Conjugate Gradient Type Methods for Unsymmetric and Inconsistent Systems of Linear Equations

Owe Axelsson
Department of Computer Sciences
Chalmers University of Technology
Fack, S-402 20 Goteborg 5, Sweden

Dedicated to Alston S. Householder
on the occasion of his seventy-fifth birthday.

Submitted by G. W. Stewart

ABSTRACT

Conjugate gradient type methods are discussed for unsymmetric and inconsistent system of equations. For unsymmetric problems, besides conjugate gradient methods based on the normal equations, we also present a (modified) minimal residual (least square) method, which converges for systems with matrices that have a positive definite symmetric part. For inconsistent problems, for completeness we discuss briefly various (well-known) versions of the conjugate gradient method. Preconditioning and rate of convergence are also discussed.

1. INTRODUCTION

Conjugate gradient methods have up to now mostly been applied to the solution of symmetric, positive semidefinite, consistent matrix problems of the form $Ax=b$, $b\in\mathfrak{R}(A)$ (see for instance [1], [2], [3], [4]). Although theoretically the method then converges in at most n steps to a solution, where n is the order of the matrix, it is now widely accepted that the method should be considered as an iterative method. This is so because the terminating property is not valid in the presense of roundoff errors and because for many important classes of problems the method converges to an acceptable accuracy in much fewer than n steps. The latter is true especially for preconditioned versions of the conjugate gradient method (see for instance [5], [6]).

0024-3795/80/010001+16$01.75

The number of iterations required to reach a relative accuracy of ε is at most

$$k=\operatorname{int}\left[\frac{\ln\left(\frac{1}{\varepsilon}+\sqrt{\frac{1}{\varepsilon^2}-1}\right)}{\ln\frac{1}{\sigma}}+1\right]$$

where

$$\sigma=\frac{1-\sqrt{\kappa^{-1}}}{1+\sqrt{\kappa^{-1}}},$$

κ being the effective spectral condition number of A; that is,

$$1\leqslant\kappa\leqslant\lambda_0/\lambda_1,$$

where λ_0,λ_1 are the extreme positive eigenvalues of A (see [7], [8]). For large values of κ and $1/\varepsilon$, the following upper bound is good enough for practical purposes:

$$k=\operatorname{int}\left[\frac{1}{2}\sqrt{\kappa}\ln\frac{2}{\varepsilon}+1\right]. \tag{1.1}$$

For preconditioned versions of the method, $\kappa(A)$ is replaced by the corresponding number of the preconditioned matrix $C^{-1}A$, that is, the spectral condition number of $C^{-1/2}AC^{-1/2}$, assuming that C is also symmetric and is positive definite.

For unsymmetric and for inconsistent problems, the method can be applied to the "normal" equations

$$A^TAx=A^Tb, \tag{1.2}$$

since here A^TA is positive semidefinite and $A^Tb\in\Re(A^TA)$. Furthermore, conjugate gradient algorithms only use the matrix in matrix-vector multiplications, so one obviously does not have to form the matrix A^TA (which could lead to cancellation and loss of sparsity; for further comments on this, see for instance [9]).

However, even so, this approach is not to be recommended in general, since as is well known, there is usually a serious amplification of the spectral condition number (it is squared). Hence, the number of iterations necessary to solve (1.2) for an almost symmetric matrix problem is typically $\sim\kappa(A)$, instead of $\sim\sqrt{\kappa(A)}$.

This type of situation is annoying, since if A is "almost" symmetric, i.e. a symmetric positive semidefinite matrix perturbed by a small skewsymmetric matrix, one would expect about the same number of iterations as for the symmetric case. Hence we look for a method, which in this situation only needs about the same number of iterations as for the symmetric part.

We also consider inconsistent systems, and show that a minimal residual conjugate gradient method will converge for $AA^T\tilde{u}=b$, $u=A^T\tilde{u}$, but that this is not so in general for the classical conjugate gradient method.

The methods are derived as special cases of a generalized conjugate gradient method, valid for the matrix problem $Bu=b$, where B has a positive definite symmetric part.

In the full version of the method, where, in the unsymmetric case, all previous search directions are used in order to calculate a new approximation, the rate of convergence is determined by the Krylov sequence $Br^0, B^2r^0, \ldots$ and not by $(B^TB)B^Tr^0, (B^TB)^2B^Tr^0, \ldots$, as would have been the case if the normal equations had been used. Here $r^0=Bu^0-b$ is the initial residual.

