ELSEVIER

Journal of Computational and Applied Mathematics 77 (1997) 35–51

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Projection methods for linear systems

C. Brezinski *

Laboratoire d'Analyse Numérique et d'Optimisation UFR IEEA – M3, Université des Sciences et Technologies de Lille, 59655 — Villeneuve d'Ascq cedex, France

Received 14 March 1996; revised 9 August 1996

Abstract

The aim of this paper is, first, to give a unified framework for deriving several known projection methods for solving systems of linear equations. We shall show that all these methods follow from a unique minimization problem. The particular cases of the methods of steepest descent, Richardson and conjugate gradients will be treated in details. Then, projection acceleration procedures for accelerating the convergence of an arbitrary iterative method will also be proposed and discussed.

Keywords: Linear systems; Projection; Acceleration

AMS classification: 65F10, 65B99

0. Introduction

The aim of this paper is, first, to give a unified framework for deriving several known projection methods for solving arbitrary systems of linear equations. Usually, these methods are obtained by various approaches: projection on different planes, minimization of the next residual or, when the matrix of the system is symmetric and positive definite, minimization of a quadratic functional. We shall show that all these approaches follow, in fact, from a unique minimization problem based on a variational formulation of the solution of a system of linear equations with a symmetric positive definite matrix and that several projection methods, which are usually presented quite independently, can be derived from it. The particular cases of the methods of steepest descent, Richardson and conjugate gradients will be treated in details. As far as I know, the synthesis presented here is new. It clarifies the connections between the methods studied and simplifies and unifies their presentation. Then, based on the previous ideas, projection acceleration procedures for accelerating the convergence of an arbitrary iterative method are proposed and studied. Finally, connections with the minimal

* E-mail: brezinsk@omega.univ.-lille1.fr

PII S0377-0427(96)00121-5

residual smoothing algorithm [43, 42] and the hybrid procedure [8] are discussed. More details could be found in [7].

1. Variational formulation

Let X and Y be subspaces of an inner product space on $\mathbb{R}$ and $M : X \longrightarrow Y$ a linear symmetric positive definite operator. It is well known (see, e.g. [11, 37]) that the solution $x \in X$ of the equation $Mx = c$, where $c \in Y$, is also the unique vector minimizing the strictly convex quadratic functional

$$J(y) = \tfrac{1}{2}(y, My) - (y, c),$$

i.e., $\forall y \in X, J(x) \leqslant J(y)$. Indeed, the gradient $\nabla J(u)$ of J at the point u is the opposite of the residual r, $\nabla J(u) = Mu - c = -r$, and $\nabla J(u) = 0$ if and only if $u = x$. Moreover, $J(x) = -\frac{1}{2}(x, Mx) = -\frac{1}{2}(c, M^{-1}c)$. Conversely, if x minimizes J over X, we have $\forall z \in X, (Mx - c, z) = 0$. However, this relation does not always imply that x is the solution of $Mx = c$. Such a property depends on X and Y. It is true, in particular, if $Y \subseteq X$ (since, for $z = Mx - c \in Y \subseteq X, (z, z) = 0$ implies $z = 0$).

Let now $Mx = c$ be a system of linear equations with a symmetric positive definite matrix M. The idea of using the preceding variational approach for solving this system is due to Temple [45]. In that case, if $\forall z$, $(Mx - c, z) = 0$ then x is the solution of the system $Mx = c$. Let u be an approximation of the solution x, z an arbitrary nonzero vector called the *search direction* or the *direction of descent* and λ a parameter called the *stepsize*.

Since M is symmetric, we have

$$J(u - \lambda z) = J(u) + \lambda(z, r) + \tfrac{1}{2}\lambda^2(z, Mz) \tag{1}$$

with $r = c - Mu$.

