the odd-subscripted variables will be the same as if all d's were updated twice, but of course with less work. By repeating this procedure, each time a component is updated it will seem as if two complete updatings of the entire vector have been made.

In general, for any k we can restructure the vector into k smaller vectors; each new vector consists of all components of the original vector whose indices are equal modulo k. The preceding discussion was for the case k = 2. Now instead of updating elements of the original vector, we cycle through the k smaller vectors. No more work (i.e., arithmetic operations) is done in each complete cycle than in one iteration on the original vector, but the effect for each updated component is the same as if k iterations on the entire vector had occurred since its last change. Thus the iteration has been accelerated by a factor of k. This technique is similar to the use of p-cyclic matrices to accelerate linear iterations (Varga [1962]).

Of course, one should not expect that this acceleration is free, although the extra cost is nominal. A more detailed discussion of this can be found in Section 8.1. In this paper we use $k = 2$ for convenience. Although this choice may be sub-optimal, it is sufficient to give striking results. The optimal choice of k will depend on the particular machine and the size of the system.

Combining these two observations we state the new algorithm studied in this paper, the Accelerated Parallel Gauss algorithm (APG). For simplicity we assume n is even.

1. Let $\underline{d}^{(0)}$ be given. Set $d_1^{(i)} = 1$ for all i.

   For $i = 1,2,\ldots,ID$, let

$$d_j^{(i)} = 1 - \frac{a_j b_{j-1}}{d_{j-1}^{(i-1)}}, \quad j \text{ even},$$

$$d_j^{(i)} = 1 - \frac{a_j b_{j-1}}{d_{j-1}^{(i)}}, \quad j \text{ odd and } > 1.$$

2. Define $\hat{\ell}_j = \frac{a_j}{d_{j-1}^{(ID)}}$ for all $j > 1$.

   Let $\underline{f}^{(0)}$ be given. Set $f_1^{(i)} = c_1$ for all i.

For $i = 1,2,\ldots,IF$, let

$$f_j^{(i)} = c_j - \hat{\ell}_j f_{j-1}^{(i-1)}, \quad j \text{ even},$$

$$f_j^{(i)} = c_j - \hat{\ell}_j f_{j-1}^{(i)}, \quad j \text{ odd and } > 1.$$

3. Define $\hat{g}_j = \dfrac{f_j^{(IF)}}{d_j^{(ID)}}$ for all j; $\hat{r}_j = \dfrac{b_j}{d_j^{(ID)}}$ for all $j < n$.

Let $\underline{x}^{(0)}$ be given. Set $x_n^{(i)} = \hat{g}_n$ for all i.

For $i = 1,2,\ldots,IX$, let

$$x_j^{(i)} = \hat{g}_j - \hat{r}_j x_{j+1}^{(i-1)}, \quad j \text{ odd},$$

$$x_j^{(i)} = \hat{g}_j - \hat{r}_j x_{j+1}^{(i)}, \quad j \text{ even and } < n.$$

Accept $\underline{x}^{(IX)}$ as an approximation to the true solution $\underline{x}$.

3. Analysis of Accelerated Parallel Gauss.

In this section we analyze the convergence properties of APG based on a matrix formulation of the algorithm. We begin by examining a sufficient condition for non-singularity of the coefficient matrix A. This condition is also sufficient for an LDU factorization of A by an iterative method. Next we discuss iterative solutions of the bidiagonal systems which arise from this multiplicative splitting of A. Finally we establish a priori bounds on the error reduction rates of the various iterations. Throughout our analysis we will use the infinity norm for both vectors and matrices.

3.1 Sufficient Conditions for Non-Singularity.

We derive a sufficient condition for the coefficient matrix (1.2) to be non-singular; this will be a natural condition for our analysis. Let

$$A = A_L + I + A_R \tag{3.1}$$

where A_L has non-zero elements only on the first sub-diagonal and A_R has non-zero elements only on the first super-diagonal. Factor A as

$$A = (I + L)D(I + R) \tag{3.2}$$

where L and R have the same non-zero structure as A_L and A_R respectively. Thus (3.1) is an additive splitting of A while (3.2) is a multiplicative decomposition. The diagonal matrix D satisfies

$$D = I - A_L D^{-1} A_R \tag{3.3}$$

and

$$L = A_L D^{-1}, \quad R = D^{-1} A_R. \tag{3.4}$$

Let $D = \mathrm{Diag}(d_1, d_2, \ldots, d_n)$, $L = \mathrm{Subdiag}(\ell_2, \ell_3, \ldots, \ell_n)$, and $R = \mathrm{Superdiag}(r_1, r_2, \ldots, r_{n-1})$. Then $d_1 = 1$, and for $j = 2,3,\ldots,n$, $d_j = 1 - \frac{a_j b_{j-1}}{d_{j-1}}$. Furthermore, $\ell_j = \frac{a_j}{d_{j-1}}$ for $j = 2,3,\ldots,n$, and $r_j = \frac{b_j}{d_j}$ for $j = 1,2,\ldots,n-1$.

We give sufficient conditions for d_j to be bounded away from zero; in this case A will be non-singular since $\det(A) = \det(D)$ which is the product of the d's. The following lemma also yields some useful bounds. Let

$$\lambda = \max_j |4a_j b_{j-1}| = 4\|A_L A_R\|. \tag{3.5}$$

We assume throughout that $\lambda > 0$.

Lemma 3.1. Suppose $\lambda \leq 1$. Then A^{-1} exists, and

$$\frac{1 + \sqrt{1-\lambda}}{2} < d_j < \frac{3 - \sqrt{1-\lambda}}{2} \quad \text{for all } j. \tag{3.6}$$

Proof. The proof is by induction. Clearly $d_1 = 1$ satisfies the bounds. Assume d_k satisfies (3.6). Then

$$d_{k+1} = 1 - \frac{a_{k+1}b_k}{d_k} > 1 - \frac{\lambda}{2(1 + \sqrt{1-\lambda})} = \frac{1 + \sqrt{1-\lambda}}{2};$$

$$d_{k+1} = 1 - \frac{a_{k+1}b_k}{d_k} < 1 + \frac{\lambda}{2(1 + \sqrt{1-\lambda})} = \frac{3 - \sqrt{1-\lambda}}{2}. \quad \blacksquare$$

It follows that if $\lambda \leq 1$ then all principal minors of A are positive and hence A is a P-matrix (Fiedler-Ptak [1966]); A is also an H-matrix, so there exists a diagonal matrix E such that AE is strictly diagonally dominant (Ostrowski [1937], Lynn [1963]. A condition equivalent to $\lambda \leq 1$ appears in the analysis of SSOR-Semi-iterative method (Young [1972]).

If $\lambda > 1$, A can be either singular or non-singular.

Proposition. For any $\mu > 1$ there exists a singular matrix of the form (1.2) for which $\max|4a_jb_{j-1}| = \mu$.

Proof. For $\mu \geq 4$, the 4×4 matrix with $b_1 = \mu/4$, $b_2 = a_3 = 0$, $b_3 = a_2 = a_4 = 1$ satisfies the proposition. Now assume $1 < \mu < 4$. We construct a matrix of order $k + 1$ for which $d_{k+1} = 0$, where k depends on μ. Define $\delta_1 = 1$ and for $j = 2,3,\ldots$ let $\delta_j = 1 - \frac{\mu}{(4\delta_{j-1})}$. Now δ_j positive implies $\delta_{j+1} < \frac{2\delta_j}{(\mu + 1)}$, so the δ's decrease and hence there is a least k such that $0 < \delta_k \leq \frac{\mu}{4}$. Let $a_j = \frac{\mu}{4}$ for $j = 2,3,\ldots,k$; $a_{k+1} = \delta_k$ and $b_j = 1$ for $j = 1,2,\ldots,k$. Then $d_j = \delta_j$ for $j = 1,2,\ldots,k$; $d_{k+1} = 0$, and the maximum value of $|4a_jb_{j-1}|$ is μ. $\blacksquare$

The usual sufficient condition for the existence of a solution of a tridiagonal system is diagonal dominance. It is easy to show that the hypothesis $\lambda \leq 1$ can be stronger or weaker than diagonal dominance. If A is in the form (1.2) and $A' = EAE^{-1}$, with E diagonal, then the λ associated with A is the same as that associated with A'. That is, λ is invariant under diagonal similarity transformations. However, diagonal dominance is not invariant under such diagonal similarity transformations. If A is symmetric and $\lambda \leq 1$ then A is diagonally dominant and positive definite. Another sufficient condition is the following.

Proposition. Suppose $a_jb_{j-1} \leq 0$ for $j = 2,3,\ldots,n$. Then A^{-1} exists and $d_j \geq 1$.

