S U326 P30-44

A GENERALIZED CONJUGATE GRADIENT METHOD FOR THE NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

by

Paul Concus
Gene H. Golub
Dianne P. O'Leary

STAN-CS-76-533
JANUARY 1976

Prepared for
The Energy Research and Development Administration
under Contract No. E(04-3) 326 PA No. 30

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

A GENERALIZED CONJUGATE GRADIENT METHOD FOR THE NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

by

Paul Concus
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Gene H. Golub
Computer Science Department
Stanford University
Stanford, CA 94305

Dianne P. O'Leary
Dept. of Mathematics
University of Michigan
Ann Arbor, MI 48104

Also issued as Lawrence Berkeley Laboratory Report 4604, University of California, Berkeley.

A GENERALIZED CONJUGATE GRADIENT METHOD FOR THE NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

Paul Concus
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Gene H. Golub
Computer Science Dept.
Stanford University
Stanford, CA 94305

Dianne P. O'Leary
Dept. of Mathematics
University of Michigan
Ann Arbor, MI 48104

ABSTRACT

We consider a generalized conjugate gradient method for solving sparse, symmetric, positive-definite systems of linear equations, principally those arising from the discretization of boundary value problems for elliptic partial differential equations. The method is based on splitting off from the original coefficient matrix a symmetric, positive-definiteonethat corresponds to a more easily solvable system of equations, and then accelerating the associated iteration using conjugate gradients. Optimality and convergence properties are presented, and the relation to other methods is discussed. Several splittings for which the method seems particularly effective are also discussed, and for some, numerical examples are given.

A GENERALIZED CONJUGATE GRADIENT METHOD FOR THE NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

Paul Concus
Lawrence Berkeley Laboratory
University of California
Berkeley,. CA 94720

Gene H. Golub
Computer Science Dept.
Stanford University
, Stanford, CA 94305

Dianne P. O'Leary
Dept. of Mathematics
University of Michigan
Ann Arbor, MI 48104

0. INTRODUCTION

In 1952, Hestenes and Stiefel [0] proposed the conjugate gradient method (CG) for solving the system of linear algebraic equations

$$A\underset{\sim}{x} = \underset{\sim}{b} ,$$

where A is an n × n,symmetric, positive-definite matrix. This elegant method has as one of its important properties that in the absence of round-off error the solution is obtained in at most n iteration steps. Furthermore, the entire matrix A need not be stored as an array in memory; at each stage of the iteration it is necessary to compute only the product $A\underset{\sim}{z}$ for a given vector $\underset{\sim}{z}$.

Unfortunately the initial interest and excitement in CG was dissipated, because in practice the numerical properties of the algorithm differed from the theoretical ones; viz. even for small systems of equations ($n \leq 100$) the algorithm did not necessarily terminate in n iterations. In addition, for large systems of equations arising from the discretization of two-dimensional elliptic partial differential equations, competing methods such as successive overrelaxation (SOR) required only $O(\sqrt{n})$ iterations to achieve a prescribed accuracy [1]. It is interesting to note that in the proceedings of the Conference on Sparse Matrices and Their Applications held in 1971 [2] there is hardly any mention of the CG method.

In 1970, Reid [3] renewed interest in CG by giving evidence that the method could be used in a highly effective manner as an iterative procedure for solving large sparse systems of linear equations. Since then a number of authors have described the use of CG for solving a variety of problems (cf. [4], [5], [6], [7], [8]). Curiously enough, although CC was generally discarded during the sixties as a useful method for solving linear equations, except in conjunction with other methods [9], there was considerable interest in it for solving nonlinear equations (cf. [10]).

The conjugate·gradient method has a number of attractive properties when used as an iterative method:

(i) It does not require an estimation of parameters.

(ii> It takes advantage of the distribution of the eigenvalues of the iteration operator.

(iii> It requires fewer restrictions on the matrix A for optimal behavior than do such methods as SOR.

Our-basic view is that CG is most effective when used as an iteration acceleration technique6

In this paper, we derive and show how to apply a generalization of the CG method and illustrate it with numerical examples. Based on our investigations, we feel that the generalized CG method has the potential for widespread application in the numerical solution of boundary value problems for elliptic partial differential equations. Additional experience should further indicate how best to take full advantage of the method's inherent possibilities.

1. DERIVATION OF THE METHOD

Consider the system of equations

$$A\underset{\sim}{x} = \underset{\sim}{b} \ , \tag{1.1}$$

where A is an n × n, symmetric, positive-definite matrix and $\underset{\sim}{b}$ is a given vector. It is frequently desirable to rewrite (1.1) as

$$M\underset{\sim}{x} = N\underset{\sim}{x} + \underset{\sim}{c} \ , \tag{1.2}$$

where M is positive-definite and symmetric and N is symmetric. In § 4 we describe several decompositions of the form (1.2). We are interested in those situations for which it is a much simpler computational task to solve the system

$$M\underset{\sim}{z} = \underset{\sim}{d} \tag{1.3}$$

than it is to solve (1.1).

We consider an iteration of the form

$$\underset{\sim}{x}^{(k+1)} = \underset{\sim}{x}^{(k-1)} + \omega_{k+1} (\alpha_k \underset{\sim}{z}^{(k)} + \underset{\sim}{x}^{(k)} - \underset{\sim}{x}^{(k-1)}) \ , \tag{1.4}$$

where

$$M\underset{\sim}{z}^{(k)} = \underset{\sim}{c} - (M-N)\underset{\sim}{x}^{(k)} \ . \tag{1.5}$$

Many iterative methods can be described by (1.4); e.g. the Chebyshev semi-iterative method and the Richardson second order method (cf. [11]). The generalized CG method is also of this form.