2. A GENERALIZED CONJUGATE GRADIENT METHOD

Consider minimizing the quadratic functional

$$f(u)=\tfrac{1}{2}(r,r)=\tfrac{1}{2}(Au-b,Au-b) \tag{2.1}$$

where $r=Au-b$ is the *residual*, $u\in\mathbb{R}^n$, $b\in\mathbb{R}^m$ and A is a real $m\times n$ matrix. Here $(\cdot,\cdot)$ is an inner product in $\mathbb{R}^m$ with corresponding norm $\|\cdot\|=(\cdot,\cdot)^{1/2}$. A vector $\hat{u}$ such that

$$f(\hat{u})=\inf_{u\in\mathbb{R}^n} f(u)$$

is called a *minimizer* of f. The existence of such a minimizer is trivial, since $\mathbb{R}^n$ is a finite dimensional vector space. If $b\in\mathfrak{R}(A)$, the range of A [in particular, if $\operatorname{rank}(A)=m$], then $f(\hat{u})=0$.

We shall now derive a general conjugate gradient method for matrices with positive (semi)definite symmetric part.

We assume in this section that $m=n$ and that the given matrix has a positive definite symmetric part. If A itself is symmetric, it suffices to assume that A is positive semidefinite. We also denote the matrix by B, to distinguish it from the matrix in the general case. The generalized conjugate gradient algorithm has the following form:

Given a vector (approximation of $\hat{u}$) u^k a set of $s=s_k$ $(s_k \leqslant s_{k-1}+1)$ vectors (*search directions*) d^{k-j}, $0 \leqslant j \leqslant s-1$, such that the set $\{Bd^{k-j}\}_{j=0}^{s-1}$ is linearly independent, we determine recursively new vectors u^{k+1} and d^{k+1} in the following way.

Along the search directions we determine s parameters $\lambda_{k-j}^{(k)} \in (-\infty, \infty)$, $0 \leqslant j \leqslant s-1$, such that $f(u^{k+1})$ is minimized, where

$$u^{k+1} = u^k + \sum_{j=0}^{s-1} \lambda_{k-j}^{(k)} d^{k-j}. \tag{2.2}$$

Hence

$$(Bu^{k+1} - b, Bd^{k-l}) = 0, \qquad 0 \leqslant l \leqslant s-1,$$

or

$$\sum_{j=0}^{s-1} \lambda_{k-j}^{(k)} (Bd^{k-j}, Bd^{k-l}) = -(r^k, Bd^{k-l}), \qquad 0 \leqslant l \leqslant s-1, \tag{2.3}$$

where

$$r^k = Bu^k - b. \tag{2.4}$$

We observe that the $s \times s$ matrix $\Lambda^{(k)} = [(Bd^{k-j}, Bd^{k-l})]$, $0 \leqslant l, j \leqslant s-1$, is positive definite, since the set of vectors $\{Bd^{k-j}\}_{j=0}^{s-1}$ are linearly independent; thus there exists a unique solution $\lambda_{k-j}^{(k)}$, $0 \leqslant j \leqslant s-1$, of (2.3).

We will sometimes find it convenient to use the recursion formula

$$r^{k+1} = r^k + \sum_{j=0}^{s-1} \lambda_{k-j}^{(k)} Bd^{k-j}, \tag{2.5}$$

which follows from (2.2), instead of using (2.4). (In this way we may save matrix-vector multiplications.) By construction,

$$(r^{k+1}, Bd^{k-l}) = 0, \qquad 0 \leqslant l \leqslant s-1. \tag{2.6}$$

We let the new search direction vector be a vector in the plane defined by the residual r^{k+1}, corresponding to the last approximation, and the most recent search direction, i.e.,

$$d^{k+1} = -r^{k+1} + \beta_k d^k, \qquad k = 0, 1, \ldots. \tag{2.7}$$

As we shall see later, the only possibility of a breakdown of the algorithm is when r^{k+1} and d^k are collinear. Furthermore, this can not happen for symmetric, positive semidefinite matrices, unless u^{k+1} is already a solution.

The new vector d^{k+1} is added to the set of search directions, possibly (if $s_k = s_{k-1}$) at the expense of the oldest direction d^{k-s+1}. We also find it convenient to define β_k so that

$$(Bd^{k+1}, Bd^k) = 0, \tag{2.8}$$

that is,

$$\beta_k = \frac{(Br^{k+1}, Bd^k)}{\|Bd^k\|^2}. \tag{2.9}$$

Then $\{Bd^{k+1}, Bd^k\}$ are linearly independent unless $d^{k+1} \in \mathcal{N}(B)$, the null-space of B.