Proof. The proof is by induction on j : $d_1 = 1$ and if $d_{j-1} \geq 1$ then,

$$d_j = 1 - \frac{a_j b_{j-1}}{d_{j-1}} \geq 1. \quad \blacksquare$$

3.2 An Iterative Multiplicative Decomposition.

Equation (3.3) suggests the following iteration. Let $D^{(0)}$ be an invertible diagonal matrix and compute

$$D^{(i)} = I - A_L (D^{(i-1)})^{-1} A_R, \text{ for all } i \geq 1. \tag{3.7}$$

From $D^{(i)}$ we can approximate L and R by

$$\hat{L} = A_L (D^{(i)})^{-1},$$

and

$$\hat{R} = (D^{(i)})^{-1} A_R .$$

If the diagonal elements of $D^{(i)}$ are labeled $d_1^{(i)}, \ldots, d_n^{(i)}$, then the iteration (3.7) is seen to be identical to (2.1).

Using the same technique as in the proof of Lemma 3.1 we can show

Lemma 3.2. Let $\lambda \leq 1$. If

$$\frac{1 + \sqrt{1-\lambda}}{2} < d_j^{(0)} < \frac{3 - \sqrt{1-\lambda}}{2}, \quad \text{for } j > 2$$

and $d_1^{(0)} = 1$, then

$$\frac{1 + \sqrt{1-\lambda}}{2} < d_j^{(i)} < \frac{3 - \sqrt{1-\lambda}}{2}, \text{ for all } i \text{ and } j. \quad \blacksquare$$

Observe that if $D^{(0)}$ is taken as the identity, this satisfies the condition of the lemma and $\|D^{(0)} - D\| < \frac{1}{2}$. The convergence of this iterative multiplicative decomposition is established by

Theorem 3.1. Let $\lambda \leq 1$ and let $D^{(0)}$ be a diagonal matrix satisfying $d_1^{(0)} = 1$ and

$$\frac{1 + \sqrt{1-\lambda}}{2} < d_j^{(0)} < \frac{3 - \sqrt{1-\lambda}}{2}, \quad \text{for } j > 2.$$

Then

$$\|D^{(i)} - D\| < \frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}} \|D^{(i-1)} - D\|, \quad \text{for all } i. \tag{3.8}$$

Proof. From the lemma

$$\frac{1}{d_j} < \frac{2}{1 + \sqrt{1-\lambda}} \quad \text{and} \quad \frac{1}{d_j^{(i)}} < \frac{2}{1 + \sqrt{1-\lambda}} \quad .$$

Since

$$d_j^{(i)} - d_j = \frac{a_j b_{j-1}}{d_{j-1} d_{j-1}^{(i-1)}} \left(d_{j-1}^{(i-1)} - d_{j-1}\right),$$

we have

$$\|D^{(i)} - D\| < \frac{\lambda}{(1 + \sqrt{1-\lambda})^2} \|D^{(i-1)} - D\|$$

and the result follows. ∎

In the event that $D^{(i)} = D$, it is easy to show that for all larger values of i equality will also hold. It is with this understanding that inequalities such as (3.8) should be interpreted in degenerate cases.

We turn to the useful technique discussed in Section Two to square the error reduction factor in the iteration for D. It will be this variation that we will use in solving tridiagonal systems in Section 3.3.

Suppose $\tilde{d}_j^{(0)} = d_j^{(0)}$ for all j, and for all $i \geq 1$ let

$$\tilde{d}_j^{(i)} = 1 - \frac{a_j b_{j-1}}{\tilde{d}_{j-1}^{(i-1)}}, \quad j \text{ even},$$

$$\tilde{d}_j^{(i)} = 1 - \frac{a_j b_{j-1}}{\tilde{d}_{j-1}^{(i)}}, \quad j \text{ odd and } > 1.$$

Then for each $i \geq 1$,

$$\tilde{d}_j^{(i)} = d_j^{(2i-1)}, \quad j \text{ even},$$

(3.9)

$$\tilde{d}_j^{(i)} = d_j^{(2i)}, \quad j \text{ odd}.$$

From this we conclude that for even j and for $i \geq 2$.

$$|\tilde{d}_j^{(i)} - d_j| = |d_j^{(2i-1)} - d_j| < \left(\frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}}\right)^2 |d_{j-2}^{(2i-3)} - d_{j-2}|$$

$$= \left(\frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}}\right)^2 |\tilde{d}_{j-2}^{(i-1)} - d_{j-2}|$$

and similarly for the odd subscripts. Thus

$$\|\tilde{D}^{(i)} - D\| < \left(\frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}}\right)^2 \|\tilde{D}^{(i-1)} - D\|, \text{ for } i \geq 2. \tag{3.10}$$

For $i = 1$, the error reduction factor is not squared.

Henceforth we will drop the tilda over d and D and refer to this accelerated iteration as the D-iteration.

3.3 The Accelerated Parallel Gauss Algorithm.

The multiplicative splitting of the matrix A, together with the solution of the two bidiagonal systems, defines an algorithm for the solution of the system $A\underline{x} = \underline{c}$. Let $A = (I + L)D(I + R)$ as before and define $\underline{f}$ by

$$(I + L)\underline{f} = \underline{c}. \tag{3.11}$$

The vector $\underline{x}$ which satisfies

$$(I + R)\underline{x} = D^{-1}\underline{f} \tag{3.12}$$

is the solution of $A\underline{x} = \underline{c}$.

Using the classical Jacobi iteration on system (3.11) produces the iteration $\underline{f}^{(i)} = \underline{c} - L\underline{f}^{(i-1)}$ or, in components, $f_j^{(i)} = c_j - \ell_j f_{j-1}^{(i-1)}$. This is a vector iteration; all components of the vector $\underline{f}$ can be updated in parallel. We note that this iteration, like the iteration for D, can be sped up by alternately updating the even-indexed components and the odd-indexed ones. A similar argument applies to the system (3.12).

This accelerated Jacobi iteration for bidiagonal systems is also a Gauss-Seidel iteration where the equations and unknowns have been re-ordered according to a red-black (or odd-even) scheme. Since for bidiagonal systems with unit diagonals the optimal SOR parameter is unity, the accelerated Jacobi iteration is an optimal SOR iteration as well.

The Accelerated Parallel Gauss algorithm consists of the following steps:

1. Compute the LDU factorization of A by the accelerated D-iteration (3.9). Compute L and R by (3.4).
2. Solve the system (3.11) iteratively using an accelerated Jacobi method.
3. Solve the system (3.12) iteratively using an accelerated Jacobi method.

The component form of the algorithm was given at the end of Section Two. For the purpose of analysis it is more convenient to use the following equivalent matrix formulation.

Let $D^{(0)}$, $\underline{f}^{(0)}$, and $\underline{x}^{(0)}$ be given. Then

$$D^{(i)} = I - A_L(I - A_L(D^{(i-1)})^{-1}A_R)^{-1}A_R, \quad i = 2,3,\ldots,ID.$$

Let $\hat{D}$ denote $D^{(ID)}$ and define $\hat{L} = A_L\hat{D}^{-1}$ and $\hat{R} = \hat{D}^{-1}A_R$. Then

$$\underline{f}^{(i)} = \underline{c} - \hat{L}\underline{c} + \hat{L}^2\underline{f}^{(i-1)}, \quad i = 2,3,\ldots,IF.$$

Let $\hat{\underline{g}} = \hat{D}^{-1}\underline{f}^{(IF)}$. Then

$$\underline{x}^{(i)} = \hat{\underline{g}} - \hat{R}\hat{\underline{g}} + \hat{R}^2\underline{x}^{(i-1)}, \quad i = 2,3,\ldots,IX.$$

Since the bidiagonal systems we solve are only approximations, we next consider the effects of using these approximations on the final accuracy obtainable by the algorithm. The following discussion leads up to the main results of this paper, Theorems 3.2 and 3.3, and reduces their proofs to simple calculations.