For the Richardson or Chebyshev methods, the optimal parameters (ω_{k+1},α_k) are given as simple, easy-to-compute functions of the smallest and largest eigenvalues of the iteration matrix $M^{-1}N$ [11]; thus good estimates of these eigenvalues are required for the methods to be efficient. The methods do not take into account the values of any of the interior eigenvalues of the iteration matrix.

The CG method, on the other hand, needs no a priori information on the extremal eigenvalues and does take into account the interior ones, but at a cost of increased computational requirements for evaluating ω_{k+1} and α_k. In § 3, we describe a technique to provide directly from the CG method good estimates for the extreme eigenvalues of the iteration matrix.

From equations (1.4) and (1.5), we obtain the relation

$$M\underset{\sim}{z}^{(k+1)} = M\underset{\sim}{z}^{(k-1)} - \omega_{k+1}(\alpha_k(M-N)\underset{\sim}{z}^{(k)} + M(\underset{\sim}{z}^{(k-1)} - \underset{\sim}{z}^{(k)})). \quad (1.6)$$

For the generalized CG method the parameters $\{\alpha_k,\omega_{k+1}\}$ are computed so that

$$\underset{\sim}{z}^{(p)^T} M\underset{\sim}{z}^{(q)} = 0 \quad (1.7)$$

$$\text{for } p \neq q \text{ and } p, q = 0,1,\ldots,n-1.$$

Since M is n × n positive-definite, (1.7) implies that for some $k \leq n$

$$\underset{\sim}{z}^{(k)} = \underset{\sim}{0}$$

and hence

$$\underset{\sim}{x}^{(k)} = \underset{\sim}{x} \quad . \quad (1.8)$$

That is, the iteration converges in no more than n steps.

We derive the above result by induction. Assume

$$\underset{\sim}{z}^{(p)^T} M\underset{\sim}{z}^{(q)} = 0 \tag{1.9}$$

for $p \neq q$ and $p,q = 0,1,\ldots, k$.

Then if

$$\alpha_k = \underset{\sim}{z}^{(k)^T} M\underset{\sim}{z}^{(k)} / \underset{\sim}{z}^{(k)^T} (M-N)\underset{\sim}{z}^{(k)} , \tag{1.10}$$

there holds

$$\underset{\sim}{z}^{(k)^T} M\underset{\sim}{z}^{(k+1)} = 0 ,$$

and if

$$\omega_{k+1} = \left(1 - \alpha_k \frac{\underset{\sim}{z}^{(k-1)^T} N\underset{\sim}{z}^{(k)}}{\underset{\sim}{z}^{(k-1)^T} M\underset{\sim}{z}^{(k-1)}} \right)^{-1} \tag{1.11}$$

then

$$\underset{\sim}{z}^{(k-1)^T} M\underset{\sim}{z}^{(k+1)} = 0 .$$

We can simplify the above expression for ω_{k+1} as follows. From (1.6) we obtain

$$M\underset{\sim}{z}^{(k)} = M\underset{\sim}{z}^{(k-2)} - \omega_k (\alpha_{k-1} (M-N)\underset{\sim}{z}^{(k-1)} + M(\underset{\sim}{z}^{(k-2)} - \underset{\sim}{z}^{(k-1)})),$$

and then from (1.9)

$$\underset{\sim}{z}^{(k)^T} N\underset{\sim}{z}^{(k-1)} = \underset{\sim}{z}^{(k)^T} M\underset{\sim}{z}^{(k)} / (\omega_k \alpha_{k-1}) .$$

Since

$$\underset{\sim}{z}^{(k-1)^T} N\underset{\sim}{z}^{(k)} = \underset{\sim}{z}^{(k)^T} N\underset{\sim}{z}^{(k-1)} ,$$

it follows

$$\omega_{k+1} = \left(1 - \frac{\alpha_k}{\alpha_{k-1}} \frac{\underset{\sim}{z}^{(k)^T} M\underset{\sim}{z}^{(k)}}{\underset{\sim}{z}^{(k-1)^T} M\underset{\sim}{z}^{(k-1)}} \frac{1}{\omega_k} \right)^{-1}$$

Fran (1.6), for $j < k-1$

$$\underset{\sim}{z}^{(j)^T} M\underset{\sim}{z}^{(k+1)} = \alpha_k \omega_{k+1} \underset{\sim}{z}^{(j)^T} N\underset{\sim}{z}^{(k)} .$$

But,

$$M\underset{\sim}{z}^{(j+1)} = M\underset{\sim}{z}^{(j-1)} - \omega_{j+1}(\alpha_j (M-N)\underset{\sim}{z}^{(j)} + M(\underset{\sim}{z}^{(j-1)} - \underset{\sim}{z}^{(j)})) ,$$

so that

$$\underset{\sim}{z}^{(k)^T} N\underset{\sim}{z}^{(j)} = 0 .$$

Thus, since $N = N^T$,

$$\underset{\sim}{z}^{(j)^T} M\underset{\sim}{z}^{(k+1)} = 0 \quad \text{for} \quad j < k-1 .$$

Hence by induction we obtain (1.7) and (1.8).

The generalized CG method is summarized as follows.

