Journal of Computational and Applied Mathematics 24 (1988) 73–87
North-Holland

Conjugate gradient type methods and preconditioning

Henk A. VAN DER VORST and Kees DEKKER
Delft University of Technology, Faculty of Technical Mathematics and Informatics, P.O. Box 356, 2600 AJ Delft, The Netherlands

Received 25 March 1988

Abstract: In this paper we consider various iterative methods for the numerical solution of very large, sparse linear systems of equations, which arise in the discretization of partial differential equations. As the performance of the methods generally depends on the characteristics of the problems to be solved, a judicious choice between the methods will require knowledge about the system. The aim of this paper is to review the properties of the various iteration methods, in order to assist the user in making a deliberate selection.

1. Introduction

In many situations iterative methods for the solution of large sparse linear systems may be preferred over direct methods, for one or more reasons:

- usually less memory requirements,
- often a sufficiently accurate approximation to the solution is obtained with less computational effort,
- in general easier to program and to adapt to different types of problems.

The main problem, however, is to make the proper choice among the available iterative methods for the problem at hand. Since the convergence behaviour of the iterative methods depends on the spectrum of the matrix of the linear system (or some related matrix) and since the methods themselves often require certain properties for the matrix, like e.g. diagonal dominance, the problem reduces to the identification of classes of linear systems for which a successful iterative method is known to exist.

Even for the classes of linear systems that arise from finite difference approximation of certain partial differential equations the selection of a successful iterative method may be rather complicated, and, in many cases, comes down to trial and error. In order to assist the reader in finding his way through the jungle of iterative methods we have chosen to describe a number of robust methods and a number of possibilities to accelerate convergence. The methods are robust in the sense that for a certain class of matrices the user does not have to specify additional (often difficult to obtain) information, like values of eigenvalues or so. Since in many cases the problem of identification of a class of matrices, for which (fast) convergence is guaranteed, is not yet solved we report also on methods for which experiments give confidence in their potential success.

The contents of this paper are as follows. In Section 2 we give some notations and mathematical preliminaries. The iterative methods are described and briefly discussed in Section 3. A well-known technique to accelerate convergence is known as preconditioning. The construction of relatively simple preconditioning matrices is subject of Section 4. We have chosen to discuss only relatively simple preconditioning techniques since we have also applications on vector and parallel computers in mind. These aspects will be treated in a separate paper. Finally we comment on the occurrence of classes of matrices when discretizing partial differential equations by finite difference approximations in Section 5 and we present tables that help to make the choice in iterative methods and preconditionings for certain types of problems.

2. Mathematical preliminaries

Consider the system of N linear equations

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}, \tag{2.1}$$

where $\boldsymbol{b}$ is a known vector, $\boldsymbol{x}$ is the unknown solution, and $\boldsymbol{A}$ is a sparse $N \times N$ matrix. We assume that the elements of $\boldsymbol{A}$ are not stored explicitly, but that there exists a recipe to compute the (non-zero) entries of $\boldsymbol{A}$, and to form the product of $\boldsymbol{A}$ with an arbitrary vector of appropriate dimension. An iterative method generates, given an initial approximation $\boldsymbol{x}^{(0)}$ to $\boldsymbol{x}$, a sequence $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)},\ldots,$ which will hopefully converge to the solution $\boldsymbol{x}$. Throughout it is assumed that $\boldsymbol{x}^{(0)}$ be the zero-vector. This is not a restriction, as (2.1) could be rewritten like

$$\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^{(0)})=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(0)}, \tag{2.2}$$

and instead of (2.1) the solution of (2.2) might be considered.

A simple iterative technique consists of the *Richardson-iteration* (see e.g. Varga [33])

$$\boldsymbol{x}^{(n+1)}=(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{x}^{(n)}+\boldsymbol{b}, \tag{2.3}$$

from which it is easily seen that $\boldsymbol{x}^{(n+1)}$ is a linear combination of the vectors $\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b},\ldots, \boldsymbol{A}^n\boldsymbol{b}$.

Definition 2.1. The space spanned by the vectors $\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b},\ldots, \boldsymbol{A}^n\boldsymbol{b}$ is called the *Krylov subspace* $\boldsymbol{K}^{(n+1)}(\boldsymbol{A};\, \boldsymbol{b})$.

Most iterative methods have in common that they select as iterates vectors from the Krylov spaces $K^{(n)}(\boldsymbol{A};\, \boldsymbol{b})$ for $n=1, 2,\ldots,$ but they differ in the choice of the selection criteria. For example, one could choose the vector $\boldsymbol{x}^{(n+1)} \in \boldsymbol{K}^{(n+1)}(\boldsymbol{A};\, \boldsymbol{b})$ such that the residual

$$\boldsymbol{r}^{(n+1)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(n+1)} \tag{2.4}$$

is minimized in Euclidean norm. The drawback is, however, that the computation of the residual requires an additional matrix-vector multiplication. For an important, special class of systems this drawback can be circumvented by minimizing in a different norm.

Definition 2.2. The matrix $\boldsymbol{A}$ is said to be *positive semi-definite* if

$$\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}, \tag{2.5a}$$

and

$$(\boldsymbol{A}\boldsymbol{x},\, \boldsymbol{x}) \geqslant 0, \quad \forall \boldsymbol{x}. \tag{2.5b}$$

Moreover, A is *positive definite* if equality in (2.5b) holds for the zero-vector only. These properties will be denoted by PSD and PD, respectively.

Definition 2.3. Let A be positive (semi) definite. Then the (semi) norm $\|\cdot\|_A$ is defined by

$$\|x\|_A=\sqrt{(Ax,\ x)}\,. \tag{2.6}$$

The discretization of the Poisson equation usually leads to a positive definite system of equations, but many other problems originating from partial differential equations yield unsymmetric systems. In these cases the matrix A can often be written as the sum of a positive definite matrix (reflecting the dissipative terms of the PDE), and an skew-symmetric matrix (from convection terms), i.e.

$$A=(A+A^{\mathrm{T}})/2+(A-A^{\mathrm{T}})/2. \tag{2.7}$$

This property is defined in:

Definition 2.4. The matrix A is said to be *positive real* (PR) if its symmetric part $(A+A^{\mathrm{T}})/2$ is positive definite.