Furthermore with $E = \frac{1}{2}(B + B^T)$, by (2.7) we get

$$\begin{aligned}(d^{k+1}, Bd^{k+1}) &= (d^{k+1}, Ed^{k+1}) = \left(-r^{k+1} + \beta_k d^k, E\left[-r^{k+1} + \beta_k d^k\right]\right) \\ &= (r^{k+1}, Er^{k+1}) + \beta_k^2 (d^k, Ed^k) - 2\beta_k (r^{k+1}, Ed^k).\end{aligned}$$

Since by assumption, if B is unsymmetric then E is positive definite, we have $(d^{k+1}, Bd^{k+1}) \geqslant 0$ and $(d^{k+1}, Bd^{k+1}) = 0$ iff $d^{k+1} = 0$, i.e. iff r^{k+1} and d^k are collinear.

If B is symmetric, by (2.6) we get

$$(d^{k+1}, Bd^{k+1}) = (r^{k+1}, Br^{k+1}) + \beta_k^2 (d^k, Bd^k) > (r^{k+1}, Br^{k+1}) \geqslant 0. \tag{2.10}$$

Hence, $d^{k+1} \in \mathcal{N}(B)$ iff $r^{k+1} \in \mathcal{N}(B) = \mathcal{N}(B^T)$. But then $B^T(Bu^{k+1} - b) = 0$, i.e., u^{k+1} is a solution of the normal equations (1.1), and $u^{k+1} = \hat{u}$ is a minimizer of (2.1) (with $A = B$).

Hence, if B is symmetric and positive semidefinite, $d^{k+1} \in \mathcal{N}(B)$ *only if u^{k+1} is already a solution.* If B is unsymmetric with positive definite symmetric part, we have $d^{k+1} \in \mathcal{N}(B)$ iff r^{k+1} and d^k are collinear. In this latter situation, or rather when the angle between r^{k+1} and d^k is smaller than a given (small) number $\delta(0<\delta<\pi/2)$, we make a *restart* of the algorithm, with the latest approximation as an initial approximation.

By (2.6) and (2.7) we have, for $s\geqslant 2$,

$$\begin{aligned}(r^{k+1},Br^l)&=(r^{k+1},-Bd^l+\beta_{l-1}Bd^{l-1})\\&=-(r^{k+1},Bd^l)+\beta_{l-1}(r^{k+1},Bd^{l-1})=0, \qquad k-s+2\leqslant l\leqslant k.\end{aligned}\tag{2.11}$$

Hence the algorithm is of *conjugate gradient* or rather conjugate residual type.

We observe that from (2.3) and (2.6) it follows that

$$\lambda_k^{(k)}=\frac{\det\Lambda_{s-1}^{(k-1)}}{\det\Lambda_s^{(k)}}(r^k,Br^k).$$

Hence, since $\Lambda_s^{(k)}$ is a positive definite matrix and since the symmetric part of B is positive (semi)definite, we have $\lambda_k^{(k)}\geqslant 0$, and $\lambda_k^{(k)}>0$ unless u^k is a solution.

Now, from (2.5), (2.6), and (2.7) we get

$$\begin{aligned}(r^{k+1},r^{k+1})&=\left(r^{k+1},r^k+\sum_{j=0}^{s-1}\lambda_{k-j}^{(k)}Bd^{k-j}\right)=(r^{k+1},r^k)\\&=\left(r^k+\sum_{j=0}^{s-1}\lambda_{k-j}^{(k)}Bd^{k-j},r^k\right)=(r^k,r^k)+\lambda_k^{(k)}(Bd^k,r^k)\\&=(r^k,r^k)-\lambda_k^{(k)}(Br^k,r^k)+\lambda_k^{(k)}\beta_{k-1}(Bd^{k-1},r^k);\end{aligned}$$

hence

$$(r^{k+1},r^{k+1})=(r^k,r^k)-\frac{\det\Lambda_{s-1}^{(k-1)}}{\det\Lambda_s^{(k)}}(r^k,Br^k)^2.\tag{2.12}$$

Hence we have *monotone convergence*

$$f(u^{k+1})<f(u^k),$$

unless u^k is already a solution, as long as the set $\{Bd^{k-j}\}_{j=0}^{s-1}$ is linearly independent. We have proven that this is so at least for $s \leqslant 2$.