Algorithm

Let $\underset{\sim}{x}^{(0)}$ be a given vector and arbitarily define $\underset{\sim}{x}^{(-1)}$. For $k = 0, 1, \ldots$

(1) Solve $M\underset{\sim}{z}^{(k)} = \underset{\sim}{c} - (M-N)\underset{\sim}{x}^{(k)}$.

(2) Compute

$$\alpha_k = \frac{\underset{\sim}{z}^{(k)^T} M\underset{\sim}{z}^{(k)}}{\underset{\sim}{z}^{(k)^T} (M-N)\underset{\sim}{z}^{(k)}} ,$$

$$\omega_{k+1} = \left(1 - \frac{\alpha_k}{\alpha_{k-1}} \; \frac{\underset{\sim}{z}^{(k)^T} M \underset{\sim}{z}^{(k)}}{\underset{\sim}{z}^{(k-1)^T} M \underset{\sim}{z}^{(k-1)}} \cdot \frac{1}{\omega_k} \right)^{-1}, \quad (k \geq 1) \; ,$$

$$\omega_1 = 1 \; .$$

(3) Compute

$$x^{(k+1)} = \underset{\sim}{x}^{(k-1)} + \omega_{k+1}(\alpha_k \underset{\sim}{z}^{(k)} + \underset{\sim}{x}^{(k)} - \underset{\sim}{x}^{(k-1)}) \; .$$

Note that the algorithm can be viewed as an acceleration **of the underlying f**irst order iteration $(\omega_{k+1} \equiv 1)$, $\underset{\sim}{x}^{(k+1)} = \underset{\sim}{x}^{(k)} + \alpha_k \underset{\sim}{z}^{(k)}$. As with other higher order methods, the storage requirements of the algorithm are greater than those of the underlying first order iteration being accelerated.

The algorithm presented above is given primarily for expository purposes. For actual computation, the following equivalent form can be more efficient in terms of **storage** [3].

<u>Algorithm</u> (alternative form)

Let $\underset{\sim}{x}^{(0)}$ be a given vector and arbitrarily define $\underset{\sim}{p}^{(-1)}$. For k = 0, 1, . . .

(1) solve $M\underset{\sim}{z}^{(k)} = \underset{\sim}{c} - (M-N)x^{(k)}$.

(2) Compute

$$b_k = \frac{\underset{\sim}{z}^{(k)^T} M \underset{\sim}{z}^{(k)}}{\underset{\sim}{z}^{(k-1)^T} M \underset{\sim}{z}^{(k-1)}} \; , \quad k \geq 1 \; ,$$

$$b_0 = 0 \, ,$$

$$\underset{\sim}{p}^{(k)} = \underset{\sim}{z}^{(k)} + b_k \underset{\sim}{p}^{(k-1)} \; .$$

(3) Compute

$$a_k = \frac{\underset{\sim}{z}^{(k)^T} M\underset{\sim}{z}^{(k)}}{\underset{\sim}{p}^{(k)^T}(M-N)\underset{\sim}{p}^{(k)}} \quad ,$$

$$\underset{\sim}{x}^{(k+1)} = \underset{\sim}{x}^{(k)} + a_k\underset{\sim}{p}^{(k)} \quad .$$

In the computation of the numerators of a_k and b_k one need not recompute $M\underset{\sim}{z}^{(k)}$, since it can be saved from step (1). Also, instead of computing the right hand side of step (1) explicitly at each iteration, it is often advantageous to compute it recursively from

$$[\underset{\sim}{c} - (M-N)\underset{\sim}{x}^{(k+1)}] = [\underset{\sim}{c} - (M-N)\underset{\sim}{x}^{(k)}] - a_k(M-N)\underset{\sim}{p}^{(k)} \quad , \qquad (1.12)$$

which equation is obtained from step (3). The quantity $(M-N)\underset{\sim}{p}^{(k)}$ appearing in (1.12) may be saved from the computation of a_k. Similar remarks hold for the algorithm in its first form as well. There is evidence that the use of (1.12) is no less accurate than use of the explicit computation (see [18], [3] for particular examples).

. The calculated vectors $\{\underset{\sim}{z}^{(k)}\ 3_{k=0}^{n}$ will not generally be M-orthogonal in practice because of rounding errors. One might consider forcing the newly calculated vectors to be M-orthogonal by a procedure such as Gram-Schmidt. However, this would require the storage of all the previously obtained vectors.

Our basic approach is to permit the gradual loss of orthogonality and with it the finite termination property of CG. We consider primarily the iterative aspects of the algorithm. In fact, for-solving large sparse systems arising from the discretization of elliptic partial differential equations, the application of principal interest for us and for which the generalized CG method seems particularly effective, convergence to desired accuracy often occurs within a number of iterations small compared with n.

2. OPTIMALITY PROPERTIES

From (1.6), we obtain

$$\underset{\sim}{z}^{(k+1)} = \underset{\sim}{z}^{(k-1)} - \omega_{k+1}(\alpha_k(I-M^{-1}N)\underset{\sim}{z}^{(k)} + \underset{\sim}{z}^{(k-1)} - \underset{\sim}{z}^{(k)}). \quad (2.1)$$