Finally we wish to mention that the accuracy of the approximation to the solution of (2.1) is determined by the relation

$$x^{(n)}-x=-A^{-1}r^{(n)}, \tag{2.8}$$

provided that A is regular, from which an estimate could be made if some information about $\|A^{-1}\|$ is available. In case A is singular, A^{-1} should be replaced by the pseudo-inverse A^{+}, under the additional requirement that system (2.1) is consistent. Such information could be generated during the iteration process [12,14], but often a priori knowledge exists. E.g., consider the time integration of the system

$$u'(t)=f(u(t)),\quad u(t)\in\mathbb{R}^m\times\mathbb{R}^m \tag{2.9}$$

with a suitable implicit one-step method (e.g. Euler, trapezoidal, midpoint), and suppose that f satisfies the contractivity condition

$$(f(u)-f(v),\ u-v)\leqslant 0,\quad \forall u,\ v\in\mathbb{R}^m. \tag{2.10}$$

Let x stand for the exact solution of the (nonlinear) equations occurring in the applied one-step method, $x^{(n)}$ be an approximation to x, and $r^{(n)}$ the residual error. Then it is nowadays a standard result [6] that for a small constant C

$$\|x^{(n)}-x\|\leqslant C\|r^{(n)}\|. \tag{2.11}$$

Hence a relatively small residual is then a convenient criterium for the termination of the iteration process.

3. Iterative methods

3.1. The conjugate gradient method

Suppose that A is positive definite. Then, also its inverse A^{-1} is positive definite, and minimization of the residual in the norm $\|\cdot\|_{A^{-1}}$ leads to the *conjugate gradient method* [10], i.e. the iterate $x^{(n+1)}$ satisfies

$$\|b-Ax^{(n+1)}\|_{A^{-1}}\leqslant\|b-Ay\|_{A^{-1}},\quad \forall y\in K^{(n+1)}(A;\ b), \tag{3.1}$$

or equivalently,

$$\| \boldsymbol{x}^{(n+1)} - \boldsymbol{x} \|_A \leqslant \| \boldsymbol{y} - \boldsymbol{x} \|_A, \quad \forall \boldsymbol{y} \in \boldsymbol{K}^{(n+1)}(\boldsymbol{A};\ \boldsymbol{b}). \tag{3.2}$$

The rate of convergence of the conjugate gradient method is known to be dependent on the distribution of the eigenvalues of the matrix $\boldsymbol{A}$. Let $\lambda_{\max}$ and $\lambda_{\min}$ be the largest and smallest eigenvalue of the PD matrix $\boldsymbol{A}$. Then, one has approximately

$$\| \boldsymbol{r}^{(k)} \| \approx \left(1 - 2\sqrt{\lambda_{\min}/\lambda_{\max}}\right)^k \| \boldsymbol{r}^{(0)} \|. \tag{3.3}$$

When the smallest (largest) eigenvalues lie isolated, the rate of convergence improves during the iteration process [28]. The condition number $\boldsymbol{k}(\boldsymbol{A})$ of $\boldsymbol{A}$, which equals $\lambda_{\max}/\lambda_{\min}$, should in general be small to have fast convergence. However, in many applications, e.g. the discretized Poisson equation, $\boldsymbol{A}$ has a quite large condition number. Consequently, it is important then to modify equation (2.1), multiplying with a suitable preconditioner $\boldsymbol{Q}^{-1}$, and solve the equation

$$\boldsymbol{Q}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{Q}^{-1}\boldsymbol{b} \tag{3.4}$$

instead. The condition number of $\boldsymbol{Q}^{-1}\boldsymbol{A}$ may be considerably less than $\lambda_{\max}/\lambda_{\min}$, resulting in a significantly decreased number of iteration steps, at the cost of some additional overhead in constructing and multiplying with $\boldsymbol{Q}^{-1}$. In Section 4 we give a survey of several preconditioners which could be of value for the equations that we want to solve.

The conjugate gradient method is not suitable for nonsymmetry problems, therefore we will now discuss methods that may be used in this case.

3.2. Solving the normal equations

One way to get around the difficulties caused by the unsymmetry of $\boldsymbol{A}$ consists in first deriving the normal equations from (2.1),

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}x = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}, \tag{3.5}$$

and then solving this positive definite system using the conjugate gradient method. However, the condition number of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ is the square of the condition number of $\boldsymbol{A}$, so one might expect slow convergence and, in particular for ill-conditioned systems, roundoff may contaminate the results. The latter disadvantage is avoided in the LSQR method [21], which is equivalent in exact arithmetic. In the LSQR method the Lanczos algorithm is applied to

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ 0 \end{pmatrix}, \tag{3.6}$$

so it is not necessary to form the product $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ explicitly. The costs of the method per iteration are nevertheless increased by a factor two in comparison with an iteration for (2.1), as two matrix-vector multiplications are necessary now.

Another disadvantage of these methods could be that a preconditioning for (3.5) and (3.6) could be more cumbersome than the one for (2.1) as these systems are less sparse, especially in the case of (3.5). However, the methods are guaranteed to converge anyhow (even in a finite number of iterations, neglecting round-off), so they might be of value in situations where other methods, that are cheaper per iteration step, fail.