Suppose that instead of solving $(I + T)\underline{y} = \underline{h}$, we use an accelerated Jacobi method to solve a slightly altered system $(I + \hat{T})\hat{\underline{y}} = \hat{\underline{h}}$, where both T and $\hat{T}$ have only one non-zero diagonal. Then

$$\hat{\underline{y}}^{(i)} = \hat{\underline{h}} - \hat{T}\hat{\underline{h}} + \hat{T}^2\hat{\underline{y}}^{(i-1)}$$

and

$$\underline{y} = \underline{h} - T\underline{h} + T^2\underline{y}$$

and hence

$$\hat{\underline{y}}^{(i)} - \underline{y} = \hat{T}^2(\hat{\underline{y}}^{(i-1)} - \underline{y}) + (I-T)(\hat{\underline{h}}-\underline{h}) + (\hat{T}^2-T^2)\underline{y} + (T-\hat{T})\hat{\underline{h}}. \tag{3.13}$$

Taking norms we have

$$\|\hat{\underline{y}}^{(i)} - \underline{y}\| \leq \eta\|\hat{\underline{y}}^{(i-1)} - \underline{y}\| + \delta$$

where

$$\eta = \|\hat{T}^2\|$$

and

$$\delta = \|I - T\|\|\hat{\underline{h}} - \underline{h}\| + \|\hat{T}^2 - T^2\|\|\underline{y}\| + \|\hat{T} - T\|\|\hat{\underline{h}}\|.$$

For $\eta < 1$, this iteration will be norm reducing as long as

$$\delta < (1 - \eta)\|\hat{\underline{y}}^{(i-1)} - \underline{y}\| \tag{3.14}$$

and this condition may be insured by requiring $\|\hat{T} - T\|$ and $\|\hat{\underline{h}} - \underline{h}\|$ to be sufficiently

small. We can rewrite (3.14) as

$$\frac{\delta}{1-\eta} < \|\hat{\underline{y}}^{(i-1)} - \underline{y}\| .$$

This means that as η approaches unity, greater accuracy is required in $\hat{T}$ and $\hat{h}$ to achieve a given accuracy in $\hat{\underline{y}}$. We give a numerical example of this phenomenon in Section Five.

An application of this discussion to the bidiagonal systems in Steps 2 and 3 of APG yields bounds on the rates of convergence of these iterations.

Theorem 3.2. Let $\|\hat{L}^2\| < 1$ and $\|\hat{R}^2\| < 1$. Then

(i) $\|\underline{f}^{(i)} - \underline{f}\| \leq \|\hat{L}^2\|\|\underline{f}^{(i-1)} - \underline{f}\| + \delta_f$, $i = 2,3,\ldots,IF$;

(ii) $\|\underline{x}^{(i)} - \underline{x}\| \leq \|\hat{R}^2\|\|\underline{x}^{(i-1)} - \underline{x}\| + \delta_x$, $i = 2,3,\ldots,IX$

where δ_f and δ_x depend upon the previous steps and can be made arbitrarily small. For $i = 1$ the error reduction factors involve $\hat{L}$ and $\hat{R}$ instead of $\hat{L}^2$ and $\hat{R}^2$.

Proof. It follows from (3.13) that

$$\underline{f}^{(i)} - \underline{f} = \hat{L}^2(\underline{f}^{(i-1)} - \underline{f}) + (\hat{L}^2 - L^2)\underline{f} + (L - \hat{L})\underline{c}$$

from which (i) holds with

$$\delta_f \leq \|\hat{L}^2 - L^2\|\|\underline{f}\| + \|\hat{L} - L\|\|\underline{c}\|.$$

Also from (3.13),

$$\underline{x}^{(i)} - \underline{x} = \hat{R}^2(\underline{x}^{(i-1)} - \underline{x}) + (\hat{R}^2 - R^2)\underline{x} + (R-\hat{R})\hat{D}^{-1}\underline{f}^{(IF)} + (I-R)(\hat{D}^{-1}\underline{f}^{(IF)} - D^{-1}\underline{f})$$

and (ii) holds with the obvious bound on δ_x. For $i = 1$ the proof is straightforward. ∎

3.4 A Priori Bounds.

Since $\hat{L}$ and $\hat{R}$ are calculated in the LDU decomposition and are available during APG, the hypotheses of Theorem 3.2 are easily verified before the solution of the bidiagonal systems is attempted. One can, however, determine conditions based only on the matrix A in order to bound the rate of convergence of APG before the LDU decomposition is calculated. This is the content of Theorem 3.3.

For this purpose we need the following definitions. Let

$$\alpha = \max_j |a_j a_{j-1}|^{\frac{1}{2}} = \|A_L^2\|^{\frac{1}{2}}$$

and

$$\beta = \max_j |b_j b_{j-1}|^{\frac{1}{2}} = \|A_R^2\|^{\frac{1}{2}} .$$

From the definitions of $\hat{L}$ and $\hat{R}$ and from the bounds on D given by Lemma 3.1, we note that if $\lambda \leq 1$, then

$$\|\hat{L}^2\| < \left(\frac{2\alpha}{1 + \sqrt{1-\lambda}}\right)^2$$

and

$$\|\hat{R}^2\| < \left(\frac{2\beta}{1 + \sqrt{1-\lambda}}\right)^2.$$

Theorem 3.3. Let λ, α, and β be as above (c.f.(3.5)). Assume that $\lambda \leq 1$, $2\alpha \leq (1 + \sqrt{1-\lambda})$, and $2\beta \leq (1 + \sqrt{1-\lambda})$. Then for APG

$$\|D^{(i)} - D\| < \left(\frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}}\right)^2 \|D^{(i-1)} - D\|, \quad i = 2,3,\ldots,ID;$$

$$\|\underline{f}^{(i)} - \underline{f}\| < \left(\frac{2\alpha}{1 + \sqrt{1-\lambda}}\right)^2 \|\underline{f}^{(i-1)} - \underline{f}\| + \delta_f, \quad i = 2,3,\ldots,IF;$$

$$\|\underline{x}^{(i)} - \underline{x}\| < \left(\frac{2\beta}{1 + \sqrt{1-\lambda}}\right)^2 \|\underline{x}^{(i-1)} - \underline{x}\| + \delta_x, \quad i = 2,3,\ldots,IX.$$

For $i = 1$ the error reduction factors are not squared.

Proof. The inequalities are immediate from Theorem 3.2 and inequality (3.10) using the above bounds on $\|\hat{L}^2\|$ and $\|\hat{R}^2\|$. ∎

The following definitions allow us to bound all the quantities which occur in APG. Let

$$\alpha^* = \max_j |a_j| = \|A_L\|,$$

and

$$\beta^* = \max_j |b_j| = \|A_R\|.$$

Note that $\alpha \leq \alpha^*$. From the definitions of L and R and from the bounds on D given by Lemma 3.1, we note that if $\lambda \leq 1$, then

$$\|L\| < \frac{2\alpha^*}{1 + \sqrt{1-\lambda}}$$

and

$$\|R\| < \frac{2\beta^*}{1 + \sqrt{1-\lambda}}.$$

Similarly,

$$\|\hat{L}\| < \frac{2\alpha^*}{1 + \sqrt{1-\lambda}}$$

and

$$\|\hat{R}\| < \frac{2\beta^*}{1 + \sqrt{1-\lambda}}.$$

If we assume that

$$\alpha^* < \frac{1 + \sqrt{1-\lambda}}{2} \quad \text{and} \quad \beta^* < \frac{1 + \sqrt{1-\lambda}}{2}$$

so that $\|L\| < 1$ and $\|R\| < 1$, then from the definition of $\underline{f}$ and $\underline{x}$ given by (3.11) and (3.12), the above bounds on $\|L\|$ and $\|R\|$ allow us to conclude

$$\|\underline{f}\| < \frac{\|\underline{c}\|}{1 - \|L\|} \tag{3.15}$$

and

$$\|\underline{x}\| < \frac{2\|\underline{f}\|}{1 - \|R\|}. \tag{3.16}$$

Analogous bounds hold for $\|\underline{f}^{(i)}\|$ and $\|\underline{x}^{(i)}\|$.

The importance of (3.6),(3.15), and (3.16) is as follows. By the bounds on D, $\underline{f}$, and $\underline{x}$ we can bound the initial errors in $D^{(0)}, \underline{f}^{(0)}$, and $\underline{x}^{(0)}$. For example, if $\underline{f}^{(0)} = \underline{0}$ then $\|\underline{f}^{(0)} - \underline{f}\| = \|\underline{f}\|$. Hence all quantities occurring in Theorem 3.3 can be bounded before the computation simply by inspection of the matrix A and the right hand side $\underline{c}$. Thus we can calculate, a priori, the number of iterations ID,IF, and IX which will be sufficient to achieve a given accuracy in the computed solution $\underline{x}$.

4. Variations of APG.

In this section we discuss four variations of APG. The first three are motivated by the observation that the divisions in the D-iteration might prove to be disproportionately expensive on some computers, so we include alternative "divisionless" algorithms. For example, on Illiac division requires about ten times as much time as addition. Division is also expensive when the matrix components themselves are matrices rather than scalars. Therefore these variations may be of interest for the block tridiagonal case which is discussed in Section Six.