Define

$$K = I - M^{-1}N \ . \quad (2.2)$$

We have $\underset{\sim}{z}^{(1)} = (I - \alpha_0 K)\underset{\sim}{z}^{(0)}$, and there follows by induction that

$$\underset{\sim}{z}^{(\ell+1)} = [I - KP_\ell(K)]\underset{\sim}{z}^{(0)} \quad (2.3)$$

where

$$P_\ell(K) = \sum_{j=0}^{\ell} \beta_j^{(\ell)} K^j \ . \quad (2.4)$$

We denote

$$p_\ell(\lambda) = \sum_{j=0}^{\ell} \beta_j^{(\ell)} \lambda^j \quad (2.5)$$

and from (2.1) we have for $k = 2, 3, \ldots, \ell$

$$p_k(\lambda) = \omega_{k+1}(1-\alpha_k\lambda)p_{k-1}(\lambda) - (\omega_{k+1}-1)p_{k-2}(\lambda) + \alpha_k\omega_{k+1} \ ,$$

and

$$p_0(\lambda) = \alpha_0, \qquad p_1(\lambda) = \omega_2(\alpha_0 + \alpha_1 - \alpha_0\alpha_1\lambda) \ .$$

The coefficients $\{\beta_j^{(\ell)}\}_{j=0}^{\ell}$ can be generated directly. From (2.3) and the relation $\underset{\sim}{z}^{(\ell+1)} = \underset{\sim}{z}^{(0)} + K(\underset{\sim}{x}^{(\ell+1)} - \underset{\sim}{x}^{(0)})$, there follows

$$\underset{\sim}{x}^{(\ell+1)} = \underset{\sim}{x}^{(0)} + P_\ell(K)\underset{\sim}{z}^{(0)} .$$

Then if

$$Z = [\underset{\sim}{z}^{(0)}, K\underset{\sim}{z}^{(0)}, \ldots , K^\ell \underset{\sim}{z}^{(0)}] , \qquad (2.6)$$

$$\underset{\sim}{x}^{(\ell+1)} = \underset{\sim}{x}^{(0)} + Z\underset{\sim}{\beta}^{(\ell)} . \qquad (2.7)$$

Consider the weighted error function:

$$E(\underset{\sim}{x}^{(\ell+1)}) = \frac{1}{2} (\underset{\sim}{x} - \underset{\sim}{x}^{(\ell+1)})^T (M-N) (\underset{\sim}{x} - \underset{\sim}{x}^{(\ell+1)}) . \qquad (2.8)$$

Assuming that $(M-N)$ is nonsingular, we obtain, using

$$\underset{\sim}{z}^{(0)} = K(\underset{\sim}{x} - \underset{\sim}{x}^{(0)}) ,$$

the relations

$$E(\underset{\sim}{x}^{(\ell+1)}) = \frac{1}{2} \underset{\sim}{z}^{(0)^T} (I-KP_\ell(K))^T M(M-N)^{-1} M(I-KP_\ell(K))\underset{\sim}{z}^{(0)}$$

$$= \frac{1}{2} \underset{\sim}{e}^{(0)^T} (I-KP_\ell(K))^T (M-N)(I-KP_\ell(K))\underset{\sim}{e}^{(0)}, \qquad (2.9)$$

where

$$\underset{\sim}{e}^{(0)} = \underset{\sim}{x} - \underset{\sim}{x}^{(0)} .$$

Equivalently, we can use (2.7) and re-write (2.8) as

$$E(\underset{\sim}{x}^{(\ell+1)}) = \frac{1}{2} (K^{-1}\underset{\sim}{z}^{(0)} - Z\underset{\sim}{\beta}^{(\ell)})^T (M-N)(K^{-1}\underset{\sim}{z}^{(0)} - Z\underset{\sim}{\beta}^{(\ell)}) . \qquad (2.10)$$

The quantity $E(\underset{\sim}{x}^{(\ell+1)})$ is minimized when we choose $\underset{\sim}{\beta}^{(\ell)}$ so that

$$G\underset{\sim}{\beta}^{(\ell)} = \underset{\sim}{h} ,$$

where

$$G = Z^T(M-N)Z, \quad \underset{\sim}{h} = Z^T M\mathbf{z}^{(0)} .$$

Let

$$\kappa = \lambda_{\max}(K)/\lambda_{\min}(K) .$$

Then using arguments similar to those given in [12], the following can be shown:

(A) $$\frac{E(\underset{\approx}{x}^{(\ell+1)})}{E(\underset{\approx}{x}^{(0)})} \leq 4\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2(\ell+1)} . \qquad (2.11)$$

(B) The generalized CG method is optimal in the class of all algorithms for which

$$\underset{\approx}{x}^{(\ell+1)} = x^{(0)} + P_\ell(K)\underset{\approx}{z}^{(0)}.$$

That is, the approximation $\underset{\approx}{x}^{(\ell+1)}$ generated by the generalized CG method satisfies

$$E(\underset{\approx}{x}^{(\ell+1)}) = \min_{P_\ell} \frac{1}{2}\, \underset{\approx}{e}^{(0)^T}(I-KP_\ell(K))^T(M-N)(I-KP_\ell(K))\underset{\approx}{e}^{(0)} ,$$

where the minimum is taken with respect to all polynomials P_ℓ of degree ℓ.

Recall that we have assumed that M and (M-N) are positive definite and symmetric. Thus the eigenvalues of $K = (I-M^{-1}N)$ are all real and K is similar to a diagonal matrix. Hence, if K has only $p < n$ distinct eigenvalues, there exists a matrix polynomial $Q_p(K)$ so that

$$Q_p(K) = 0 .$$

In this case, $E(\underset{\approx}{x}^{(p)}) = 0$ and hence

$$\underset{\approx}{x}^{(p)} = \underset{\approx}{x} ,$$

so that the iteration converges in only p steps. The same result also holds if K has a larger number of distinct eigenvalues but $\underset{\sim}{e}^{(0)}$ lies in a subspace generated by the eigenvectors associated with only p of these eigenvalues.