The value of λ minimizing $J(u - \lambda z)$ is given by

$$\lambda = -(z, r)/(z, Mz). \tag{2}$$

It corresponds to a minimum since J is strictly convex and, for this value of λ, we have

$$J(u - \lambda z) = J(u) - \frac{1}{2}\frac{(z, r)^2}{(z, Mz)}.$$

Thus, taking z such that $(z, r) \neq 0$ ensures a reduction in the value of J. Formula (2) for λ can be used even if the matrix M is arbitrary. However, in this case, the scalar product in the denominator can be zero.

Setting $y = u - \lambda z$ and $\rho = c - My$, we see that $(\rho, z) = 0$. We have $\rho = r + \lambda Mz$ and, so, if Mz is collinear to r (i.e., $Mz = \alpha r$), we have $\rho = 0$ and $y = x$. Thus, since ρ is independent of α, the best choice for z is $z = M^{-1}r$. Since this choice is impossible in practice, we could take $z = Cr$ where C is an approximation, in some sense, of M^{-1}. Such a matrix C is called a preconditioner and some choices are discussed in [6].

We have

$$\rho = r - \frac{(Mz)z^{\mathrm{T}}r}{(z, Mz)} = \left(I - \frac{(Mz)z^{\mathrm{T}}}{(z, Mz)}\right) r = Pr,$$

where $P = I - (Mz)z^{\mathrm{T}}/(z, Mz)$. It is easy to see that $P^2 = P$ which shows that ρ is the oblique projection of r on $z^{\perp}$ along Mz.

It is easy to see that $J(u) - J(x) = \frac{1}{2}(r, M^{-1}r)$ and that $\forall z, J(y) - J(x) = \frac{1}{2}(\rho, M^{-1}\rho)$.

Let us now examine in more detail the particular case where $z = r$. This choice for the direction of descent is called *steepest descent* since it corresponds to the direction of the gradient of J at the point u. It was first proposed by Temple [45] but it can be, in fact, traced back to Cauchy [10]. Using Kantorovich inequality [28, p. 83]

$$\frac{(v, v)^2}{(v, Mv)(v, M^{-1}v)} \geqslant 1 - \left(\frac{\kappa - \kappa^{-1}}{\kappa + \kappa^{-1}}\right)^2,$$

where $\kappa = \|M\| \cdot \|M^{-1}\|$ is the condition number of the matrix M and the norm is the l_2–norm, we obtain

$$\frac{(\rho, M^{-1}\rho)}{(r, M^{-1}r)} \leqslant \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 < 1.$$

This result shows the gain brought by the steepest descent.

2. Solving linear systems

Let us now apply this variational formulation to the solution of the system of linear equations $Ax = b$ where A is an arbitrary nonsingular matrix. The case where A is symmetric positive definite will also be treated.

For transforming an arbitrary system into an equivalent one with a symmetric positive definite matrix we shall consider the following possibilities:

(i) Symmetric positive-definite case: $Ax = b$ with A symmetric positive definite.
(ii) Normal residuals: $A^{\mathrm{T}}Ax = A^{\mathrm{T}}b$.
(iii) Normal equations: $AA^{\mathrm{T}}x' = b$ with $x = A^{\mathrm{T}}x'$.
(iv) Expanded system: We consider the system $My = c$ with

$$M = \begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix} \quad \text{and } c = \begin{pmatrix} b \\ b' \end{pmatrix},$$

where b' is an arbitrary vector and $y = (x', x)^{\mathrm{T}}$. The matrix M is symmetric but indefinite and the system is equivalent to $Ax = b$ and $A^{\mathrm{T}}x' = b'$. Thus, the variational formulation can still be applied to this system, since the symmetry is the only property which is required in the scalar products, i.e., for writing (1) and obtaining the formula (2) for λ. It must be noticed that, in this case, the functional J is no longer convex and, thus, the sign of $J(y) - J(x)$ can depend on y. However, the solution x of the system $Mx = c$ is still the unique vector such that $\nabla J(x) = 0$. An important practical point is that divisions by zero (that is *breakdowns*) could now occur since there is no guarantee that scalar products of the form (v, Mv) be nonzero for any nonzero vector v.