So far we have not commented on the choice of u^0, d^0. We may let u^0 be arbitrary and let

$$d^0 = -r^0.$$

The advantage of this choice will become clear in the analysis to follow. Since $d^0 = -r^0$, by construction d^k and r^k are linear combinations of vectors in the so called *Krylov sequence*

$$r^0, Br^0, \dots, B^k r^0,$$

that is

$$d^k \in \operatorname{span}\{r^0, Br^0, \dots, B^k r^0\}$$

and

$$r^k \in \operatorname{span}\{r^0, Br^0, \dots, B^k r^0\}.$$

Hence, r^k can be written as a polynomial of degree k in the monomials $B^j r^0$, $j = 0, 1, \dots, k$, with constant coefficient $= 1$, i.e.,

$$r^k [1 + p_k(B)] r^0, \qquad k \geqslant 1,$$

where $p_k(0) = 0$. Hence $f(u^k) = \frac{1}{2}(r^k, r^k) = \frac{1}{2}\|r^0 + p_k(B) r^0\|^2$. Different choices of $\{\lambda_j^{(k-1)}\}$ will produce, in general, different polynomials p_k.

We may regard $-p_k(B)r^0$ as an approximation of r^0, and $\|r^0 + p_k(B)r^0\|$ as the corresponding error. If $s = s_k = k+1$, that is, if all previous search directions are used, then

$$(r^k, r^k) = \min_{\substack{p_k \in \pi_k \\ p_k(0)=0}} \|[1 + p_k(B)] r^0\|, \tag{2.13}$$

whatever the choice of β_l, as long as $\{Bd^{k-j}\}$, $0 \leqslant j \leqslant s-1$, is linearly independent.

It would in general, however, cost too many arithmetic operations and storage to keep all previous vectors. Hence only a few, or sometimes even only one, search direction is retained. In particular, this will be the case if B is a symmetric matrix, as we shall see below.

After at most n steps of the algorithm with $s=s_k=k+1$, we would have reached a solution, had only all numbers been calculated without rounding errors. Consequently, the method may be viewed as a direct method. However, since in many important problems one may reach a satisfactory solution after much fewer steps, we shall use the method as an iterative method. Thus we need some criterion to determine when to stop. Such a criterion will be given for each particular algorithm to be presented.

We also realize that the rate of convergence of the algorithm depends on the rate with which linear combinations of the vectors $\{Br^0,B^2r^0,\ldots,B^kr^0\}$ will approximate r^0 *in the norm chosen* (i.e., in the norm defined by the inner product). Also, we notice that the matrix B is only needed in forming products with a vector. This may in many important applications be possible without actually having formed the entries of B.

For symmetric matrices it is easy to prove that $(Bd^k,Bd^l)=0$, $l\neq k$, and that by chosing inner products $(u,w)=v^TMw$, where M is symmetric and positive definite, the algorithm reduces to the standard conjugate gradient algorithm with $M=B^{-1}$ (assuming that B is positive definite). With $M=I$ we get the minimal residual algorithm.

3. INCONSISTENT SYSTEMS OF EQUATIONS

For completeness we briefly list various methods to deal with rectangular and inconsistent matrix problems, although these techniques are well known (see [1], [2], [3], [10], [11], [12], and for a survey [13]).

3.1. *Variable transformations*

Let us now consider a system with a general $m\times n$ matrix A. We make a transformation of the vector

$$u=Q^T\tilde{u}, \tag{3.1}$$

where Q is an $m\times n$ matrix of full rank and $\tilde{u}\in\mathbb{R}^m$. We solve for $\tilde{u}$ and then substitute this solution into (3.1). We get $r=\tilde{r}=AQ^T\tilde{u}-b$, and we may use a *minimal residual conjugate gradient algorithm* as described in Sec. 2 with $B=AQ^T$, where B is an $m\times m$ matrix. If the symmetric part of B is positive definite, we have convergence to a solution satisfying

$$||B\tilde{u}-b||=\inf_{v\in\mathbf{R}^m}||Bv-b||, \tag{3.2}$$

even if the system is inconsistent. A particularly interesting choice of Q is $Q=A$.