We remark also that Statement (B) implies CG is optimal for the particular eigenvector mix of the initial error $\underset{\sim}{e}^{(0)}$, taking into account interior as well as extremal eigenvalues. As will be discussed in the next section, the extremal eigenvalues are approximated especially well as CG proceeds, the iteration then behavingasif the corresponding vectors are not present. Thus the error estimate (2.11), which is based on the extremal eigenvalues,tends to be pessimistic asymptotically. One often observes, in practice (see § 5), a superlinear rate of convergence for the CG method.

3. EIGENVALUE COMPUTATIONS

The CG method can be used in a very effective manner for computing the extreme eigenvalues of the matrix $K = I - M^{-1}N$. We write (see (2.1))

$$\underset{\sim}{z}^{(k+1)} = \underset{\sim}{z}^{(k-1)} - \omega_{k+1}(\alpha_k K\underset{\sim}{z}^{(k)} + \underset{\sim}{z}^{(k-1)} - \underset{\sim}{z}^{(k)}), \qquad (3.1)$$

as

$$K\underset{\sim}{z}^{(k)} = c_{k-1}\underset{\sim}{z}^{(k-1)} + a_k\underset{\sim}{z}^{(k)} + b_{k+1}\underset{\sim}{z}^{(k+1)},$$

or

$$K[\underset{\sim}{z}^{(0)},\underset{\sim}{z}^{(1)},\ldots,\underset{\sim}{z}^{(n-1)}]$$

$$= [\underset{\sim}{z}^{(0)},\underset{\sim}{z}^{(1)},\ldots,\underset{\sim}{z}^{(n-1)}] \begin{vmatrix} a_0 & c_0 & & & \\ b_1 & a_1 & c_1 & & \text{\huge 0} \\ & \ddots & \ddots & \ddots & \\ & & & & c_{n-2} \\ \text{\huge 0} & & & b_{n-1} & a_{n-1} \end{vmatrix}$$

thus defining a_k, b_k, and c_k. In matrix notation, the above equation can be **written** as

$$KZ = ZJ \ . \tag{3.2}$$

Assuming that the columns of Z are linearly independent, there follows from (3.2) that

$$K = ZJZ^{-1} \ ,$$

hence the eigenvalues of K are equal to those of J. As pointed out in §2, if K has repeated eigenvalues or if the vector $z^{(0)}$ is deficient in the direction of some eigenvectors of K, iteration (3.1) will terminate in $k < n$ steps.

The process described by (3.1) **is essentially** the Lanczos algorithm [13]. It has been shown by Kaniel [14] and by Paige [15] that good estimates of the extreme eigenvalues of K often can be obtained from the truncated matrix

$$J_k = \left| \begin{array}{cccccc} a_0 & c_0 & & & & \\ b_1 & a_1 & c_1 & & \mathbf{0} & \\ & \cdot & \cdot & \cdot & & \\ & & & & & \\ & \mathbf{0} & & \cdot & \cdot & c_{k-2} \\ & & & & b_{k-1} & a_{k-1} \end{array} \right|$$

where k is considerably less than n. This result holds even in the presence of round-off error [16].

It was pointed out in §1 that the equation describing the CG method is of the same form as that describing the Chebyshev semi-iterative method and Richardson second order method, but that a knowledge of the extreme eigenvalues of K is required for obtaining parameters for the latter two methods. Thus one could construct a polyalgorithm in which the CG method is used initially to obtain good approximations to the solution and to the extreme eigenvalues of K, after which the Chebyshev semi-iterative method (say) is used, thereby avoiding the additional work of repeatedly calculating CG parameters. This technique has been used in an effective manner by O'Leary [17].

4. CHOICE OF M

For the splitting M = I, N = I-A one obtains the basic, unmodified CG algorithm, for which

$$\underset{\sim}{z}^{(k)} = \underset{\sim}{r}^{(k)} = \underset{\sim}{b} - A\underset{\sim}{x}^{(k)}$$

is simply the residual at the k^{th} step. Since the rate of convergence of the generalized CG method, as given by the estimate (2.11), decreases with increasing

$$\kappa = \lambda_{max}(K)/\lambda_{min}(K),$$

it is desirable to choose a splitting for which κ is as small as possible. If A= L + D + U, where D consists of the diagonal elements of A and L(U) is a strictly lower (upper) triangular matrix, then it is reasonable to consider the choice

$$M = D, \quad N = - \quad (L + U) .$$

This M, which is equivalent to a rescaling of the problem, is one for which (1.3) can be solved very simply for $\underset{\sim}{z}$. It has-been shown by Forsythe and Straus [19] that if A is two-cyclic then among all diagonal matrices this choice of M will minimize K.