The preceding results will be used in two different ways
- for constructing an iterative method of the form $x_{n+1} = x_n - \lambda_n z_n$,
- for accelerating a given iterative method (x_n) by $y_n = x_n - \lambda_n z_n$.

The first strategy will be called a *projection iterative method*, while the second one will be called a *projection acceleration procedure*. Let us mention that, in the literature, what is usually called acceleration of a basic iterative method consists, in fact, of replacing the basic iterations by modified ones. For example, Richardson acceleration (or extrapolation) consists of replacing the basic iterations $x_{n+1} = Tx_n + c$ by $x_{n+1} = \gamma_n(Tx_n + c) + (1 - \gamma_n)x_n$, see [46], or by more complicated ones [14, 32–34, 36].

2.1. Projection iterative methods

For solving the system $Mx = c$ with M symmetric positive definite, we consider the iterative method

$$x_{n+1} = x_n - \lambda_n z_n,$$
$$r_{n+1} = r_n + \lambda_n M z_n,$$

with

$$\lambda_n = -\frac{(z_n, r_n)}{(z_n, Mz_n)} \tag{3}$$

and $r_n = c - Mx_n$.

The choice $z_n = -\nabla J(x_n) = r_n$ corresponds to the method of steepest descent introduced in [45]. This method always converges [11, Theorem 8.4.3, p. 189]. In that case, we have $\|x_n - x\| \leqslant \|r_n\|/\lambda_{\min}$ where $\lambda_{\min}$ is the smallest eigenvalue of M and r_n and r_{n+1} are orthogonal. The choice $(z_i, Mz_n) = 0$ for $i = 0, \ldots, n-1$ corresponds to the conjugate gradients method [27] which terminates in p iterations at most where p is the dimension of the system. When applied to the normal equations it is called CGNE [13], while it is called CGNR when applied to the normal residuals [27]; see [18] for a review and also [15]. The convergence behavior of these methods is discussed in [38] where it is shown that it is governed by the singular values of A. These choices will be discussed below.

Our strategies give

(i) *Symmetric positive definite case*: We take $M = A$ and $c = b$ and we have

$$x_{n+1} = x_n - \lambda_n z_n \quad \text{with} \quad \lambda_n = -\frac{(z_n, r_n)}{(z_n, Az_n)}$$

and $r_n = b - Ax_n$.

As explained in Section 1, this choice minimizes the norm $(x_{n+1} - x, A(x_{n+1} - x))$ and we have

$$\|x_{n+1} - x\|_A^2 = \|x_n - x\|_A^2 - \frac{(z_n, r_n)^2}{(z_n, Az_n)},$$

where $\|u\|_A^2 = (u, Au)$. Thus $\|x_{n+1} - x\|_A \leqslant \|x_n - x\|_A$. The usual projection methods for symmetric positive definite systems are recovered; see, for e.g., [21]. When z_n is one of the vectors of the canonical basis of $\mathbb{R}^p$, the Gauss–Seidel iterations are recovered for the cyclic choice, while Southwell

relaxation procedure [44] is obtained for the choice corresponding to the largest component (in absolute value) of r_n. The method of steepest descent corresponds to the choice $z_n = r_n$ while, in the method of conjugate gradients, the vectors z_n are such that $\forall i \neq j, (z_i, Az_j) = 0$. These two particular cases will be discussed below.

(ii) *Normal residuals*: We take $M = A^T A$ and $c = A^T b$ and we have

$$x_{n+1} = x_n - \lambda_n z_n \quad \text{with} \quad \lambda_n = -\frac{(Az_n, r_n)}{(Az_n, Az_n)}$$

and $r_n = b - Ax_n$.