3.2. *The pseudoinverse solution algorithm*

Let

$$u = A^T\tilde{u}, \qquad \tilde{r} = AA^T\tilde{u} - b = B\tilde{u} - b. \tag{3.3}$$

Here $B = AA^T$ is symmetric and positive semidefinite. In this case it follows from Sec. 2 that the minimal residual conjugate gradient method for symmetric matrices will converge to a minimal residual solution, satisfying (3.2). It is quite easy to see that the method in fact converges to the pseudoinverse solution (see for instance [10]). On the other hand, the classical conjugate gradient method for solving $AA^Tu = b$ will not converge in general if b is not consistent [i.e., if $b \notin \mathcal{R}(A)$] (cf. [14]). A simple example showing this is provided by

$$B = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad b^T = (1,0,2), \qquad \tilde{u}^0 = (1,0,0),$$

in which B is positive semidefinite and $b \notin \mathcal{R}(B)$. Here we have $r^{1T}r^1 > r^{0T}r^0$ and $d^1 \in \mathcal{N}(B)$.

3.3. *A minimal error conjugate gradient algorithm*

The classical conjugate gradient method converges however for all consistent systems $Bu = b$, $b \in \mathcal{R}(B)$, even if B is only positive semidefinite. This follows because in this case we are actually minimizing

$$f(u) = \tfrac{1}{2}(u - \hat{u})^T B (u - \hat{u}),$$

where $\hat{u}$ is a solution of $Bu = b$. Even if the initial approximation u^0 has a component in $\mathcal{N}(B)$, $r^0 = Bu^0 - b$ (and d^0) does not have such a component, since $b \in \mathcal{R}(B)$. Hence the corresponding Krylov sequence consists of vectors in $\mathcal{R}(B)$ only, and the rate of convergence of the classical conjugate gradient method will then be determined by the positive part of the spectrum (cf. Sec. 1). All iterates u^k have the same component in $\mathcal{N}(B)$ as u^0.

In particular, if $b \in \mathcal{R}(A)$, applied to $B\tilde{u} = b$, $\tilde{u} = B^+b$, $B = AA^T$, we get a minimal norm solution, where the error $[(u - \hat{u})^T(u - \hat{u})]^{1/2}$, $\hat{u} = A^+b$, is minimized; see [11], [12], [15], and [9].

3.4. *Residual transformations*

In (3.1)–(3.3) we have transformed the unknown vector. Let us now make a transformation of the system (the residual) instead. Then we get

$$\tilde{r}=Q(Au-b),\qquad Q \quad \text{an} \quad n\times m \quad \text{matrix of full rank.}$$

We call $\tilde{r}$ a *pseudoresidual.*

Again (cf. Sec. 3.1) we may chose Q in order to get a spectrum of QA more favorable than that of A itself. Obviously we may choose $Q=A^T$, from which we get the normal equations

$$A^TAu=A^Tb. \tag{3.4}$$

These are always consistent, but usually with much worsened spectrum, so that the rate of convergence of a conjugate gradient method will be far too slow.

On the other hand, we may apply any of the algorithms for symmetric systems, either to $BB^T\tilde{u}=Qb$, as in Secs. 3.2 and 3.3, or to $B^TBu=B^TQb$ as above, where $B=QA$ and Q is a preconditioning matrix (see for instance [8]).

3.5. *A symmetric preconditioning*

If A is symmetric and positive semidefinite and if the system is consistent, we may, as was already noted in Sec. 3.3, apply the classical conjugate gradient method in order to minimize $f(u)=(u-\hat{u})^TA(u-\hat{u})$, where $A\hat{u}=b$.

If we transform both the residual and the vector, that is, $\tilde{r}=LAL^T\tilde{u}-\tilde{b},\tilde{b}=Lb, u=L^T\tilde{u}$, we preserve symmetry of the matrix. Here we assume that L is an invertible matrix. To this system we may apply the classical conjugate gradient method with $B=LAL^T$. Here L is usually chosen so that LAL^T has a favorable distribution of eigenvalues and is of low computational complexity; see for instance [8].

4. MODIFIED MINIMAL RESIDUAL CONJUGATE GRADIENT METHODS FOR UNSYMMETRIC SYSTEMS OF EQUATIONS

For unsymmetric matrices B, preconditioned or not, the use of the normal equations or of the transformation $BB^T\tilde{u}=b$, $u=B^T\tilde{u}$, in order to get a symmetric and positive semidefinite matrix, leads to the solution of a system with a matrix with usually large condition numbers (typically squared). Consequently, the slow convergence of any conjugate gradient method can be expected. This type of situation is particularly annoying when B is almost symmetric, because then we would expect a good method to give about the same number of iterations as for the symmetric case.