In many cases, the matrix A can be written in the form

$$A = \left(\begin{array}{c|c} M_1 & F \\ \hline F^T & M_2 \end{array}\right) , \tag{4.3}$$

where the systems

$$M_1\underset{\sim}{z}_1 = \underset{\sim}{d}_1 \qquad \text{and} \qquad M_2\underset{\sim}{z}_2 = \underset{\sim}{d}_2$$

are easy to solve, and for such matrices, it is convenient to choose

$$M = \left(\begin{array}{c|c} M_1 & 0 \\ \hline 0 & M_2 \end{array}\right) \text{and} \qquad N = \left(\begin{array}{c|c} 0 & -F \\ \hline -F^T & 0 \end{array}\right) .$$

Using (4.3), we can write the system (1.1) in the form

$$M_1\underset{\sim}{x}_1 + F\underset{\sim}{x}_2 = \underset{\sim}{b}_1 \tag{4.4a}$$

$$F^T\underset{\sim}{x}_1 + M_2\underset{\sim}{x}_2 = \underset{\sim}{b}_2 . \tag{4.4b}$$

Let the initial approximation for $\underset{\sim}{x}_1$ be $\underset{\sim}{x}_1^{(0)}$, and obtain $\underset{\sim}{x}_2^{(0)}$ as the solution to (4.4b) so that

$$M_2\underset{\sim}{x}_2^{(0)} = \underset{\sim}{b}_2 - F^T\underset{\sim}{x}_1^{(0)} .$$

This implies that

$$\underset{\sim}{z}_2^{(0)} = \underset{\sim}{0} ,$$

and hence by (1.10)

$$\alpha_0 = 1 ,$$

and thus

$$\underset{\sim}{x}^{(1)} = \begin{pmatrix} \underset{\sim}{x}^{(0)} + \underset{\sim}{z}_1^{(0)} \\ \underset{\sim}{x}_2^{(0)} \end{pmatrix} .$$

A short calculation shows that $\underset{\sim}{z}_1^{(1)} = \underset{\sim}{0}$ and hence $\alpha_1 = 1$. Using (1.6), a simple inductive argument then yields that for $j = 0, 1, 2, \ldots$

$$\alpha_j \equiv 1, \quad \underset{\sim}{z}_1^{(2j+1)} = \underset{\sim}{0}, \quad \underset{\sim}{z}_2^{(2j)} = \underset{\sim}{0} . \qquad (4.5)$$

This result was first observed by Reid [8] for the case in which M_1 and M_2 are diagonal, i.e., in which the matrix A has "Property A" and is suitably ordered. Other cases for elliptic boundary value problems in which matrices of the form (4.3) arise will be discussed in § 5. For these cases convergence can be rapid because K has only a few distinct eigenvalues, even though κ is not especially small.

Various other splittings of the matrix A can occur quite naturally in the solution of elliptic partial differential equations. For example, if one wishes to solve

$$\begin{aligned} -\Delta u + \sigma(x,y)u &= f \qquad (x,y) \in R \\ u &= g \qquad (x,y) \in \partial R , \end{aligned}$$

where R is a rectangular region, it is convenient to choose M as the finite difference approximation to a separable operator, such as the Helmholtz operator $-\Delta + C$, for which fast direct methods can be used [23]. A numerical example for this case is discussed in § 5. If one wishes to solve a separable equation, but on a nonrectangular region S, then by extending the problem to one on a rectangle R in which S is embedded, M can be chosen as the discrete approximation to the separable operator on R, for which fast direct methods can be used. Such a technique provides an alternative to the related capacitance matrix method [25] for handling such problems. Forms of this method utilizing CG, but in a different manner than here, are described in [26] and [27].

Several authors [4], [20], [21] have used CG in combination with symmetric successive overrelaxation (SSOR). For this method the solution of the equation $Mzz^{(k)} = \underset{\sim}{c} - (M-N)\underset{\sim}{x}^{(k)}$ reduces to the solution of

$$(D+\omega L)\, D^{-1}(D+\omega U)z^{(k)} = \omega(2-\omega)r^{(k)}$$

where D, L, and U are as described previously in this section (although D may be block diagonal), $\underset{\sim}{r}^{(k)} = \underset{\sim}{b} - \underset{\sim}{A}\underset{\sim}{x}^{(k)}$, and ω is a parameter in the open interval $(0,2)$. SSOR is particularly effective in combination with CG because of the distribution of the eigenvalues of K (cf. [22]).

Meijerink and van der Vorst [7] have proposed that the following factorization of A be used:

$$A = FF^T + E,$$

so that

$$M = FF^T, \qquad N = -E .$$

The matrix F is chosen with a sparsity pattern resembling that of A. This splitting appears to yield a matrix K with eigenvalues that also are favorably distributed for CG. A block form of this technique recently developed by Underwood [24] achieves a more accurate approximate factorization of A with less computer storage and about the same number of arithmetic operations per iteration.

Generally, in addition to the requirement that (1.5) be -"easy" to solve, M should have the following features if the generalized CG algorithm is to be computationally efficient. For rapid convergence one seeks a splitting so that

(i) $M^{-1}N$ has small or nearly equal eigenvalues

or (ii) $M^{-1}N$ has small rank.

Often a choice for M satisfying these restrictions comes about naturally from the inherent features of a given problem.

5. NUMERICAL EXAMPLES

For the first example, we consider the test problem discussed in [23]

$$-\mathrm{div}(a(x,y)\nabla u) = f \qquad (x,y) \in R$$

$$u - g \qquad (x,y) \in \partial R ,$$

where $a(x,y) = [1 + \frac{1}{2}(x^4 + y^4)]^2$ and R is the unit square $0 < x,y < 1$. After a transformation the problembecomes

$$\begin{aligned} -\Delta w + \sigma(x,y)w &= a^{-1/2}f \qquad (x,y) \in R \\ w &= a^{1/2}g \qquad (x,y) \in \partial R, \end{aligned} \qquad (5.1)$$

where $\sigma(x,y) = 6(x^2 + y^2)/a^{1/2}$. As in [23] we discretize (5.1) -on a uniform mesh of width h, using for Δ the standard five-point approximation Δ_h, and we choose the splitting

$$M = A + N = -\Delta_h + CI$$

with $C = 0 = \sigma_{\min}$ or $C = 3 = \frac{1}{2}(\sigma_{\max} + \sigma_{\min})$.