3.3. The methods orthodir, orthomin and orthores

In the previous subsection the equation (2.1) was modified to a positive definite system. The methods orthodir, orthores and orthomin [34] iterate on (2.1), but they use modified norms and inner products instead of the ones which occur in the conjugate gradient method. For example, let $\boldsymbol{Z}$ be an arbitrary matrix such that $\boldsymbol{ZA}$ is positive definite (a possible choice could be $\boldsymbol{Z}=\boldsymbol{A}^{\mathrm{T}}$). Then, the vector $\boldsymbol{x}^{(n)}$ defined by

$$\| \boldsymbol{x}^{(n)}-\boldsymbol{x} \|_{ZA} \leqslant \| \boldsymbol{y}-\boldsymbol{x} \|_{ZA}, \quad \forall \boldsymbol{y} \in \boldsymbol{K}^{(n)}(\boldsymbol{A};\, \boldsymbol{b}), \tag{3.7}$$

is the nth iterate of the methods orthodir and orthomin, and it can be obtained from the Galerkin equations

$$\left(\boldsymbol{ZA}(\boldsymbol{x}^{(n)}-\boldsymbol{x}),\, \boldsymbol{y}\right)=0, \quad \forall \boldsymbol{y} \in \boldsymbol{K}^{(n)}(\boldsymbol{A};\, \boldsymbol{b}). \tag{3.8}$$

However, the Galerkin equations can be solved also when $\boldsymbol{ZA}$ is PR, which leads to recurrence relations of the form

$$\boldsymbol{x}^{(n+1)}=\boldsymbol{x}^{(n)}+\lambda_n \boldsymbol{q}^{(n)}, \tag{3.9}$$

$$\boldsymbol{q}^{(n)}=\boldsymbol{A}\boldsymbol{q}^{(n-1)}+\sum_{i=0}^{n-1} \beta_{n,i}\boldsymbol{q}^{(i)} \quad \text{(orthodir)}, \tag{3.10a}$$

$$\boldsymbol{q}^{(n)}=\boldsymbol{r}^{(n)}+\sum_{i=0}^{n-1} \alpha_{n,i}\boldsymbol{q}^{(i)} \quad \text{(orthomin)}, \tag{3.10b}$$

$$\boldsymbol{q}^{(0)}=\boldsymbol{r}^{(0)}. \tag{3.11}$$

If both $\boldsymbol{Z}$ and $\boldsymbol{ZA}$ are PD matrices, then (3.10a) and (3.10b) reduce to three-term and two-term recurrence relations, respectively, which are equivalent to the conjugate gradient method.

From the idealized forms of orthodir and orthomin, given by (3.9)–(3.11), one can derive truncated forms orthodir(s) and orthomin(s) by putting the coefficients $\beta_{k,i}$ and $\alpha_{k,i}$ to zero for $i<k-s$. This leads to an enormous reduction in the computational effort, and one can hope that for almost symmetric problems these coefficients are small anyway, as orthodir(2) = orthomin(1) = cg in the symmetric case (i.e. $\boldsymbol{Z}$ and $\boldsymbol{ZA}$ PD). It can be shown that the truncated versions also satisfy a minimizing property,

$$\| \boldsymbol{x}^{(n)}-\boldsymbol{x} \|_{ZA} \leqslant \| \boldsymbol{y}-\boldsymbol{x} \|_{ZA}, \quad \forall \boldsymbol{y} \in \hat{\boldsymbol{K}}^{(s)}, \tag{3.12}$$

where $\hat{\boldsymbol{K}}^{(s)}$ is a subspace of dimension $(s+1)$ of the Krylov subspace $K^{(n)}(A;\, b)$.

The method orthomin(s) seems the more attractive one, as the recurrence relation is simpler in the symmetric case. However, there are examples that the method breaks down if $\boldsymbol{Z}$ is not a PR matrix (see [34]). The method orthodir(s) can be applied for arbitrary matrices $\boldsymbol{Z}$ such that $\boldsymbol{ZA}$ is PR. Although the process may break down with $\boldsymbol{q}^{(n)}=0$, this only occurs when the solution lies in $\boldsymbol{K}^{(n)}(\boldsymbol{A};\, \boldsymbol{b})$, assuming exact arithmetic. Consequently, the exact solution could be recovered then from previous iterates when these are still available, or reconstructed by performing the iteration once more. Saad and Schultz [24] propose a more stable variant of orthodir, which they call GMRES. They show that truncated versions of this algorithm often are to be preferred over other methods. We finally remark that the convergence properties of these truncated methods are not well understood, but numerical results of Axelsson are very promising [34,1].

In the method orthores the recurrence relations,

$$x^{(n+1)} = \gamma_{n+1} f_{n+1,n} r^{(n)} + \sum_{i=0}^{n} f_{n+1,i} x^{(i)} \tag{3.13}$$

are constructed such that the residual vectors $r^{(n)}$ are semi-orthogonal with respect to Z

$$(Zr^{(i)}, r^{(j)}) = 0, \quad j < i, \tag{3.14}$$

and this property remains valid for $j \geqslant i - s$ in the truncated version orthores(s). If Z is positive definite, then the expressions for the coefficients γ_{n+1}, $f_{n+1,i}$ greatly simplify because of the orthogonality of the residuals, so in that case the method could be preferred over orthodir and orthomin. If, moreover, ZA is also PD, then orthores(1) is also equivalent to the conjugate gradient method. However, the method might fail if ZA is not a PR-matrix (see [34]).

In [34] some possible choices for Z are considered, e.g. $Z = A^{\mathrm{T}}$, $Z = A + A^{\mathrm{T}}$, $Z = A^{\mathrm{T}}(A + A^{\mathrm{T}})$ or $Z = A^{\mathrm{T}}Q$ where Q is a preconditioner for (2.1). The methods orthomin, orthores and orthodir can also be applied to the systems which arise after multiplication with a convenient preconditioner (Jacobi, Gauss–Seidel, SSOR, ILU, ADI). The various options will be considered in Section 4.