This choice minimizes (r_{n+1}, r_{n+1}). We have

$$\|r_{n+1}\|^2 = \|r_n\|^2 - \frac{(r_n, Az_n)^2}{(Az_n, Az_n)}.$$

If θ_n denotes the angle between r_n and Az_n, we have $\|r_{n+1}\|^2 = \|r_n\|^2 \sin^2 \theta_n$. Thus, $\|r_{n+1}\| \leqslant \|r_n\|$ which shows that the convergence is monotone. Defining x'_n by $x'_n = x_n - z_n$, this projection iterative method is equivalent to applying the minimal residual smoothing (MRS) procedure [43, 42] to the sequence (x'_n). If $z_n = r_n$, we obtain the Richardson iterative method [41] which is equivalent to applying the method of steepest descent to the normal residuals since, in that case, $\lambda_n = -(z_n, A^T b - A^T A x_n)/(z_n, A^T A z_n) = -(Az_n, r_n)/(Az_n, Az_n)$. If $z_n = C_n r_n$ for some matrix C_n, the method can be considered as a preconditioned Richardson method [40, pp. 38ff]. Such a choice is discussed in detail in [6].

(iii) *Normal equations*: We take $M = AA^T$ and $c = b$. Since u is now an approximation of the solution x' of the system $AA^T x' = b$, we shall assume that our iterative method produces iterates (x'_n) that are approximations of x'. We have

$$x'_{n+1} = x'_n - \lambda_n z_n, \tag{4}$$

with

$$\lambda_n = -\frac{(z_n, r'_n)}{(z_n, Mz_n)} = -\frac{(z_n, r'_n)}{(A^T z_n, A^T z_n)}$$

and $r'_n = b - AA^T x'_n$. Multiplying both sides of (4) by A^T and setting $x_n = A^T x'_n$ and $r_n = b - Ax_n = b - AA^T x'_n = r'_n$, we obtain

$$x_{n+1} = x_n - \lambda_n A^T z_n \quad \text{with} \quad \lambda_n = -\frac{(z_n, r_n)}{(A^T z_n, A^T z_n)}.$$

Thus, the usual projection methods for systems with an arbitrary matrix are recovered [21]. This choice minimizes $(x_{n+1} - x, x_{n+1} - x)$ and, since $(A^T z_n, x_{n+1} - x) = 0$, we have

$$\|x_{n+1} - x\|^2 = \|x_n - x\|^2 - \frac{(z_n, r_n)^2}{(A^T z_n, A^T z_n)}.$$

Since $(z_n, r_n) = (A^T z_n, x - x_n)$, we have

$$\|x_{n+1} - x\|^2 = \|x_n - x\|^2 \sin^2 \varphi_n,$$

where φ_n is the angle between the vectors $A^{\mathrm{T}}z_n$ and $x-x_n$ and, thus, $\|x_{n+1}-x\| \leqslant \|x_n-x\|$. For the choice $z_n = r_n$, this method can be considered as an extension of the method of steepest descent to an arbitrary matrix.

Several choices for the directions of descent have been studied in the literature.

(a) *Gastinel method*: It consists of taking for z_n a vector such that $(z_n, r_n)=\varphi(r_n)$ where $\varphi(r_n)$ designates an arbitrary norm of the vector r_n. This method, due to Gastinel [19, 20], always converges. If φ is the l_1–norm, then z_n is the vector whose components are the signs of the corresponding component of r_n. If φ is the l_2–norm, then $z_n = r_n/(r_n, r_n)^{1/2}$. If φ is the l_∞–norm, the ith component of z_n is 0 if $i \neq k$ and, for $i = k$, equal to the sign of the kth component of r_n where k is the smallest index such that the absolute value of the kth component of r_n is equal to $\|r_n\|_\infty$. This last choice is similar to the choice made in Southwell relaxation method [44].

(b) *Kaczmarz method*: Let k be defined by $n+1 = k \pmod p$. We shall take $z_n = e_k$, the kth vector of the canonical basis of $\mathbb{R}^p$. This method is similar to Gastinel's for the l_∞–norm, the only difference being that, now, the vectors e_k are taken in a cyclic order. This method is due to Kaczmarz [31]. Kaczmarz method for solving $Ax = b$ coincides with the Gauss–Seidel iteration for the normal equations [24, Lemma 8.2.1, p. 211]. Let us also quote Cimmino method [12] which is the same as Kaczmarz's except that λ_n is replaced by $2\lambda_n$.