4.1 The N-iteration.

We can eliminate the division in the iteration $D^{(i)} = I - A_L(D^{(i-1)})^{-1}A_R$ by approximating the diagonal matrix $(D^{(i-1)})^{-1}$ by a diagonal matrix $N^{(i-1)}$, then updating our approximation. If we use one step of Newton iteration to approximate the matrix inverse we obtain for all i

$$\bar{D}^{(i)} = I - A_L N^{(i-1)} A_R,$$

$$N^{(i)} = N^{(i-1)}(2I - \bar{D}^{(i)} N^{(i-1)}). \tag{4.1}$$

These two equations can be combined as

$$N^{(i)} = N^{(i-1)}(2I - (I - A_L N^{(i-1)} A_R) N^{(i-1)}). \tag{4.2}$$

Let $N^{(i)} = \mathrm{Diag}(n_1^{(i)}, n_2^{(i)}, \ldots, n_n^{(i)})$. Then in component form (4.2) is

$$n_1^{(i)} = 1 \quad \text{for all } i,$$

$$n_j^{(i)} = n_j^{(i-1)}(2 - (1 - a_j b_{j-1} n_{j-1}^{(i-1)}) n_{j-1}^{(i-1)}), \quad \text{for } j > 1 .$$

As before, let $\lambda = \max_j |4a_j b_{j-1}|$.

Theorem 4.1. Let $\lambda \leq 1$ and $N^{(0)} = I$. Then

$$\|N^{(i)}\| < \frac{2}{1 + \sqrt{1-\lambda}}$$

and

$$\|N^{(i)} - D^{-1}\| < \frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}} \|N^{(i-1)} - D^{-1}\| + \|D\| \cdot \|N^{(i-1)} - D^{-1}\|^2, \text{ for all } i. \tag{4.3}$$

Proof. We write (4.1) in component form. For all i we have

(4.4) $$\bar{d}_j^{(i)} = 1 - a_j b_{j-1} n_{j-1}^{(i-1)} \quad \text{for} \quad j > 1$$

and

(4.5) $$n_j^{(i)} = n_j^{(i-1)}(2 - \bar{d}_j^{(i)} n_j^{(i-1)}) \quad \text{for} \quad j > 1.$$

From (4.5) it follows that

(4.6) $$n_j^{(i)} \leq \frac{1}{\bar{d}_j^{(i)}}$$

and as in the proof of Lemma 3.1, we have, using (4.4) and (4.6),

$$\bar{d}_j^{(i)} > \frac{1 + \sqrt{1-\lambda}}{2} .$$

Hence

$$n_j^{(i)} < \frac{2}{1 + \sqrt{1-\lambda}}$$

and the bound on the norm of $N^{(i)}$ is established. After some algebraic manipulation,

$$N^{(i)} - D^{-1} = N^{(i-1)} A_L (N^{(i-1)} - D^{-1}) A_R N^{(i-1)} - D(N^{(i-1)} - D^{-1})^2$$

and the rest of the theorem follows from the bound on $N^{(i)}$. ∎

Thus asymptotically, the quadratic term in (4.3) becomes negligible and this iteration behaves like the iteration for D. The reader is referred to Section Five for a numerical example showing this effect. Also, the same technique for squaring the error reduction factor in the iteration for D can be applied mutatas mutandi to (4.2) and it is this accelerated form that will be used when we wish to avoid division in the LDU factorization. We shall refer to this as the N-iteration. Note that $N^{(IN)}$ may be used in place of $(D^{(ID)})^{-1}$ in (3.4) to obtain the multiplicative decomposition of A, and hence the $\underline{f}$ and $\underline{x}$ iterations of APG can still be used to solve the system $A\underline{x} = \underline{c}$.

Global convergence of the N-iteration is not implied by (4.3). It can be shown, however, that with the starting value $N^{(0)} = I$ the N-iteration will converge to D^{-1}. Our proof of this is a tedious case analysis and is omitted.

4.2 Other variations.

Our second iteration, despite its slower convergence rate, is faster in execution time for large dominance cases because of its smaller operation count per step. Note that the multiplicative splitting of A involves D^{-1} rather than D. We

can write

$$D^{-1} = I + A_L D^{-1} A_R D^{-1}$$

which is immediate from multiplying both sides of (3.1) by D^{-1} and rearranging the equation. This suggests the iteration

$$W^{(i)} = I + A_L W^{(i-1)} A_R W^{(i-1)}$$

which can be shown to have an error reduction rate of $1 - \sqrt{1-\lambda}$, where λ is as above.

The third variation may be derived by viewing $d_j^{(i)}$ in (2.1) as a continued fraction that equals its k^{th} convergent for $k \geq i$. Using the well known relations connecting the numerators and denominators of three consecutive convergents (c.f. Khovanskii [1963]), we have the associated iteration:

$$e_j^{(-1)} = 1; \; e_j^{(0)} = 1 - a_j b_{j-1} \; .$$

Then

$$e_j^{(i)} = e_j^{(i-1)} - a_{j-i} b_{j-i-1} e_j^{(i-2)} \; .$$

If $d_j^{(0)} = 1 - a_j b_{j-1}$, then for exact arithmetic

$$d^{(i)} \equiv \frac{e_j^{(i)}}{e_{j-1}^{(i-1)}} \text{ , for all } i \text{ and } j.$$

In particular, we can recover $d_j^{(ID)}$ from $e_j^{(ID)}$ and $e_{j-1}^{(ID-1)}$ using one division at the end; thus the "rate of convergence" of this variation is the same as that of the iteration for D, (3.8). On the other hand, for $a_j b_{j-1} > 0$ the $e_j^{(i)}$ approach zero and scaling may be necessary.

Finally, we note that using cyclic reduction on a tridiagonal matrix produces tridiagonal matrices whose diagonal dominance is no less than the original (c.f. Stone [1974] and Heller [1974]). Since iterative methods become more attractive as the dominance of a matrix increases, it may be feasible to switch to an iterative method after a number of steps of cyclic reduction rather than using cyclic reduction to reduce the system to a single equation.

5. Numerical Behavior of APG.

In this section we discuss various aspects of the numerical behavior of APG in order to illuminate the theory of Sections Three and Four. For this purpose we report on the system $A\underline{x} = \underline{c}$ of size 5000 where A is such that all off-diagonal non-zero elements are 0.48 ($\lambda = 0.9216$) and $\underline{c}$ is chosen such that all components of $\underline{x}$ will be unity. The following starting values were chosen: $D^{(0)} = N^{(0)} = I$, $\underline{f}^{(0)} = \underline{c}$, and $x^{(0)} = \hat{D}^{-1}\underline{f}^{(IF)}$.

Theorem 3.3 predicts that the error reduction factor for the D-iteration will be no worse than 0.3164. On the first iteration, the observed error reduction factor was 0.36, falling to 0.2404 on the second iteration, and thereafter increasing to the bound 0.3163 observed on the eighth iteration. The poor error reduction in the first iteration was due to the squaring effect of the re-ordering on the D-iteration not being felt until after the first "half" iteration; that is, the error reduction factor was determined by odd-indices of D, which, by equation (3.9), did not benefit from the re-ordering until the second iteration.

The N-iteration began with an error reduction factor of 0.5904, which improved steadily as the algorithm progressed and the quadratic term of (4.3) became negligible. By the eighth iteration the error reduction factor had become 0.3169.

The intermediate vector $\underline{f}^{(IF)}$ was calculated such that $\|\underline{f}^{(IF)} - \underline{f}\| < 8.7 \times 10^{-6}$. This inaccuracy in $\underline{f}^{(IF)}$, together with the inaccuracy in $D^{(ID)}$, effectively limited the final accuracy possible for $\underline{x}^{(i)}$ such that $\|\underline{x}^{(i)} - \underline{x}\| \geq 2.7 \times 10^{-5}$, for all i. The error reduction factor for the $\underline{x}$-iteration was predicted to be 0.5625 (setting $\delta_f = 0$ in Theorem 3.3 for simplicity). The actual error reduction factor was observed to be slightly better throughout the iteration.

In summary, this section has demonstrated the following points about APG. The error reduction bounds of Theorem 3.3 are generally pessimistic, aside from the first iteration where the squared aspect of the acceleration is ineffective (the un-squared bound holds, however). They are, however, approached asymptotically. The error reduction for the N-iteration approaches that of the D-iteration as the quadratic term becomes negligible in comparison with the linear term in the error reduction bound. Finally, approximating D and $\underline{f}$ limits the final accuracy obtainable in $\underline{x}$.

6. APG for Block Tridiagonal Systems.

The APG algorithm has a natural extension to the case of a block tridiagonal system. If the blocks are dense (or the fill-in for sparse blocks can be tolerated), then APG may be attractive. One important special case of this situation is a pentadiagonal system viewed as block tridiagonal. In this section we confine ourselves to proving the main convergence results for block APG and do not compare it with other direct or iterative methods. Again, for the analysis, it is no restriction to assume that the matrix has been normalized to have identity blocks on the diagonal.