On the other hand, if we work with a conjugate gradient method directly on an unsymmetric matrix, all previous vectors have to be kept if we want to preserve the best approximation property (2.13). Hence it is of interest to examine methods where we still only keep one vector, as in the symmetric case, or possibly only a few more vectors.

Let us assume that B may be unsymmetric but has a *positive definite symmetric part* and that we carry along a fixed number of s vectors (of each u, r, d, Bd, etc.). In general, we cannot have orthogonality to all previous vectors if B is unsymmetric. However, from Sec. 2.1, Eqs. (2.6), (2.11), and (2.12), we do know that

$$(r^{k+1},Bd^{k-l})=0, \qquad 0\leqslant l\leqslant s-1,$$

$$(r^{k+1},Br^{l})=0, \qquad k-s+2\leqslant l\leqslant k,$$

$$(r^{k+1},r^{k+1})=(r^{k},r^{k})-\frac{\det\Lambda_{s-1}^{(k-1)}}{\det\Lambda_{s}^{(k)}}(r^{k},Br^{k})^{2},$$

so that we have monotone convergence towards a solution.

Let us consider two cases for the minimal residual method: $s=1$ (which turns out to be equivalent to the case $s=2$), and $s=3$.

$s=1$ $(s=2)$: From (2.3), (2.6), (2.7), and (2.9) we get

$$\begin{aligned}\lambda_{k}^{(k)}&=r^{kT}Br^{k}/\|Bd^{k}\|^{2},\\ \beta_{k}&=(Br^{k+1})^{T}Bd^{k}/\|Bd^{k}\|^{2}.\end{aligned}\tag{4.1}$$

One more inner product than in the case where B is symmetric is now needed. For $s=2$ we have $(r^{k},Bd^{k-1})=0$, and since the matrix $\Lambda_{2}^{(k)}$ is diagonal [because of (2.8)] and positive definite, we have $\lambda_{k-1}^{(k)}=0$. Hence the above formulas (4.1) are valid also in this case.

$s=3$: From (2.3), (2.9) it again follows that $\lambda_{k-1}^{(k)}=0$, because $(Bd^{l},Bd^{l-1})=0$, $l=k,k-1$. Hence we have

$$\begin{bmatrix}\|Bd^{k}\|^{2} & (Bd^{k},Bd^{k-2})\\ (Bd^{k-2},Bd^{k}) & \|Bd^{k-2}\|^{2}\end{bmatrix}\begin{bmatrix}\lambda_{k}^{(k)}\\ \lambda_{k-2}^{(k)}\end{bmatrix}=\begin{bmatrix}(r^{k},Br^{k})\\ 0\end{bmatrix}.$$

Now four inner products and the equivalent of six recursion formula operations have to be performed per iterative step. Still only one matrix-vector multiplication (Br^{k+1}) is needed.

In this algorithm, the term $\lambda_{k-2}^{(k)}Bd^{k-2}$ can serve as a check of the extent of asymmetry in B (or the extent of loss of orthogonality). If the norm of this term is small in comparison with $\lambda_k^{(k)}\|Bd^k\|$ for all k, then we have an indication that B is almost symmetric. Depending on the degree of unsymmetry, it will pay off in fewer iterations to use the algorithm with $s=3$, which demands $\sim 10n$ operations ("flops") instead of $\sim 7n$ for $s=1$. If the matrix-vector multiplication Br^{k+1} costs γn flops, then the former method will be better if the number of iterations is $<(7+\gamma)/(10+\gamma)$ of the number for $s=1$. The above algorithms for unsymmetric matrices we call *modified minimal residual conjugate gradient methods.*

In this case the algorithm breaks down if Bd^k becomes collinear to Bd^{k-2}. Should this happen, one can change to the algorithm for $s=1$ (2). This latter algorithm breaks down only if r^{k+1} and d^k are collinear. One can make a restart if the angle between these vectors becomes too small. Practical experience with such restarts will be reported elsewhere.

The generalization of the conjugate gradient method considered in this paper is based on the minimization of a functional, namely the least square residual functional (2.1). A similar method may be based on a Galerkin method which satisfies $(r^{k+1},v)=0$ for all v in the Krylov set (see [16]). This latter approach has already been considered in [17] and [18], in the special case where $(u,v)=u^TMv$, with M the symmetric part of MB, $B=1-M^{-1}N$, where N is skew symmetric. When M^{-1} is used as a preconditioning matrix for MB, the recursion formulas then simplify considerably (only one parameter is needed). However, this latter method in general works well only when the skew symmetric part is not too large (see [18]).