Defining x'_n by $x'_n = x_n - z_n$, the projection iterative method is equivalent to applying the MRS to the sequence (x'_n) but with a choice of λ_n minimizing the error $x_{n+1}-x$ instead of the residual r_{n+1} as proposed in [43, 42]. However, it must be noticed that such a method requires using A^{T}.

(iv) *Expanded system*: Let us set

$$\widetilde{x}_n = \begin{pmatrix} x'_n \\ x_n \end{pmatrix}, \qquad \widetilde{z}_n = \begin{pmatrix} z'_n \\ z_n \end{pmatrix}, \qquad \widetilde{r}_n = \begin{pmatrix} r'_n = b' - A^{\mathrm{T}}x'_n \\ r_n = b - Ax_n \end{pmatrix}.$$

It must be noticed that

$$M\widetilde{z}_n = \begin{pmatrix} Az_n \\ A^{\mathrm{T}}z'_n \end{pmatrix}$$

and that

$$r(\widetilde{x}_n) = c - M\widetilde{x}_n = \begin{pmatrix} r_n \\ r'_n \end{pmatrix} \neq \widetilde{r}_n.$$

We have

$$\widetilde{x}_{n+1} = \widetilde{x}_n - \lambda_n \widetilde{z}_n,$$
$$r(\widetilde{x}_{n+1}) = r(\widetilde{x}_n) + \lambda_n M\widetilde{z}_n.$$

It follows that $(\widetilde{z}_n, M\widetilde{z}_n) = 2(z'_n, Az_n)$. Moreover, since $\widetilde{r}_n$ must be replaced by $r(\widetilde{x}_n)$ in the relations (1) and (2), we have

$$\lambda_n = -\frac{(\widetilde{z}_n, r(\widetilde{x}_n))}{(\widetilde{z}_n, M\widetilde{z}_n)} = -\frac{(r_n, z'_n) + (r'_n, z_n)}{2(z'_n, Az_n)}, \tag{5}$$

and the method becomes

$$x_{n+1} = x_n - \lambda_n z_n, \qquad x'_{n+1} = x'_n - \lambda_n z'_n,$$
$$r_{n+1} = r_n + \lambda_n A z_n, \qquad r'_{n+1} = r'_n + \lambda_n A^{\mathrm{T}} z'_n.$$

The fact that M is indefinite and, as explained above, that the sign of $J(y)-J(x)$ can depend on y explains why no monotony result can be proved in this case. Indeed, oscillations in the convergence behavior of the method are frequently observed as, e.g., in the biconjugate gradient algorithm [17] that will be discussed below.

Let us mention that the relations existing between the preceding methods can also be explained as a change in the norm used [23].

We shall now discuss several choices for the directions of descent.

2.1.1. Method of conjugate directions

As explained in [22, pp. 516–517], the gain of the method of steepest descent (i.e., when $z_n = r_n$) can be arbitrarily small if the condition number of the matrix M is large. In order to avoid this drawback, we shall now study another choice for the vectors z_n keeping in mind that z_n must not be orthogonal to r_n. We shall mostly follow [22, pp. 518ff]. We consider the $n+1$ vectors $z_0, \ldots, z_n$ and we shall look for $x_{n+1} = x_n - \lambda z_n$ solving the $(n+1)$-dimensional minimization problem

$$\min_{y \in \operatorname{span}(z_0, \ldots, z_n)} J(y).$$

Let $Z_n = [z_0, \ldots, z_{n-1}]$ be the matrix with columns $z_0, \ldots, z_{n-1}$. Thus, y has the form $y = Z_n w - \alpha z_n$ with $w \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, and we have

$$J(y) = J(Z_n w) - \alpha(Z_n w, Mz_n) + \tfrac{1}{2}\alpha^2(z_n, Mz_n) + \alpha(z_n, c).$$