3.4. Bi-cg and cgs

The bi-conjugate gradient method [8] constructs two sequences of residuals, $r^{(n)}$ and $\hat{r}^{(n)}$, $n = 0, 1, \ldots$ which satisfy similar recurrence relations

$$p^{(n)} = r^{(n)} + \beta_n p^{(n-1)}, \tag{3.15a}$$

$$\hat{p}^{(n)} = \hat{r}^{(n)} + \beta_n \hat{p}^{(n-1)}, \tag{3.15b}$$

$$r^{(n+1)} = r^{(n)} - \alpha_n A p^{(n)}, \tag{3.16a}$$

$$\hat{r}^{(n+1)} = \hat{r}^{(n)} - \alpha_n A^{\mathrm{T}} \hat{p}^{(n)}. \tag{3.16b}$$

The coefficients α_n and β_n are determined by the requirement that the vectors $r^{(i)}$ and $\hat{r}^{(j)}$, $Ap^{(i)}$ and $\hat{p}^{(j)}$ are mutually orthogonal for $i \neq j$. When A is positive definite, and $r^{(0)} = \hat{r}^{(0)}$, the algorithm reduces to the standard conjugate gradient method, and in that case convergence is guaranteed. It seems likely that convergence also occurs if A is almost positive definite. Otherwise the algorithm breaks down when one of the inner products used in the computation of the coefficients α_n and β_n becomes nonpositive.

The explicit occurrence of A^{T} in the computations can be considered as a disadvantage of the bi-cg method. Moreover, the fact that $\hat{r}^{(n)}$ will hopefully converge to zero just as the sequence $r^{(n)}$ is not exploited in the algorithm. These disadvantages are circumvented in the *conjugate gradient squared method* [25]. The inner products used in bi-cg are rewritten using the equalities

$$(\hat{r}^{(n)}, r^{(n)}) = \left(P_n(A^{\mathrm{T}})\hat{r}^{(0)}, P_n(A) r^{(0)}\right) = \left(\hat{r}^{(0)}, \{P_n(A)\}^2 r^{(0)}\right). \tag{3.17}$$

The sequence $\{P_n(A)\}^2 r^{(0)}$, $n = 1, 2, \ldots$, obtained in this way will converge faster than the corresponding sequence $P_n(A) r^{(0)}$ in bi-cg (if there is convergence at all), and no multiplication with A^{T} is required. Sonneveld [25] reports results on convection diffusion equations which compare favourably to a variant of orthomin. For implementations and numerical results see, e.g., [15]. Finally we remark that both algorithms allow for preconditioning.

3.5. Chebyshev-iteration

The elements of the Krylov space $K^{(n)}(A;\ b)$ satisfy the relation

$$y - x = P_n(A)(x^{(0)} - x) \tag{3.18}$$

for some polynomial P_n of degree n, with $P_n(0) = 1$. If P_n is suitably chosen, then y will be a reasonable approximation to the solution x. A good choice is a polynomial which takes on small values in the eigenvalues of A. The conjugate gradient method, for example, implicitly generates a sequence of polynomials which are optimal in achieving this in some sense. When the eigenvalues of A are known to be real, positive and lying in the interval $[a,\ b]$, P_n can be chosen as a shifted Chebyshev polynomial (see e.g. [33]), such that the maximal absolute value on $[a,\ b]$ is minimal. Manteuffel [18] has proposed an algorithm which can be applied in the case where A has complex eigenvalues with positive real part. When all eigenvalues lie in an ellipse in the right half plane with centre d and foci $d + c$, $d - c$, then the rate of convergence is determined by

$$\max_{\lambda} \left| \frac{(d-\lambda) + \sqrt{(d-\lambda)^2 - c^2)}}{d + \sqrt{d^2 - c^2}} \right|. \tag{3.19}$$

The problem here is to find the optimal values of d and c, which parameters are necessary in the Chebyshev-iteration process. In the algorithm the eigenvalues of A are estimated during the iteration process, and the values of d and c are adapted accordingly. In practice the method often shows slow convergence (or even divergence) in the first iteration steps, which improves when better values of d and c are found. The method would suffer less from this drawback, when a large number of similar systems has to be solved, because then information about the optimal choices of d and c is already available from previous iterations. This situation may occur when the linear systems originate from the solution of time-dependent PDEs.

We note that the eigenvalues of A are known to lie in the right half plane if A is PR. Of course, the method can be applied after a suitable preconditioning too. However, care should be taken in that the eigenvalues of the preconditioned system are also in the right half plane. A more detailed discussion is given in Section 4.

3.6. Relaxation methods

Let the matrix A be positive real, and let

$$A = L + D + U \tag{3.20}$$

be a splitting of A in a strictly lower-triangular matrix L, a diagonal D and a strictly upper-triangular matrix U. Then for ω sufficiently small the Jacobi overrelaxation method (JOR) can be applied

$$x^{(n+1)} = x^{(n)} + \omega D^{-1}(b - Ax^{(n)}). \tag{3.21}$$

Here, the admissible values of ω depend on the eigenvalues of the Jacobi matrix $I - D^{-1}A$. Suppose these eigenvalues lie in a rectangle with vertices $\pm\mu_R \pm i\mu_I$. Then,

$$\omega_{\text{opt}} = \begin{cases} 1 & \text{if } (2\mu_R - 1)^2 + 4\mu_I^2 \leqslant 1, \\ \dfrac{1-\mu_R}{(1-\mu_R)^2 + \mu_I^2} & \text{if } (2\mu_R - 1)^2 + 4\mu_I^2 \geqslant 1, \end{cases} \tag{3.22}$$

$$\omega \leqslant \omega_{\max} = 2\omega_{\text{opt}}. \tag{3.23}$$

We note that all eigenvalues of $D^{-1}A$ have positive real part as a consequence of A being PR.

The SOR-method, which is based upon the Gauss–Seidel iteration method (cf. e.g. [33]), is given by

$$x^{(n+1)} = x^{(n)} + \omega D^{-1}(b - Lx^{(n+1)} - Dx^{(n)} - Ux^{(n)}). \tag{3.24}$$

The method is convergent if A is consistently ordered and if all eigenvalues of $I - D^{-1}A$ lie within the ellipse

$$\mu_R^2 + \left(\frac{\omega}{2-\omega}\right)^2 \mu_I^2 = 1. \tag{3.25}$$

Hence, there is convergence for

$$\omega \leqslant \omega_{\max} = 2\sqrt{1-\mu_R^2} \Big/ \left(\mu_I + \sqrt{1-\mu_R^2}\right). \tag{3.26}$$

The expression for the optimal relaxation factor is rather complicated in the case of complex eigenvalues. Moreover, in practice these eigenvalues are not known, so they must be estimated somehow. A possibility then is to use local relaxation, i.e. the relaxation factor is different for the various equations within an iteration step, and the eigenvalues are estimated using a local Fourier analysis with frozen coefficients. Botta and Veldman [4] report satisfactory results for JOR.