5. A NUMERICAL TEST EXAMPLE

In order to test the algorithms for unsymmetric matrices, the following model example was used. We discretized

$$-\Delta u+\beta u_x=f(x),\qquad x\in\Omega\subset R^2,$$

$$u=0\qquad \text{on } \partial\Omega, \tag{5.1}$$

where Ω was the unit square and $\beta>0$ a constant. The discretization chosen was a central approximation for Δ and a so-called upwind first order approximation for u_x (in this case a backward difference approximation) (cf. [19]). This results in an unsymmetric square matrix but with positive definite symmetric part. We observe however that care has to be taken in more general turning point problems of the type (5.1), where β changes sign, in order to get a positive definite matrix.

We solved the resulting linear system (which in this case is consistent) for different values of β and the discretization parameter h.

When the modified minimal residual algorithm with $s=3$ (see Sec. 4) was used, it was observed that for β not too large, say $\beta \leqslant 10$, the rate of convergence during the first (say) 10 iterations was not much slower than in the symmetric case, where $\beta = 0$. For β very large, say $\beta = 1000$, $h = \frac{1}{8}$, it was observed that during the first 11 iterations, convergence was slow (about 10 iterations for one decimal) but that during the following 3 or 4 iterations the rate on the other hand was extremely fast. This type of situation seemed to repeat itself during the following iterations, i.e., slow convergence for about 10 iterations and very fast for about 3 iterations. The size of the norm of the last correction term (index $k-2$) was about 10^{-2} to 10^{-1} times the norm of the first correction term (index k); cf. (2.3).

When a preconditioned system $\tilde{r} = C^{-1}(Au - b) = 0$ was solved with the preconditioning matrix C a modified incomplete factorization of A (see [20]) and with the same algorithm but with $s=1$ as above, a much more favourable rate of convergence was noted; see Table 1. The rate of convergence was now approximately constant during the iterations, i.e., $\log(\|r^k\|/\|r^0\|)$ as a function of k decreased linearly. For $s=3$ the number of iterations differed at most by 1.

TABLE 1
NUMBER OF ITERATIONS, k, FOR A RELATIVE ACCURACY
$\|\tilde{r}^k\|/\|\tilde{r}^0\| \leqslant 10^{-5}$[a]

β h^{-1}	0	1	10	100	1000
8	6	6	6	4	3
16	10	10	8	6	4
32	14	14	12	10	6

[a] Preconditioned modified minimal residual ($s=1$).

Two versions of preconditioned conjugate gradient methods were used, one of unsymmetric form and one based on the symmetric preconditioning (see Sec. 3.5). For an unsymmetric problem, the preconditioned matrix is however obviously still unsymmetric. A modified approximate factorization $A \approx LU$, L, U lower and upper triangular matrices, was used. One could expect some difference in the number of iterations for small values of β. However, we found at most a difference of one iteration, although the second correction term (for the case $s=3$) was smaller in the "symmetric" version, in particular of course for $\beta=0$ when the size of the second term is about the size of the machine accuracy (double precision was used).

Finally a comparison with the solution of the normal equations for the preconditioned system was made. We then have to solve

$$B^TBu=B^TC^{-1}b,$$

where $B=C^{-1}A$. The classical conjugate gradient method (Sec. 2.2) was used, and the iterations were stopped when the pseudoresidual $\tilde{r}^k=C^{-1}(Au^k-b)$ was small enough, $\|\tilde{r}^k\|\leqslant 10^{-5}\|\tilde{r}^0\|$. This was also used for stopping in Table 1. In Table 2 the corresponding numbers of iterations are given. It is noticed that for not too large values of β, the number of iterations grows as $O(h^{-1})$ with $h\to 0$, whereas in the modified minimal residual algorithm, the numbers of iterations were fewer and grew only as $O(h^{-1/2})$. In particular, we notice that the unsymmetric problem was solved as fast as the symmetric one. That the number of iterations even decreased with increasing large values of β is due to the fact that the approximate factorization becomes more and more accurate then, since the given matrix becomes more and more triangular.

TABLE 2

NUMBER OF ITERATIONS, k, FOR A RELATIVE ACCURACY $\|\tilde{r}^k\|/\|\tilde{r}^0\|\leqslant 10^{-5}$[a]

h^{-1} \ β	0	1	10	100	1000
8	10	11	11	7	4
16	19	21	20	12	6

[a] Classical conjugate gradient method on preconditioned normal equations.