In [23] Chebyshev acceleration was used, which requires an estimate of the ratio of the extremal eigenvalues of the iteration matrix. Here we use the modified CG algorithm of

§ 1. For an initial guess $W^{(0)} \equiv \underset{\sim}{0}$ and choice or' f and g corresponding to the solution $w = 2[(x-1/2)^2 + (y-1/2)^2]$, the results are given in Table 1 for $h = 1/64$. The results obtained for $h = 1/32$ were essentially identical, as the iteration is basically independent of h for this problem (see [23]).

Note that the Chebyshev method is sensitive both to the value of C and to the accuracy of the eigenvalues from which the parameters are calculated. The parameters used for the middle column were based on Gerschgorin estimates from the Rayleigh quotient, which gave a ratio of largest to smallest eigenvalue about three times too large. The CG method appears to be less sensitive to the value of C. After several iterations CG begins to converge more rapidly than does the optimal Chebyshev method, which behavior is typical of the CG superlinear convergence property discussed in § 2. This example is one for which rapid convergence results because the eigenvalues of $M^{-1}N$ are small.

	Chebyshev (from [23])			CG	
	C = 0	C = 3	C = 3		
.teration	exact eigenvalues	approximate eigenvalues	exact eigenvalues	C = 0	C = 3
1			1.6(-2)	4.5(-2)	1.6(-2)
2			7.4(-4)	2.6(-3)	6.7(-4)
3			1.1(-5)	3.0(-5)	1.0(-5)
4			2.7(-7)	5.7(-7)	1.1(-7)
5	2.4(-6)	1.1(-6)	4.3(-9)	5.1(-9)	8.2(-10
6			1.2(-10)	4.4(-11)	5.7(-12

TABLE 1

Maximum error vs. iteration number for first example

We give as the second example

$$-\Delta u = f \qquad (x,y) \in T$$

$$u = g \qquad (x,y) \in \partial T$$

where T is the domain shown in Fig. 1. For a uniform square mesh of width h, and $0 < \ell < (2h)^{-1}$ a whole number, so that all boundary segments are mesh lines, the coefficient matrix A for the standard five-point discretization and natural ordering has the form (4.3).

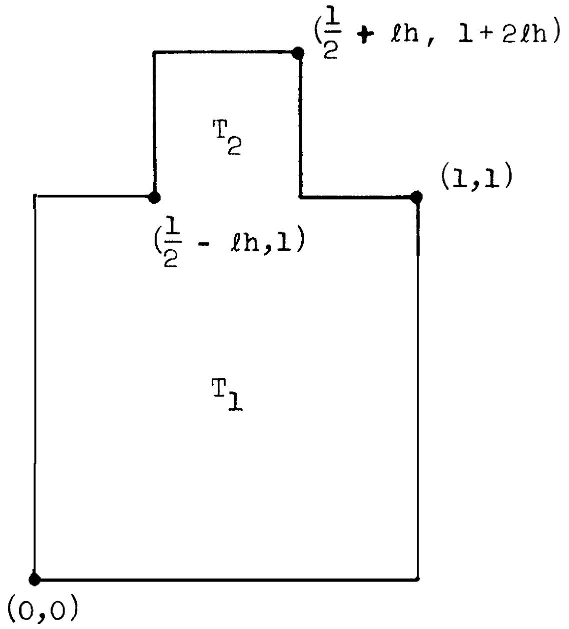
$(\frac{1}{2} + \ell h,\ 1+2\ell h)$
T_2
$(1,1)$
$(\frac{1}{2} - \ell h,1)$
T_1
$(0,0)$

FIGURE 1

T-shaped domain

M_1 and M_2 correspond to the mesh points in each of the two squares, T_1 and T_2, and F to the coupling between them. F has non-zero entries in only $p = 21 - 1$ of its rows.

According to the discussion following (4.3) we choose

$$M = \left(\begin{array}{c|c} M_1 & 0 \\ \hline 0 & M_2 \end{array}\right)$$

and for initial approximation

$$\underset{\sim}{U}^{(0)} = \left(\begin{array}{c} \underset{\sim}{U}_1^{(0)} \\ \hline M_2^{-1}(\underset{\sim}{b}_2 - F^T\underset{\sim}{U}_1^{(0)}) \end{array}\right) .$$

Then for the generalized CG algorithm, there holds $\alpha_k \equiv 1$ and that $\underset{\sim}{z}_1$ and $\underset{\sim}{z}_2$ are alternately zero, thereby reducing computational and storage requirements. We use a fast direct Poisson solver for the systems involving M_1 and M_2.

The results for $\underset{\sim}{U}_1^{(0)}$ uniformly distributed random numbers in $(0,2)$ and $f(x,y)$ and $g(x,y)$ such that $u = x^2 + y^2$ is the solution are given in Table 2. Here the average error per point, the two norm of the error divided by the square root of the number of interior mesh points, is given for each of the test problems.

For this example, the eigenvalues of $M^{-1}N$ are not especially small in magnitude, however since $M^{-1}N$ has rank of only $2p$, convergence is obtained in only a moderate number of iterations. For Case I and Case II the last row represents full convergence to machine accuracy subject to rounding errors, as would be expected since $2p = 14$ for these cases.