We remark that these overrelaxation methods can also be applied to splittings of the matrix A which are more general than (3.20), e.g. when D is a block diagonal matrix (BJOR and BSOR, cf. Varga [33]). The advantage will be that the eigenvalues of the block Jacoby matrix $I - D^{-1}A$ are usually less in modulus than the corresponding ones in the strictly diagonal case, which leads to an improved rate of convergence. However, the complications arising from the inversion of the block diagonal matrix D may annihilate this improvement (see section 4.5).

3.7. Summary of methods

Method	Requirements		Inner products	A	A^T	Remarks
	Necessary	Sufficient				
cg	a	a	2	1	0	conv. rate depends on $k(A)$
cg on A^TA			2	1	1	conv. rate depends on $k(A)^2$
LSQR			2	1	1	conv. rate depends on $k(A)^2$ less sensitive to round-off
orthodir(s)	b	d, e	$s+2$	2	0	additional Z required
orthomin(s)	b, c	d, e	$s+3$	1	0	additional Z required
orthores(s)	b, c	d, e	$s+2$	1	0	additional Z required
Chebyshev	f		0	1	0	convergence depends on eigenvalue estimation
SOR	f		0	1	0	eigenvalue estimation required
bi-cg		a	2	1	1	
cgs		a	2	2	0	conv./div. twice as fast as bi-cg

Explanations: a: A is PSD, b: ZA is PR, c: Z is PR, d: Z is PD, e: ZA is PD, f: A is PR.

4. Preconditioning

In this section we discuss the use of preconditioning with the aim of reduction of the number of iteration steps required to obtain a good approximation to the solution x of equation (2.1). The best preconditioner in some sense is given by the inverse of A (cf. equation (3.4)). Then the solution is attained in one iteration step. The amount of work to construct the inverse, however, will be excessively high in practical circumstances. Consequently, preconditioning must be regarded as a trade off between the cost of constructing and manipulating the preconditioner, and the acceleration of the iteration process. We will survey some preconditioning methods which have proved to be of value in the last decade.

4.1. Scaling by the diagonal of A

The simplest form of preconditioning is scaling of the rows and columns of the matrix A, with e.g. the intention of obtaining a unit diagonal, rows or columns of equal norm, or in special cases a symmetric system. The scaling by the diagonal of A is in some respects optimal, since it approximately minimizes the condition number of $D^{-1}A$ among all diagonal scalings [26]. Forsythe and Strauss [9] proved that it is indeed the optimal scaling if A has property (A) [33, p. 99]. Scaling by the diagonal of A has also the advantage that within an iteration step the number of multiplications is reduced (see e.g. [19]).

4.2. Incomplete LL^{T} factorizations for PD matrices

Let A be a positive definite matrix. An approved method for the solution of (2.1) is obtained using the Cholesky decomposition

$$A = LDL^{\mathrm{T}}, \tag{4.1}$$

where L is a lower triangular matrix with unit diagonal elements. If A is a large sparse matrix this decomposition has a drawback, because in general L does not reflect the sparsity of A. It is observed however that the entries of L corresponding to the zero values of A are usually small. Hence, putting these entries equal to zero, may yield a reasonable approximation of A,

$$A + R = \tilde{L}D\tilde{L}^{\mathrm{T}}, \tag{4.2}$$

where $\| R \|$ is hopefully small compared to $\| A \|$.

In the incomplete factorization methods elements of L are replaced by zero during the factorization process. There is a choice, then, to allow for some fill-in. For example, if A is a sparse structured band matrix originating from the discretized Poisson equation, one could allow for the fill-in of s extra diagonals in L, leading to the IC(s) method [19,20]. Of course, IC(s) with $s > 0$ will yield a better approximation of A than IC(0) at the cost of increased storage and computation time. Meyerink and Van der Vorst proved that these factorization processes do not break down in quite general circumstances.

Definition 4.1. A is said to be an *M-matrix* if A is nonsingular, the off-diagonal entries are nonpositive, and all entries of A^{-1} are nonnegative.

Table 1
Eigenvalues for IC and MIC for a model problem

	IC	MIC
λ_1	0.0342	1.0
λ_2	0.08179	1.0007
λ_{900}	1.2045	9.0068

Theorem 4.2. *If A is a symmetric M-matrix, then for each set $P \subset \{(i, j) \mid i > j\}$ there exists a unique factorization $\boldsymbol{A} = \boldsymbol{LDL}^{\mathrm{T}} - \boldsymbol{R}$, such that*

$$l_{ij} = 0 \quad \textit{if } (i, j) \in P, \tag{4.3a}$$

$$r_{ij} = r_{ji} = 0 \quad \textit{if } (i, j) \notin P. \tag{4.3b}$$

Further it is proved that the iteration process

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + (\boldsymbol{LDL}^{\mathrm{T}})^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(n)}) \tag{4.4}$$

converges under the conditions of this theorem. The eigenvalue ratio $\lambda_{\max}/\lambda_{\min}$ of $(\boldsymbol{LDL}^{\mathrm{T}})^{-1}\boldsymbol{A}$ is usually much smaller than the condition number of the original matrix A (see e.g. [20], [30] and [2]), so an iterative procedure applied to

$$(\boldsymbol{LDL}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{LDL}^{\mathrm{T}})^{-1}\boldsymbol{b} \tag{4.5}$$

then converges faster than the same method applied to (2.1). In [20] examples of IC(s) combined with conjugate gradient are given.

A slight modification of the incomplete factorization is obtained by lumping the neglected values of $\boldsymbol{L}$ to the diagonal [11]. This so-called MIC factorization is feasible if A is a weakly diagonally dominant matrix. Van der Vorst [30] shows that this modification has a strong effect on the eigenvalue distribution of the preconditioned matrix. He presents an example for which the condition number for MIC is much smaller than for IC (see Table 1).