6.1 Iterative Block LDU Decomposition.

If we let the diagonal blocks of the block diagonal matrix D be denoted by $D_1, D_2, \ldots, D_n$ then (2.1) becomes

$$D_j = I - A_j D_{j-1}^{-1} B_{j-1}$$

and the associated iteration is

$$D_j^{(i)} = I - A_j (D_{j-1}^{(i-1)})^{-1} B_{j-1}. \tag{6.1}$$

Just as Lemma 3.1 was the key to the analysis of the APG algorithm, the following definition and lemma are crucial to the block case. For this section only, let

$$\lambda = \max_j 4\|A_j\|\|B_{j-1}\| .$$

Lemma 6.1. Let $\lambda \leq 1$. Then A is non-singular, and for all j, $\|I - D_j\| < \frac{1 - \sqrt{1-\lambda}}{2}$ and $\|D_j^{-1}\| < \frac{2}{1 + \sqrt{1-\lambda}}$. Furthermore, if $\|I - D_j^{(0)}\| < \frac{1 - \sqrt{1-\lambda}}{2}$ and $D_1^{(i)} = I$ for all i, then $\|I - D_j^{(i)}\| < \frac{1 - \sqrt{1-\lambda}}{2}$ and $\|(D_j^{(i)})^{-1}\| < \frac{2}{1 + \sqrt{1-\lambda}}$ for all i and j.

Proof. The bounds are proved by induction. $D_1 = I$, $\|I - D_1\| = 0$, and $\|D_1^{-1}\| = 1$ satisfy the bounds. Since $I - D_j = A_j D_{j-1}^{-1} B_{j-1}$ we have

$$\|I - D_j\| \leq \|A_j\|\|D_{j-1}^{-1}\|\|B_{j-1}\| \leq \frac{\|A_j\|\|B_{j-1}\|}{1 - \|I-D_{j-1}\|} < \frac{1 - \sqrt{1-\lambda}}{2}$$

and also

$$\|D_j^{-1}\| \leq \frac{1}{1 - \|I-D_j\|} \leq \frac{2}{1 - \sqrt{1-\lambda}} .$$

The bounds on the iterates $D_j^{(i)}$ are established similarly. ∎

Thus the condition $\lambda \leq 1$ and $D_j^{(0)}$ being "near" the identity are sufficient to insure that all $D_j^{(i)}$ are invertible and the iteration (6.1) is well defined. By using this lemma in the same way that Lemmas 3.1 and 3.2 were used in the proof of Theorem 3.1, it is easy to see that

$$\|D^{(i)} - D\| < \frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}} \|D^{(i-1)} - D\|$$

and hence that the iteration (6.1) leads to the block LDU factorization of A. The condition $\lambda < 1$ also appears in Richtmyer and Morton [1967], page 279, as one assumption to show that A is non-singular. Varah [1972] proves a related theorem.

6.2 The Block APG Algorithm.

Given a block LDU factorization of A, one can solve the attendant block bidiagonal systems iteratively as in the scalar case. All three iterations can be accelerated by reordering the computation. We give the component form of the block APG algorithm:

Let $D^{(0)}, \underline{f}^{(0)}, \underline{x}^{(0)}$ be given. Then

1. Set $D_1^{(i)} = I$ for all i.

   For $i = 1,2,\ldots,ID$, let

$$D_j^{(i)} = I - A_j (D_{j-1}^{(i-1)})^{-1} B_{j-1}, \quad j \text{ even},$$

$$D_j^{(i)} = I - A_j (D_{j-1}^{(i)})^{-1} B_{j-1}, \quad j \text{ odd and } > 1.$$

   Define $\hat{D}_j = D_j^{(ID)}$, $\hat{L}_j = A_j \hat{D}_{j-1}^{-1}$, $\hat{R}_j = \hat{D}_j^{-1} B_j$.

2. Set $\underline{f}_1^{(i)} = \underline{c}_1$ for all i.

   For $i = 1,2,\ldots,IF$, let

$$\underline{f}_j^{(i)} = \underline{c}_j - \hat{L}_j \underline{f}_{j-1}^{(i-1)}, \quad j \text{ even},$$

$$\underline{f}_j^{(i)} = \underline{c}_j - \hat{L}_j \underline{f}_{j-1}^{(i)}, \quad j \text{ odd and } > 1.$$

3. Define $\hat{\underline{g}}_j = \hat{D}_j^{-1} \underline{f}_j^{(IF)}$ for all j.

   Set $\underline{x}_n^{(i)} = \hat{\underline{g}}_n$ for all i.

   For $i = 1,2,\ldots,IX$, let

$$\underline{x}_j^{(i)} = \hat{\underline{g}}_j - \hat{R}_j \underline{x}_{j+1}^{(i-1)}, \quad j \text{ odd},$$

$$\underline{x}_j^{(i)} = \hat{\underline{g}}_j - \hat{R}_j \underline{x}_{j+1}^{(i)}, \quad j \text{ even and } < n.$$

To derive bounds on the rate of convergence of the iterations we need the following definitions. For this section only, let

$$\alpha = \max_j (\|A_j\|\|A_{j-1}\|)^{\frac{1}{2}}$$

and

$$\beta = \max_j (\|B_j\|\|B_{j-1}\|)^{\frac{1}{2}}.$$

Then we have the following.

Theorem 6.1. Assume $\lambda \leq 1$, $2\alpha < 1 + \sqrt{1-\lambda}$, and $2\beta < 1 + \sqrt{1-\lambda}$. Then for the block APG algorithm,

$$\|D^{(i)} - D\| < \left(\frac{1 - \sqrt{1-\lambda}}{1 + \sqrt{1-\lambda}}\right)^2 \|D^{(i-1)} - D\|, \quad i = 2,3,\ldots,ID,$$

$$\|\underline{f}^{(i)} - \underline{f}\| < \left(\frac{2\alpha}{1 + \sqrt{1-\lambda}}\right)^2 \|\underline{f}^{(i-1)} - \underline{f}\| + \delta_{\underline{f}}, \quad i = 2,3,\ldots,IF,$$

$$\|\underline{x}^{(i)} - \underline{x}\| < \left(\frac{2\beta}{1 + \sqrt{1-\lambda}}\right)^2 \|\underline{x}^{(i-1)} - \underline{x}\| + \delta_{\underline{x}}, \quad i = 2,3,\ldots,IX,$$

where $\delta_{\underline{f}}$ and $\delta_{\underline{x}}$ can be made arbitrarily small.

Proof. The proof is the same, mutatas mutandi, as that for Theorem 3.3, using Lemma 6.1 in place of Lemma 3.1. ∎

7. Parallel Algorithms Based on Additive Decompositions.

We briefly review the classical iterative algorithms for solving $A\underline{x} = \underline{c}$ based on an additive decomposition of A. These iterations are easily interpreted as parallel computations. If we write $A = I - J$, the Jacobi iteration takes the form $\underline{x}^{(i)} = \underline{c} + J\underline{x}^{(i-1)}$ for all i. In components,

$$x_j^{(i)} = c_j - a_j x_{j-1}^{(i-1)} - b_j x_{j+1}^{(i-1)}$$

for all j, where improper subscripts by convention denote zero elements.

By partitioning the $\underline{x}$ vector according to odd and even subscripts, we can accelerate the convergence in a manner similar to the iteration for D:

$$\begin{aligned} x_j^{(i)} &= c_j - a_j x_{j-1}^{(i-1)} - b_j x_{j+1}^{(i-1)}, \quad j \text{ odd}, \\ x_j^{(i)} &= c_j - a_j x_{j-1}^{(i)} - b_j x_{j+1}^{(i)}, \quad j \text{ even}. \end{aligned} \tag{7.1}$$

This doubles the rate of convergence; however, the problem is that unlike the iteration for D, where one could arbitrarily accelerate the iteration by a factor of k merely by rearranging the vectors so that indices equivalent modulo k are grouped together, here such a grouping modulo k yields a factor less than k for $k > 2$. For example, in a numerical experiment the vectors were rearranged modulo three; the observed error reduction term was not cubed but rather was raised to the power 2.2.