	Case I'	Case II	Case III
h	1/32	1/64	1/64
ℓ	4	4	8
p	7	7	15
iteration	ave. error/pt	ave. error/pt	ave. error/pt
1	8.58(-2)	3.70(-2)	1.08(-1)
2	7.05(-2)	3.13(-2)	9.82(-2)
3	1.30(-2)	6.66(-3)	4.94(-2)
4	3.35(-3)	2.53(-3)	1.80(-2)
5	2.71(-4)	6.03(-4)	4.28(-3)
10	2.65(-7)	5.13(-8)	7.35(-5)
15	1.14(-13)	5.60(-13)	4.71(-8)

TABLE 2

Average error per point vs. iteration number

We wish to thank Myron Stein of the Los Alamos Scientific Laboratory for his careful computer programming of the second test problem. This work was supported in part by the Energy Research and Development Administration, by the Hertz Foundation, and by the National Science Foundation.

REFERENCES

[0] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," J. Research NBS 49 (1952), 409-436.

[1] D. Young, Iterative Solution of Large Linear Systems, Academic Press, New York, 1972.

[2] D. J. Rose and R. A. Willoughby (ed.), Sparse Matrices and Their Applications, Plenum Press; New York-London, 1972.

[3] J. K. Reid, "On the method of conjugate gradients for the solution of large sparse systems of linear equations," Proc. Conference on 'Large sparse sets of linear equations," Academic Press, New York, 1971.

[4] O. Axelsson, "On preconditioning and convergence acceleration in sparse matrix problems," Report CERN 74-10 of the CERN European Organization for Nuclear Research, Data Handling Division, Laboratory I, 8 May 1974.

[5] R. Bartels and J.W. Daniel, "A conjugate gradient approach to nonlinear elliptic boundary value problems in irregular regions," Proc. Conf. on "Numerical solution of differential equations," Springer-Verlag, Berlin, 1974.

[6] R. Chandra, S.C. Eisenstat, and M.H. Schultz, 'Conjugate gradient methods for partial differential equations," in Advances in Computer Methods for Partial Differential Equations, R.Vichnevtsky (ed), Publ. A.I. C.A. -1975.

[7] J. A. Meijerink and H.A. van der Vorst, "Iterative solution of linear systems arising from discrete approximations to partial differential equations," Academisch Computer Centrum, Utrecht, The Netherlands, 1974.

[8] J. Reid, "The use of conjugate gradients for systems of linear equations possessing *Property A'," SIAM J. Numer. Anal. 9(1972), 325-332.

[9] E.L. Wachspress, "Extended applications of alternating direction implicit iteration model problem theory," SIAM J. 11 (1963), 994-1016.

[10] R. Fletcher and C.M. Reeves, "Function minimization by conjugate gradients," Comput. J., 7 (1964), 145-154.

[11] G. H. Golub and R.S. Varga, "Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods," Numer. Math. 3 (1961), 147-68.

[12] D.K. Faddeev and V.N. Faddeeva, Computational Methods of Linear Algebra, W.H. Freeman and Co., San Francisco, and London, 1963.

[13] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," J. Research NBS 45 (1950), 255-282.

[14] S. Kaniel, "Estimates for some computational techniques in linear algebra," Math. Comp. 20 (1966), 369-378.

[15] C.C. Paige, "The computation of eigenvalues and eigenvectors of very large sparse matrices," Ph.D. Thesis, London Univ., Institute of Computer Science, 1971.

[16] C.C. Paige, "Computational variants of the Lanczos method for the eigenproblem," J. Inst. Math. Appl. 10 (1972), 373-381.

[17] D. O'Leary, "Hybrid conjugate gradient algorithms," Ph.D. Thesis, Computer Science Dept., Stanford Univ., 1975.

[18] M. Engeli, T. Ginsburg, H. Rutishauser and E. Stiefel, Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems, Birkhäuser Verlag, Basel/Stuttgart, 1959.

[19] G. Forsythe and E.G. Straus, "On best conditioned matrices," Proc. *Amer.* Math. Soc. 6 (1955), 340-345.

[20] D.M. Young, L. Hayes, and E. Schleicher,"The use of the accelerated SSOR method to solve large linear systems," Abstract, 1975 SIAM Fall Meeting, San Francisco.

[21] L.W. Ehrlich, "On some experience using matrix splitting and conjugate gradient," Abstract, 1975 SIAM Fall Meeting, San Francisco.

[22] L.W. Ehrlich, "The block symmetric successive overrelaxation method," J. SIAM I.2 (1964), 807-826.

[23] P. Concus and G.H. Golub,"Use of fast direct methods for the efficient numerical solution of nonseparable elliptic equations,"SIAM J. Numer. Anal. 10 (1973), 1103-1120.

[24] R.R. Underwood,"An approximate factorization procedure based on the block Cholesky factorization and its use with the conjugate gradient method/Tech. Rep-t., General Electric Nuclear Energy Division, San Jose, CA (to appear).

[25] B.L. Buzbee, F.W. Dorr, J.A. George, and G.H. Golub,"The direct solution of the discrete Poisson equation in irregular regions:' SIAM J. Num. Anal. 8 (1971), 722-736.

[26] J.A. George/The use of direct methods for the solution of the discrete Poisson equation on non-rectangular regions,"Report CS-70-159, Computer Science Dept., Stanford, Univ., Stanford, CA. (1970).

[27] W. Proskurowski and O. Widlund,"On the numerical solution of Laplace's and Helmholtz's equations by the capacitance matrix method,"Tech. Rept., Courant Institute, NYU (to appear!.