However the smallest eigenvalues for IC lie more isolated which is in practise often more advantageously for conjugate gradient (see Section 3, after (3.3)). Axelsson and Lindskog [2,3] discuss a class of relaxed incomplete factorization methods (RIC) where the lumping of elements to the diagonal depends on a parameter $\omega \in [0, 1]$. For $\omega = 0$ IC is obtained, and $\omega = 1$ yields MIC. The parameter ω is then chosen such that the condition number is small whereas the smaller eigenvalues remain reasonably isolated. Experiments suggest that values of ω slightly less than 1 give quite good results.

4.3. *Incomplete factorization for unsymmetric matrices*

In the general case where A is an unsymmetric matrix, similar ideas as in Section 2 can be applied. E.g., for M-matrices the result of Theorem 4.2 remains valid. However, it is often difficult to establish A to be an M-matrix, especially if the off-diagonal entries are large compared with the diagonal. Moreover, proofs about the position of the eigenvalues are much more complicated or even not available. The factorization of the symmetric part $A + A^{\mathrm{T}}$ may be an alternative, then. We quote the following standard results.

Theorem 4.3. *The eigenvalues of* $(A+A^{\mathrm{T}})^{-1}A$ *have all positive real part* 0.5.

Theorem 4.4. *If* $(A+A^{\mathrm{T}})/2$ *is an M-matrix, and* $\boldsymbol{LDL}^{\mathrm{T}}$ *its incomplete Cholesky decomposition, then all eigenvalues of* $(\boldsymbol{LDL}^{\mathrm{T}})^{-1}A$ *have positive real part.*

Consequently, Chebyshev iteration could be used in both cases. When the matrix-vector product $(A+A^{\mathrm{T}})^{-1}y$ can be generated efficiently, e.g. by a Fast Poisson Solver, the first possibility becomes attractive. Otherwise one may resort to an incomplete factorization of the symmetric part of A (for comparisons, see [29]).

Now, let us consider a decomposition of A itself,

$$A=\boldsymbol{LDU}, \tag{4.6}$$

with $\boldsymbol{U}$ an upper triangular matrix with unit diagonal. Gustafsson [11] states a result for the special case of A being (weakly) diagonally dominant.

Definition 4.5. The matrix A is said to be *weakly diagonally dominant if* $a_{ii} \geqslant \Sigma_j \mid a_{ij} \mid$, $i=1,\ldots,N$.

Theorem 4.6. *Let A be weakly diagonally dominant. Then the modified incomplete factorizations* $\boldsymbol{LDU}$ *exist with positive diagonal matrix* $\boldsymbol{D}$. *The same holds for incomplete factorizations of a diagonally dominant matrix* A.

For M-matrices a result of Van der Vorst [29] is useful:

Theorem 4.7. *If A is an M-matrix, then an incomplete factorization* $\boldsymbol{LDU}$ *of A exists, and all eigenvalues of* $(\boldsymbol{LDU})^{-1}A$ *have positive real part.*

When A is not an M-matrix, the existence of an incomplete factorization could be doubtful. This situation may arise if the off-diagonal entries are large compared with the diagonal, e.g. when A is obtained from the discretization of first order derivatives with central differences. An alternative is then given in some situations by the decomposition

$$A=\boldsymbol{LD}^{-1}\boldsymbol{U}-\boldsymbol{R}, \tag{4.7}$$

where $\boldsymbol{L}$, $\boldsymbol{D}$, $\boldsymbol{U}$ and $\boldsymbol{R}$ satisfy

$$\operatorname{diag}(\boldsymbol{L})=\operatorname{diag}(\boldsymbol{U})=\boldsymbol{D}, \tag{4.8a}$$

$$l_{ij}=a_{ij},\quad u_{ji}=a_{ji},\qquad i>j, \tag{4.8b}$$

$$\operatorname{diag}(\boldsymbol{R})=(\sigma-1)\operatorname{diag}(A)\quad \text{for some } \sigma>1. \tag{4.8c}$$

For $\sigma=1$ the elements of $\boldsymbol{D}$ are all positive if $A+A^{\mathrm{T}}$ is an M-matrix [29]. However, the factors $\boldsymbol{L}$ and $\boldsymbol{U}$ may be ill-conditioned when the skew-symmetric part of A is large, while A itself is reasonably well-conditioned. The ill-conditioning is prevented by choosing a suitable value of $\sigma>1$. For σ sufficiently large, the elements of $\boldsymbol{D}$ will be comparable in size to those of $\boldsymbol{L}$ and $\boldsymbol{U}$. Van der Vorst [29] gives an example for which an optimal value σ_{opt} is determined from a quadratic equation. The number of steps in a Chebyshev iteration process then turns out to be minimal for values of σ in the neighbourhood of this optimal value. About the applicability of these decompositions in combination with Chebyshev iteration Van der Vorst [31] states:

Theorem 4.8. *If $A + A^{\mathrm{T}}$ is an M-matrix, and if $\boldsymbol{L}_\sigma \boldsymbol{D}_\sigma^{-1} \boldsymbol{U}_\sigma$ is the incomplete decomposition of $A + (\sigma - 1)\mathrm{diag}(A)$ according to (4.7), then there exists a $\sigma_0 \geqslant 1$ such that the eigenvalues of $(\boldsymbol{L}_\sigma \boldsymbol{D}_\sigma^{-1} \boldsymbol{U}_\sigma)^{-1} A$ have positive real part for $\sigma \geqslant \sigma_0$.*

In practical applications it is, of course, not clear whether the given σ satisfies the condition of this theorem. Moreover, it is hard to determine anything like σ_{opt} when the elements of A do not originate from a constant coefficient model problem. Therefore, the following parameterless decomposition, denoted by $\boldsymbol{L}_{\mathrm{EQ}} \boldsymbol{D}_{\mathrm{EQ}}^{-1} \boldsymbol{U}_{\mathrm{EQ}}$, is a useful alternative in those situations.