Another way of viewing (7.1) is that it is the block Gauss-Seidel iteration applied to a permutation of the original system. In particular, let

$$A^* = \begin{pmatrix} I & A_{12} \\ A_{21} & I \end{pmatrix}$$

where I denotes the identity matrix, A_{12} is lower bidiagonal and A_{21} is upper bidiagonal. Specifically,

$$A_{12} = \begin{pmatrix} b_1 & & & \\ a_3 & b_3 & & \\ & a_5 & b_5 & \\ & & \ddots & \ddots \end{pmatrix}$$

and

$$A_{21} = \begin{pmatrix} a_2 & b_2 & & \\ & a_4 & b_4 & \\ & & a_6 & b_6 \\ & & & \ddots \end{pmatrix} .$$

If we reorder and partition the vectors $\underline{x}$ and $\underline{c}$ to be compatible with A, we may write the system as

$$\begin{pmatrix} I & A_{12} \\ A_{21} & I \end{pmatrix} \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix} = \begin{pmatrix} \underline{c}_1 \\ \underline{c}_2 \end{pmatrix} .$$

Using this ordering, we have the block Gauss-Seidel iteration

$$\underline{x}_1^{(i)} = \underline{c}_1 - A_{12}\underline{x}_2^{(i-1)} ,$$

$$\underline{x}_2^{(i)} = \underline{c}_2 - A_{21}\underline{x}_1^{(i)}$$

which, when expressed in components, is the same as (7.1).

The block SOR method is defined by

$$\underline{x}_1^{(i)} = \underline{x}_1^{(i-1)} + \omega(\underline{c}_1 - \underline{x}_1^{(i-1)} - A_{12}\underline{x}_2^{(i-1)}),$$

(7.2)

$$\underline{x}_2^{(i)} = \underline{x}_2^{(i-1)} + \omega(\underline{c}_2 - \underline{x}_2^{(i-1)} - A_{21}\underline{x}_1^{(i)}).$$

The two-cyclic Chebyshev method (henceforth referred to simply as Chebyshev) is defined similarly to (7.2), the only difference being that the ω changes with each iteration.

If we denote the spectral radius of $J = I - A$ by ρ, the error reduction factor of block Gauss-Seidel can be estimated by ρ^2. For block SOR, the optimal parameter $\omega = \dfrac{2}{1 + \sqrt{1-\rho^2}}$ yields

$$\|\underline{x}^{(i)} - \underline{x}\| \approx \frac{1 - \sqrt{1-\rho^2}}{1 + \sqrt{1-\rho^2}} \|\underline{x}^{(i-1)} - \underline{x}\|. \tag{7.3}$$

The same estimate holds for Chebyshev. Because of the asymptotic nature of these estimates, the above error reduction estimates are generally optimistic. They will, however, be used in an analytic comparison of the various parallel iterative methods for a model problem defined in the next section.

In the case where A is symmetric and positive definite the method of conjugate gradients may be applied. The formulation due to Reid [1972] expresses the algorithm in a form similar to the Chebyshev iteration, but with a different set of parameters which must be computed as part of the iteration.

8. Comparison of Methods.

In this section we compare the various methods presented in the paper. We begin with a brief explanation of operation counts for algorithms on a vector computer. We use this technique in our analysis, first for a model problem which in a certain sense serves as a worst-case problem for all the methods, and then for numerical examples. The conclusion is that generally APG is the best of these iterative methods.

8.1 Timing Considerations for Vector Computers.

The distinguishing characteristic of a vector computer is that it can operate homogeneously component by component on vectors of length n much faster than it can perform the same operation n times in succession on scalars. This may be accomplished by having multiple arithmetic units which perform the same operation on different operands, as in Illiac where there are sixty-four processing elements which execute the same instruction. Another method is by decomposing an operation into several stages, as in an assembly line, so that at any instant many vector components can be in different stages. This technique is called pipelining; the STAR processes vectors in such a fashion. Both techniques are sometimes combined, resulting in multiple pipeline arithmetic units which provide even greater speed.

The time required to operate on two vectors of length n in a vector computer can be estimated by $t\lceil\frac{n}{m}\rceil + s$ where m depends on the machine architecture (being 8 for the STAR and 64 for Illiac), and t and s depend on the particular operation. One can associate t with the time required to produce m results and s with the overhead associated with processing vectors. This is to be contrasted with the same number of arithmetic operations performed sequentially, whose time is estimated by $t'n$, where t' is the time associated with each operation. For vector instructions to be efficient, $\frac{t}{m}$ must be somewhat less than t'. On the STAR, they differ roughly by a factor of about ten; on Illiac, by a factor of sixty-four.

For convenience we will approximate the form of the timing estimates to be simply $\tau_{op}n + \sigma_{op}$. For example, a vector addition on the STAR takes $\frac{1}{2}n + 96$ clock cycles, and on Illiac $\frac{5n}{64}$ clock cycles for $n \geq 64$. When processing a vector of length n as k smaller vectors, the time required is $k(\frac{\tau_{op}n}{k} + \sigma_{op}) = \tau_{op}n + k\sigma_{op}$ rather than $\tau_{op}n + \sigma_{op}$. Thus there is some penalty associated with processing smaller vectors, but if the convergence rate accelerates significantly, it may be advisable to pay this additional cost.

When we apply this method of estimating the time required for each iteration, we find that APG requires

(8.1) $$(\tau_{\div} + \tau_{+})n + 2(\sigma_{\div} + \sigma_{+})$$

for each D-iteration, and

(8.2) $$(\tau_{x} + \tau_{+})n + 2(\sigma_{x} + \sigma_{+})$$

for each $\underline{f}$ and $\underline{x}$ iteration. Chebyshev, on the other hand, requires

(8.3) $$(3\tau_{x} + 4\tau_{+})n + 6\sigma_{x} + 8\sigma_{+}$$

for each iteration. Conjugate gradients requires

$$(5\tau_{x} + 3\tau_{+} + \tau_{ip})n + 10\sigma_{x} + 6\sigma_{+} + 2\sigma_{ip}$$

for each iteration, where ip is the inner product operation.

8.2 A Model Problem.

We consider a model problem where the diagonals have constant coefficients. This turns out to be a worst-case problem (in a sense discussed below) for all the algorithms, and is also analytically tractable. Let

$$A_{a,b,n} = \begin{pmatrix} 1 & b & & & \\ a & 1 & b & & \\ & a & 1 & b & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

where $A_{a,b,n}$ is of dimension n. For any matrix A with normalized diagonal such that $|A| \leq A_{a,b,n}$, $\rho(I - A) \leq \rho(I - A_{a,b,n})$. The classical methods of Section Seven have asymptotic error reduction estimates based on the spectral radius; the appropriate model problem with positive coefficients which dominates a given matrix provides a worst-case analysis of the behavior of the algorithm. The following monotonicity theorem, which generalizes a result of Cryer [1973], establishes a similar result for APG.

Theorem 8.1. Let A and A' be of the form (1.2), $|A| \leq A'$, and suppose that both A and A' satisfy the hypotheses of Theorem 3.2. Then $|L| \leq L'$ and $|R| \leq R'$. If $D^{(0)} \geq D'^{(0)}$, then $|L^{(i)}| \leq L'^{(i)}$ and $|R^{(i)}| \leq R'^{(i)}$, for all i.

Proof. By assumption $|A_L| \leq A'_L$ and $|A_R| \leq A'_R$, so we need only show $D \geq D'$ and $D^{(i)} \geq D'^{(i)}$. The proof procedes with an induction on j. Clearly $d'_1 = d_1 = 1$, and

if $d_{j-1} \geq d'_{j-1}$ then $d_j = 1 - \frac{a_j b_{j-1}}{d_{j-1}} \geq 1 - \frac{|a_j b_{j-1}|}{d_{j-1}} \geq 1 - \frac{a'_j b'_{j-1}}{d'_{j-1}} = d'_j$. The remainder is proved similarly. ∎

As a result of Theorem 8.1, the error reduction factors of Theorem 3.2 for A will be no worse than the corresponding factors for A'. The simplicity of the constant diagonal matrix $A_{a,b,n}$ together with the majorant properties, motivate choosing $A' = A_{a,b,n}$.

The spectral radius $\rho = \rho(I - A_{a,b,n})$ is $2|ab|^{\frac{1}{2}}\cos(\frac{\pi}{n+1}) = \lambda^{\frac{1}{2}}\cos(\frac{\pi}{n+1})$. Thus $\rho^2 = \lambda\cos^2(\frac{\pi}{n+1})$ and by comparing the error reduction rates of Chebyshev (the best of the iterative methods, see Varga [1962]) with APG (compare Theorem 3.3 with (7.3)) we see that the D-iteration should converge about twice as fast as Chebyshev. A simple calculation shows that if $a = b$ then the $\underline{f}$ and $\underline{x}$ iterations should converge at about the same rate as Chebyshev converges for the original system. From (8.1) - (8.3) it follows that each iteration of APG is about a third as time consuming as Chebyshev, so that we expect that the total time required for APG will be about the same as that required for Chebyshev.