As a conclusion we note that this test indicates that the modified minimal residual method [with $s=1$ (2)] is a reliable method when the matrix is preconditioned by a modified incomplete (first order) factorization method, and in such a way that the resulting matrix has a positive definite symmetric part.

In [21] tests are reported with a seemingly less appropriate preconditioning technique, because more iterations were needed and a value (corresponding to ours) of $s = 5$ was recommended. In that case it seems as if the danger that the algorithm will break down due to almost linearly dependent vectors $\{Bd^l\}$ is too large.

I acknowledge interesting comments by Professors David M. Young and Stanley Eisenstat regarding an early version of Sec. 2. My student Gunhild Lindskog made the numerical tests reported here on the computer.

REFERENCES

1 M. R. Hestenes and E. Stiefel, Method of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards*, No. 49, 409–436 (1952).

2 C. Lanczos, Solution of the systems of linear equations by minimized operations, *J. Res. Nat. Bur. Standards*, No. 49, 33–53 (1952).

3 M. R. Hestenes, The conjugate gradient method for solving linear systems, in *Proceedings of the Symposium on Applied Mathematics*, Vol. 6, New York, 1956, pp. 83–102.

4 J. K. Reid, On the method of conjugate gradients for the solution of large sparse systems of linear equations, in *Proceedings of the conference on Large Sparse Sets of Linear Equations* (J. K. Reid, Ed.), Academic, 1971, pp. 231–254.

5 O. Axelsson, A generalized SSOR method, *Nordisk Tidskr. Informationsbehandling (BIT)* 13:443–467 (1972).

6 O. Axelsson, On preconditioning and convergence acceleration in sparse matrix problems, CERN 74-10, Geneva, 1974.

7 O. Axelsson, A class of iterative methods for finite element equations, *Computer Methods in Applied Mechanics and Engineering* 9:123–137 (1976).

8 O. Axelsson, Solution of linear systems of equations: iterative methods, in *Sparse Matrix Techniques* (V. A. Barker, Ed.), Copenhagen, 1976; Lecture Notes in Mathematics #572, Springer, 1977.

9 Å. Björck and T. Elfving, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, Report LITH-MAT-R-1977, Linköping Univ., Sweden, 1977.

10 M. Z. Nashed, Generalized inverses, normal solvability and iteration for singular operator equations, in *Nonlinear Functional Analysis and Applications* (L. B. Rall, Ed.), Proceedings, Univ. of Wisconsin, Madison (1970); Academic, 1971.

11 V. M. Fridman, New methods for solving linear operator equations, *Dokl. Akad. Nauk. SSSR* 128 (3):482–484 (1959).

12 D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*, Freeman, San Francisco, 1963.

13 Å. Björck, Methods for sparse linear least squares problems, in *Sparse Matrix Computations* (J. Bunch and D. Rose, Eds.), Academic, 1976.

14 T. Elfving, On computing generalized solutions of sparse linear systems with application to some reconstruction problems, Linköping Studies in Science and Technology. Dissertations. No. 27, Linköpings Univ., Sweden, 1978.

15 E. J. Craig, The N-step iteration procedure, *J. Mathematical Phys.* 34:65–73 (1955).

16 O. Axelsson, A generalized conjugate direction method, in preparation.

17 P. Concus and G. H. Golub, A generalized conjugate gradient method for nonsymmetric systems of linear equations, in *Proceedings of the Second International Symposium on Computing Methods in Applied Sciences and Engineering,* IRIA, Paris, Dec. 1975; Lecture Notes in Economics and Mathematical Systems, Vol. 134 (R. Glowinski and J-L. Lions, Eds.), Springer, Berlin, 1976.

18 O. Widlund, A Lanczos method for a class of nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.* 15:801–812 (1978).

19 O. Axelsson and I. Gustafsson, A modified upwind scheme for convective transport equations and the use of a conjugate gradient method for the solution of non-symmetric systems of equations, Computer Sciences Department 77.12 R, Chalmers University of Technology, Gothenburg, Sweden, 1977; *J. Inst. Math. Appl.* 23:321–337 (1979).

20 I. Gustafsson, A class of first-order factorization methods, *Nordisk Tidskr. Informationsbehandling (BIT)* 18:142–156 (1978).

21 P. K. W. Vinsome, Orthomin, an iterative method for solving sparse sets of simultaneous linear equations, Society of Petroleum Engineers of AIME, paper number SPE 5729, 1976.

Received 30 November 1978