Form $\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{U}$ according to (4.8), (4.9a)

Replace the diagonal entries of $\boldsymbol{L}$, $\boldsymbol{D}$, $\boldsymbol{U}$,

during the computation of $\boldsymbol{D}$ from (4.8c), by

$$d_k = \max\left\{ d_{kk},\ \sum_{j=1}^{k-1} |a_{kj}|,\ \sum_{j=k+1}^{n} |a_{kj}| \right\}. \tag{4.9b}$$

Although there is no theorem available stating that the eigenvalues of $(\boldsymbol{L}_{\mathrm{EQ}} \boldsymbol{D}_{\mathrm{EQ}}^{-1} \boldsymbol{U}_{\mathrm{EQ}})^{-1} A$ have positive real part, results in combination with Chebyshev iteration are promising [29, 31].

We finally remark that the versions of the decompositions which allow for no fill-in can be implemented in the conjugate gradient method or the Chebyshev iteration almost without an increase in computational effort when compared with the unpreconditioned process [7,29], and the same is true for cgs and bi-cg [15].

4.4. Summary of preconditioners

Method	A symm.	Requirements	Eigenvalues of $A^{-1}A$
IC(s)	y	A M-matrix	$\lambda > 0$
MIC(s)	y	A weakly diagonally dominant	$\lambda > 0$
RIC(ω)	y	A M-matrix ($\omega < 1$)	$\lambda > 0$
RIC(ω)	y	A weakly diagonally dominant ($\omega = 1$)	$\lambda > 0$
IC on $A + A^{\mathrm{T}}$	n	$A + A^{\mathrm{T}}$ M-matrix	Re $\lambda > 0$
Cholesky on $A + A^{\mathrm{T}}$	n	A PR	Re $\lambda = 0.5$
MI $\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{U}$	n	A weakly diagonally dominant	
I $\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{U}$	n	A M-matrix	Re $\lambda > 0$
$\boldsymbol{L}_1 \boldsymbol{D}_1^{-1} \boldsymbol{U}_1$	n	$\boldsymbol{L}_1$ and $\boldsymbol{U}_1$ could be ill-conditioned	
$\boldsymbol{L}_\sigma \boldsymbol{D}_\sigma^{-1} \boldsymbol{U}_\sigma$	n	$A + A^{\mathrm{T}}$ M-Matrix; estimation of σ required	Re $\lambda > 0$ for $\sigma \geqslant \sigma_0$
$\boldsymbol{L}_\sigma \boldsymbol{D}_\sigma^{-1} \boldsymbol{U}_\sigma$	n	Estimation of σ required	
$\boldsymbol{L}_{\mathrm{EQ}} \boldsymbol{D}_{\mathrm{EQ}} \boldsymbol{U}_{\mathrm{EQ}}$	n		

4.5. Other preconditioners

When we are willing to spend more computational effort in the construction of the preconditioner, other methods come into view. For example, incomplete block factorizations [5,3], line relaxation methods, alternating direction methods [33]. Experiences indicate, however, that the block incomplete preconditioners for 3D problems are less promising than they seem to be for 2D problems [17]. In those methods a (simple) implicit system must be solved in each iteration step. Although the solution process can be implemented almost as efficient as ordinary matrix-vector multiplications on scalar computers, for vector machines the inherent recursion is often a serious drawback. Therefore, it is expected that the possible gain in the convergence acceleration is contaminated by the slow calculations within the iteration steps.

5. What to do in practical situations?

Although it is often not clear which method is optimal in many cases, we will try to provide some knowledge and some guidelines in this section.

5.1. Self-adjoint elliptic partial differential equations

The standard 5-point finite difference discretization of a self-adjoint elliptic partial differential equation, like, e.g.

$$\left(a(x, y)u'_x\right)'_x - \left(b(x, y)u'_y\right)'_y + c(x, y)u = f(x, y), \tag{5.1}$$

with boundary conditions of the form

$$\alpha(x, y)u + \beta(x, y)u_n = \gamma(x, y), \tag{5.2}$$

over a rectangular grid leads to a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. We assume that a, $b > 0$ and $c \geqslant 0$. In this case, when both c and α are not identical to zero along the complete boundary, $\boldsymbol{A}$ is a symmetric M-matrix (and hence $\boldsymbol{A}$ is PD). The linear system can be solved by the conjugate gradient method and a (modified) incomplete Choleski decomposition can be used as a preconditioner. When α vanishes along the entire boundary, and c is identically zero then we have a pure Neumann-problem. In that case $\boldsymbol{A}$ is singular, however it is proven in [16] that an incomplete decomposition exists, and $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ can still be solved by the (preconditioned) conjugate gradient method. Care must be taken that $\boldsymbol{b}$ is in the span of the columns of $\boldsymbol{A}$, which can be done by simply removing the null space component of $\boldsymbol{b}$ with respect to $\boldsymbol{A}$. For details, see [16].

When the coefficient c is negative, then $\boldsymbol{A}$ is not necessarily positive definite and an alternative might be to solve implicitly the normal equations $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ (see e.g. [21]), using the conjugate gradient method. Another possibility is to use a Lanczos-type method, i.e. to form a basis for the Krylov-space, to project the given linear system on this space and to solve the projected system (see, e.g. [23], [22] and [32]).

5.2. Non-symmetric problems

Partial differential equations with first order terms which cannot be written in the form (5.1), usually lead, after discretization, to non-symmetric linear systems. As an example consider the convection diffusion equation

$$-\epsilon\Delta u + d(x, y)u'_x + e(x, y)u'_y = f. \tag{5.3}$$

If one uses central differences for the first order derivative terms then the grid spacings can be chosen so small that all off-diagonal elements in $\boldsymbol{A}$ are nonpositive. In that case $\boldsymbol{A}$ is an

M-matrix and one might use, e.g. the preconditioned Chebyshev iteration method, some truncated orthogonal method like orthomin, GMRES or the cgs method.