What prevents us from a more precise comparison is the following consideration. The asymptotic bounds for Chebyshev are generally optimistic, while those for APG are conservative, so that APG should do better than we estimate. However, for the model problem $\rho^2 < \lambda$ so that the asymptotic bounds predict that Chebyshev will require fewer iterations. In fact, for general matrices A one can show that $\rho^2 < \lambda$ by using the Perron-Frobenius theory (together with a similarity transformation) to pass to a model problem which has the same value for λ but a larger spectral radius than the original matrix. For this majorizing matrix the result is true and hence it follows for the original matrix as well.

In summary, what the model problem shows is that in the worst case APG should behave like optimal Chebyshev (but without having to calculate optimal parameters). The numerical testing which follows confirms this observation, and indicates that over a wide range of problems, APG is at least as good, and sometimes strikingly better, than Chebyshev.

8.3 Numerical Testing.

The theory we present gives APG asymptotic error reduction factors slightly larger than Chebyshev. However, the error reduction factors for APG are strict upper

bounds while that for Chebyshev is an asymptotic lower bound and thus we have not derived a general theoretical conclusion. The numerical testing allows us to make some empirical observations. We find that APG is superior to the other iterations studied in this paper. In general it behaves like the optimal Chebyshev iteration, but with no optimal parameter needed. In addition, it is applicable to some systems where the theory of optimal Chebyshev iteration is inapplicable.

All testing was carried out on the Carnegie-Mellon University Computer Science Department's PDP-10 in APL. We estimated the total time to solve a system by the product of the number of iterations and the arithmetic complexity of each iteration. For timing estimates we used manufacturers' estimates (as of Fall, 1974) in the model described in Section 8.1. We neglected the cost of testing for convergence for all iterations. In addition, for APG we neglected the pre-processing associated with calculating the products $a_j b_{j-1}$, and for calculating $L = A_L D^{-1}$, $R = D^{-1} A_R$, and $\underline{g} = D^{-1}\underline{f}$. For Chebyshev we neglected the calculation of optimal parameters. For Illiac we neglected communication overhead as well. Since division is expensive on Illiac, we replaced the D-iteration of APG with the N-iteration when estimating times for Illiac. Unless otherwise noted, $\underline{c}$ was chosen such that all components of $\underline{x}$ were unity. We consistently chose the initial values $D^{(0)} = N^{(0)} = I$, $\underline{f}^{(0)} = \underline{c}$, and $\underline{x}^{(0)} = \hat{D}^{-1}\underline{f}(IF)$ for APG and $\underline{x}^{(0)} = \underline{c}$ for the other iterations.

8.3a The Model Problem.

We first consider the case of constant diagonals. The number of iterations required to reduce the initial error by 2^{-15} is given in the accompanying table. M1 is the case $a = b = 0.3$; M2, $a = b = 0.45$; M3, $a = b = 0.48$; M4, $a = b = 0.49$. The reason for the $\underline{f}$ iteration taking longer than the $\underline{x}$ iteration is that it must be continued to greater accuracy in order to achieve the desired accuracy in the $\underline{x}$-iteration: to an accuracy of 2^{-17} in the D and $\underline{f}$ iterations in M2 and to an accuracy of 2^{-18} in the D and $\underline{f}$ iterations in M3 and M4. For M5, $a = 0.49$ and $b = -0.49$; since all eigenvalues of I-A are pure imaginary, the Chebyshev iteration does not converge. M6 is the case $a = 0.1$ and $b = 0.5$. Numbers in parentheses indicate the estimates based on λ and ρ for the number of iterations necessary to reduce the initial error. By neglecting δ_f and δ_x we occasionally underestimated IF and IX in APG.

For all of these examples we estimated the total time for APG on the STAR would be less than the total time for Chebyshev; for example, for M4, APG would require

TABLE

Numerical Comparisons of Iterative Methods

Problem Name	APG				G-S	Cheb	λ	$\rho^2(I-A)$
	D	N	f	x				
M1	3(4)	4	6(5)	6(5)	11(10)	6(5)	0.36	0.3529
M2	7(7)	8	14(13)	12(12)	50(46)	13(11)	0.81	0.7940
M3	11(11)	13	24(22)	19(19)	128(103)	21(17)	0.9216	0.9034
M4	15(16)	17	36(31)	27(26)	258(256)	29(26)	0.9604	0.9509
M5	4(16)	5	8(31)	7(26)	122(256)	diverges	0.9604	0.9509
M6	3(3)	3	3(3)	9(9)	11(7)	9(4)	0.2	0.1960
R1	9(11)	10	17(21)	15(18)	77(75)	17(14)	0.9612	0.8691
R2	7(11)	8	12(21)	12(18)	48(45)	diverges	0.9612	0.7907
R3	5(11)	6	9(21)	8(18)	43(56)	diverges	0.9612	0.8319
R4	4(11)	4	7(21)	7(18)	48(75)	diverges	0.9612	0.8691
R5	5(7)	6	8(13)	8(12)	22(21)	9(7)	0.8326	0.5958

Problems M1 through M6 are described in Section 8.3a, R1 through R5 in Section 8.3b. Numbers under APG, G-S (Gauss-Seidel) and Cheb (Chebyshev) indicate the number of iterations required; numbers in parentheses are estimates based on λ, α, and β for APG and ρ for G-S and Cheb. The quantities λ and ρ^2 are given for comparison.

132n + 26,000 cycles while Chebyshev would take 145n + 30,000 cycles. On Illiac the situation would be reversed, with APG requiring 21.8n cycles and optimal Chebyshev requiring 19.9n cycles.

The conjugate gradient method was also applied to several of these problems (M1 through M4). It required slightly fewer iterations compared to Chebyshev, but the difference was not significant if $\rho(I - A)$ is known. When $\rho(I - A)$ is unknown and must be estimated as part of the Chebyshev iteration, conjugate gradients may be more effective in terms of number of iterations. For example, conjugate gradients required 294n + 65,000 cycles for M4 on the STAR. The inner product was responsible for the doubling of the time; however, the estimation of optimal parameters for Chebyshev requires inner products as well.

8.3b Random Coefficients.

A model problem with $n = 100$, $a = b = 0.48$ was perturbed by subtracting a uniformly distributed random quantity X, where $0 \leq X \leq 0.03$, from the off-diagonal elements. Call this problem R1. Next, one third of the lower diagonal elements were negated at random. Call this problem R2. Then an additional one half of the positive elements of the lower diagonal were negated, creating R3. Finally, R4 was the result of negating all lower diagonal elements. The results are presented in the Table. For all cases $\lambda = 0.9612$, as for the model problem M3. For APG the initial error was reduced by 2^{-17} for the D and $\underline{f}$ iterations and by 2^{-15} for the $\underline{x}$ - iteration. For Gauss-Seidel and Chebyshev the initial error was reduced by a factor of 2^{-15}. The relationship between M3 and R1 is that of a model problem which dominates a matrix with varying coefficients. The remaining random coefficient matrices illustrate the effect of negative quantities in the convergence of APG. The problems R1-R4 illustrate the conclusions of Theorem 8.1.

A similar problem, R5, was created by using larger perturbations so that all off-diagonal elements fell in the range 0.23 to 0.48 with $\lambda = 0.8326$. In APG, the initial error was reduced by a factor of 2^{-16} for D and 2^{-15} for $\underline{f}$ and $\underline{x}$. In Gauss-Seidel and Chebyshev, the initial error was reduced by 2^{-15}.

8.3c A Boundary Value Problem.

The boundary value problem on $[0,\pi]$ given by

$$-x'' + 1000\sin(t)x = \cos(t)(1000\sin(t) + 1)$$

with $x(0) = 1$ and $x(\pi) = -1$ has an analytic solution $x(t) = \cos(t)$. The problem was discretized with uniform mesh spacing of $\Delta t = \frac{\pi}{101}$ to generate the tridiagonal system

$$\frac{-x_{i-1} + 2x_i - x_{i+1}}{\Delta t^2} + 1000 \sin(t_i) x_i = \cos(t_i)(1000 \sin(t_i) + 1)$$

where $t_i = i\Delta t$. The equations were normalized with the result that the off-diagonal elements fell within the interval (-0.4926,-0.3369). If the discretized problem were solved exactly, it would agree with the analytic solution to six decimal places at the mesh points.

For this example, $\lambda = 0.9564$ and $\rho(I - A) = 0.8958$. The Chebyshev iteration took thirteen iterations and Gauss-Seidel required forty-nine to reduce the initial error by 2^{-15} while APG took seven, twelve, and ten iterations for the D,$\underline{f}$, and $\underline{x}$ iterations to achieve a comparable error reduction in the $\underline{x}$ iteration. On the STAR the times would be 50.5n + 15,000 cycles for APG, 65n + 23,000 cycles for Chebyshev and 147n + 50,000 cycles for Gauss-Seidel.