Often it is not very practical to choose the grid spacings so small as to have an M-matrix as the resulting matrix. Unless d and e vary wildly with respect to the given grid one has then the situation that $A+A^{\mathrm{T}}$ is an PD M-matrix and one could use one of the orthogonal methods or the Chebyshev method with a preconditioning based upon (approximate) inversion of $A+A^{\mathrm{T}}$ (see [29]). It is also possible to use a stabilized incomplete decomposition of A itself and to apply the preconditioned Chebyshev iteration (or to solve the normal equations). In [29] and [31] several variants are discussed and compared.

Another way to avoid problems in A is to use one-sided differences, such that the diagonal elements in A increase, compared with the terms coming from the $\epsilon\Delta u$ part. In this case A is still an M-matrix and, e.g., without any problem an incomplete LU decomposition can be constructed as a preconditioner for the orthogonal methods or the Chebyshev method. In [29] experiments are reported which show that central differences may lead to both less iterations and a smaller discretization error.

References

[1] O. Axelsson, *A Generalized Conjugate Gradient Direction Method and its Application on a Singular Perturbation Problem*, Lecture Notes Math. **773** (Springer, Berlin/Heidelberg/New York, 1980) 1–11.

[2] O. Axelsson and G. Lindskog, On the eigenvalue distribution of a class of preconditioning methods, *Numer. Math.* **48** (1986) 479–498.

[3] O. Axelsson and G. Lindskog, On the rate of convergence of the preconditioned conjugate gradient method, *Numer. Math.* **48** (1986) 499–523.

[4] E.F.F. Botta and A.E.P. Veldman, On local relaxation methods and their application to convection-diffusion equations, *J. Comp. Phys.* **48** (1982) 127–149.

[5] P. Concus, G.H. Golub and G. Meurant, Block preconditioning for the conjugate gradient method, *SIAM J. Sci. Stat. Comput.* **6** (1985) 220–252.

[6] K. Dekker and J.G. Verwer, *Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations* (North-Holland, Amsterdam/New York/Oxford, 1984).

[7] S.C. Eisenstat, Efficient implementation of a class of preconditioned conjugate gradient methods, Research Report No. 185, Yale University.

[8] R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems*, Lecture Notes Math. **506** (Springer, Berlin/Heidelberg/New York, 1976) 73–89.

[9] G. Forsythe and E.G. Strauss, On best conditioned matrices, *Proc. Amer. Math. Soc.* **6** (1955) 340–345.

[10] G.H. Golub and C.F. van Loan, *Matrix Computations* (North Oxford Academic, Oxford, 1983).

[11] I. Gustafsson, A class of first order factorization methods, *BIT* **1** (1978) 142–156.

[12] L.A. Hageman and D.M. Young, *Applied Iterative Methods* (Academic Press, New York, 1981).

[13] A. Jennings, Influence of the eigenvalue spectrum on the convergence rate of the conjugate gradient method, *J. Inst. Maths Applics* **20** (1977) 61–72.

[14] E.F. Kaasschieter, A Fortran implementation of the preconditioned method of conjugate gradients, Report 85-33, Faculty of Mathematics and Informatics, Delft University of Technology; and, *BIT*, to appear.

[15] E.F. Kaasschieter, The solution of non-symmetric linear systems by bi-conjugate gradients or conjugate gradients squared, Report 86-21, Faculty of Mathematics and Informatics, Delft University of Technology.

[16] E.F. Kaasschieter, A finite element conjugate gradient method for the solution of the pure Neumann boundary value problem, Report 87-29, Faculty of Mathematics and Informatics, Delft University of Technology.

[17] R. Kettler, Linear multigrid methods in numerical reservoir simulation, Thesis, Department of Mathematics and Informatics, Delft University of Technology.

[18] T.A. Manteuffel, The Tchebychev iteration for nonsymmetric linear systems, *Numer. Math.* **28** (1977) 307–327.

[19] J.A. Meyerink and H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Math. Comp.* **31** (1977) 148–162.

[20] J.A. Meyerink and H.A. van der Vorst, Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems, *J. Comp. Phys.* **44** (1981) 134–155.

[21] C.C. Paige and M.A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software* **8** (1982) 43–71.

[22] B.N. Parlett, A new look at the Lanczos algorithm for solving symmetric systems of linear equations, *Lin. Algebra Appl.* **29** (1980) 323–346.

[23] Y. Saad, On the Lanczos method for solving symmetric linear systems with several right-hand sides, Techn. Report YALEU/DSC/RR-396, Yale University, New Haven.

[24] Y. Saad and M.H. Schultz, GMRES, A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7** (3) (1986) 856–869.

[25] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, Delft University of Technology Department of Mathematics and Informatics, Report 84-16.

[26] A. van der Sluis, Condition numbers and equilibration of matrices, *Numer. Math.* **14** (1969) 14–23.

[27] A. van der Sluis and H.A. van der Vorst, The convergence behavior of Ritz values in the presence of close eigenvalues, *Lin. Algebra Appl.* **88/89** (1987) 651–694.

[28] A. van der Sluis and H.A. van der Vorst, The rate of convergence of conjugate gradients, *Numer. Math.* **48** (1986) 543–560.

[29] H.A. van der Vorst, Iterative solution for certain sparse linear systems with non-symmetric matrix arising from PDE-problems, *J. Comp. Phys.* **44** (1981) 1–19.

[30] H.A. van der Vorst, Preconditioning by incomplete decompositions, Thesis, University of Utrecht.

[31] H.A. van der Vorst, Stabilized incomplete LU-decompositions as preconditionings for the Tchebycheff iteration, in: D.J. Evans, Ed., *From Preconditioning Methods: Theory and Applications* (Gordon and Breach, New York/London/Paris, 1983).

[32] H.A. van der Vorst, An alternative solution method for solving $f(A)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix A, *J. Comput. Appl. Math.* **18** (1987) 249–263.

[33] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962).

[34] D.M. Young and K.C. Jea, Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods, *Lin. Algebra Appl.* **34** (1980) 159–194.