8.3d Counterexamples.

The next two examples are included for mathematical completeness. The first shows that the sufficient condition $\lambda \leq 1$ is not necessary; the second that diagonal dominance is not necessary either.

A model problem with $n = 50$ and $a = b = 0.4$ was perturbed by setting $a_{26} = a_{27} = b_{25} = b_{26} = 0.6$. For this case $\lambda = 1.44$ so the APG theory does not apply. The spectral radius of $I - A$ is 0.962 and the system has a solution for which Chebyshev took twenty iterations and Gauss-Seidel 138 iterations to reduce the initial error by 2^{-15}. The APG algorithm took seven, eleven and ten iterations for the D,$\underline{f}$, and $\underline{x}$ iterations to achieve comparable error reduction in the $\underline{x}$ iteration. On the STAR the times would be 49n + 9772 cycles for APG and 105n + 22,000 cycles for Chebyshev. This example shows that the $\lambda \leq 1$ condition is not necessary. However, $d_{27} = 0.345 < \frac{1}{2}$, i.e. both the hypothesis and conclusion of Lemma 3.1 (which governs the convergence theory of APG) have been violated. The applicability of APG, and its desirability over Chebyshev, is greater than the theory we have presented would seem to indicate.

To show that diagonal dominance is not required for APG, consider the case where $a_j = b_j = \frac{1}{4}$ for j odd and $a_j = b_j = \frac{7}{8}$ for j even. Here $\lambda = \frac{7}{8}$ and the spectral radius of $I - A$ is $0.9354 \cos(\frac{\pi}{n+1})$ since this matrix is similar to $A_{a,a,n}$ with

$a = (\frac{7}{32})^{\frac{1}{2}} = 0.4677$. In order to pick up sixteen bits in the D and $\underline{f}$ iterations, and fifteen in the $\underline{x}$ iteration, APG required eight, seventeen, and fifteen iterations. In comparison, Gauss-Seidel required seventy-eight and Chebyshev sixteen.

8.4 Comparison to Direct Methods.

A direct method for the solution of $A\underline{x} = \underline{c}$ yields an answer in a time dependent only on the size of the system. This is in contrast to the iterative methods where the time for an answer also depends on the desired accuracy, the rate of convergence and the initial estimates. Moreover, the choice of a method from either class is computer dependent. Analysis shows that there are crossover points between the two classes of methods.

In general, if λ and $\rho(I - A)$ are near unity, the iterations converge very slowly and a direct method is preferred. Lambiotte and Voigt [1974] report that cyclic reduction is the most efficient direct method for the STAR for systems of size greater than approximately 120. Based on our test cases, cyclic reduction should be used for large n and when $\lambda > 0.8$ and good initial values for APG are not available. For small n, Gaussian Elimination should be used. In particular, the model problem $A_{0.5,0.5,n}$, which arises from a discretization of Poisson's equation in one dimension, should be solved using a direct method. The error reduction estimates in Theorem 3.3 are unity and the predicted convergence of APG does in fact occur numerically, albeit slowly. In fact, the error reduction is so slow that a finiteness property of APG is operative: it is easy to show that for $i \geq \frac{n}{2}$, $D^{(i)} = D$. Also if $D^{(ID)} = D$ then for $i \geq \frac{n}{2}$, $\underline{f}^{(i)} = \underline{f}$; and if $D^{(ID)} = D$ and $\underline{f}^{(IF)} = \underline{f}$ then for $i \geq \frac{n}{2}$, $\underline{x}^{(i)} = \underline{x}$.

For systems such as M1 with $n = 1000$, APG is much better than cyclic reduction, requiring (on the STAR) 33,000 cycles versus 40,000 for Chebyshev and 69,000 for cyclic reduction. In general, for systems with some dominance APG will be faster than cyclic reduction. For systems of size 128 to 512, APG will take no longer than cyclic reduction when α and β are both less than 0.46.

8.5 Computational Considerations.

Two important computational aspects should be considered in comparing APG with the other methods. The first deals with the calculation of optimal parameters, the second with termination criteria. Neither of these was included in the above analysis; their consideration makes APG even more attractive than the other iterative methods.

For general tridiagonal systems Chebyshev requires the estimation of $\rho(I - A)$. This estimation is usually in terms of two inner products involving residuals. For the STAR the inner product operation is considerably more expensive than a vector addition (by a factor of about eight). The result is that an inner product requires about as much time as the Chebyshev iteration for tridiagonal systems. This has a deleterious effect on the actual performance of Chebyshev on general systems. Our analysis which showed APG as roughly equivalent to Chebyshev with respect to total execution time ignored this problem by calculating $\rho(I - A)$ before Chebyshev began and by excluding this calculation from the timing estimate. The APG algorithm, of course, needs no optimal parameter estimation. But, on the other hand, APG requires the termination of three iterations whereas Chebyshev requires only one termination.

The above analysis also ignored the cost of checking to see if termination criteria for the iterations had been met. Depending upon the implementation strategy, testing for convergence can be quite expensive on vector computers; in any case, because of the arithmetic simplicity of these iterations for tridiagonal systems, convergence tests are not negligible. In striking contrast, the steps of APG, although presented as iterations, have readily calculable a priori upper bounds for the number of iterations required (and also for the degree of accuracy needed for intermediate quantities). In fact, this can be done by table look-up.

Acknowledgements

We wish to thank N. Madsen, LLL, H. Wozniakowski, CMU and University of Warsaw, and L. Hageman, Westinghouse Electric Corporation - Bettis, for comments on the manuscript.

REFERENCES

Cryer, C. W., "The numerical solution of boundary value problems for second order functional differential equations by finite differences", Numer. Math., 20 (1973), pp. 288-299.

Fiedler, M., and V. Ptak, "Some generalizations of positive definiteness and monotonicity", Numer. Math., 9(1966), pp. 163-172.

Heller, D. E., "Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems", ICASE, Carnegie-Mellon University Report, 1974.

Khovanskii, A.N., The application of continued fractions and their generalizations to problems in approximation theory, P. Noordhoff, Groningen, Netherlands, 1963.

Lambiotte, J. J., Jr., and R. G. Voigt, "The solution of tridiagonal linear systems on the CDC STAR-100 computer", ICASE Report, 1974.

Lynn, M.S., "Some infra-max bounds for the spectral radii of splittings of H-matrices", Numer. Math., 5(1963), pp. 152-174.

Ostrowski, A., "Über die determinanten mit überwiegender Hauptdiagonale", Comm. Math. Helv., 10(1937), pp. 69-96.

Reid, J. K., "The use of conjugate gradients for systems of linear equations possessing "Property A"", SIAM J. Numer. Anal., 9(1972), pp. 325-332.

Richtmyer, R. D., and K. W. Morton, Difference methods for initial-value problems, Interscience, New York, 1967.

Stone, H. S., "An efficient parallel algorithm for the solution of a tridiagonal linear system of equations", Journal ACM, 20(1973), pp. 27-38.

———————, "Parallel tridiagonal solvers", Digital Systems Laboratory, Stanford University, 1974.

Traub, J. F., "Iterative solution of tridiagonal systems on parallel or vector computers", in Complexity of Sequential and Parallel Numerical Algorithms, J. F. Traub, ed., Academic Press, New York, 1973, pp. 49-82.

Varah, J. M., "On the solution of block tridiagonal systems arising from certain finite-difference equations", Math. Comp., 26(1972), pp. 859-868.

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

Young, D. M., "Second-degree iterative methods for the solution of large linear systems", Journal Approx. Th., 5(1972), pp. 137-148.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ACCELERATED ITERATIVE METHODS FOR THE SOLUTION OF TRIDIAGONAL SYSTEMS ON PARALLEL COMPUTERS	5. TYPE OF REPORT & PERIOD COVERED Interim 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) D. E. Heller, D. K. Stevenson, J. F. Traub	8. CONTRACT OR GRANT NUMBER(s) N00014-67-A-0010 NR 044-422
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Department of Computer Science Pittsburgh PA 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington VA 22217	12. REPORT DATE December 1974 13. NUMBER OF PAGES 37
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We consider iterative methods for the solution of tridiagonal systems and present a new iteration whose rate of convergence is comparable to that of the optimal two-cyclic Chebyshev iteration but which does not require the calculation of optimal parameters. The theory has a natural extension to block tridiagonal systems. Numerical experiments suggest that on a parallel computer this new algorithm is the best of the iterative algorithms we consider.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)