Thus, the $(n+1)$-dimensional minimization problem is made more complicated by the term $\alpha(Z_n w, Mz_n)$. Without it, the problem splits into a minimization over the range of Z_n whose solution x_n is already known, and a minimization over the scalar α. Thus, we have in this case

$$\min_{y \in \operatorname{span}(z_0, \ldots, z_n)} J(y) = \min_{w} J(Z_n w) + \min_{\alpha \neq 0} \left[\tfrac{1}{2}\alpha^2(z_n, Mz_n) + \alpha(z_n, c)\right].$$

Thus, the best α is given by $\alpha_n = -(z_n, c)/(z_n, Mz_n)$. But, since $x_n \in \operatorname{span}(z_0, \ldots, z_{n-1})$, it follows that $x_n = Z_n w$ and $(z_n, Mx_n) = (z_n, MZ_n w) = (Mz_n, Z_n w) = 0$. Thus,

$$\alpha_n = -\frac{(z_n, r_n)}{(z_n, Mz_n)}$$

which shows that α_n coincides with λ_n given by (3).

The condition $\forall w \in \mathbb{R}^n, (Z_n w, Mz_n) = 0$ is equivalent to

$$(z_i, Mz_n) = 0 \quad \text{for } i = 0, \ldots, n-1$$

which shows that the vectors z_i must be M–conjugate. Since M is symmetric positive definite, it follows that $\forall i \neq j, (z_i, Mz_j) = 0$ and $\forall i, (z_i, Mz_i) > 0$ if $z_i \neq 0$.

We must now check if it is possible to find a vector z_n, M–conjugate to $z_0, \ldots, z_{n-1}$, such that $(z_n, Mz_n) \neq 0$ and $(z_n, r_n) \neq 0$. Assume that $z_0, \ldots, z_{n-1}$ are such that $\forall i \neq j, (z_i, Mz_j) = 0$. Since $x_n = Z_n w$

then $\forall z,(z,r_n)=(z,c)-(Mz,Z_nw)$. If, for all z M–conjugate to $z_0,\ldots,z_{n-1}$, we have $(z,r_n)=0$ it follows that $(z,c)=0$ which means that $c\in\text{span}(Mz_0,\ldots,Mz_{n-1})$. This condition implies that $M^{-1}c=x\in\text{span}(z_0,\ldots,z_{n-1})$ and thus, thanks to the minimization property of x_n over $\text{span}(z_0,\ldots,z_{n-1})$, we have $x_n=x$ and $r_n=0$. Hence, if $r_n\neq 0$, we can find a nonzero vector z_n (and thus $(z_n,Mz_n)\neq 0$) which is M–conjugate to $z_0,\ldots,z_{n-1}$ and satisfies $(z_n,r_n)\neq 0$. Moreover, since nonzero conjugate vectors are linearly independent, we have the

Property 1. *$\exists n\leqslant p$ (the dimension of the system) such that $x_n=x$ and $r_n=0$.*

Such a procedure is called *method of conjugate directions*, see [26, p. 108ff]. Let us study some of its properties in more detail. We have

Property 2. $\forall n>0,(z_i,r_n)=0$ for $i=0,\ldots,n-1$.

Proof. $(z_0,r_1)=(z_0,r_0)+\lambda_0(z_0,Mz_0)=0$ by definition of λ_0. Let us assume that $(z_i,r_n)=0$ for $i=0,\ldots,n-1$. We have

$$(z_i,r_{n+1})=(z_i,r_n)+\lambda_n(z_i,Mz_n).$$

For $i=0,\ldots,n-1$, the first term on the right-hand side is zero by the induction assumption and the second one also by the M–conjugacy of the vectors z_i. So, $(z_i,r_{n+1})=0$ for $i=0,\ldots,n-1$ and $(z_n,r_{n+1})=0$ by definition of λ_n. □

Let us now assume that $z_0=r_0$ and that $\forall n\geqslant 1, z_n-r_n\in\text{span}(r_0,\ldots,r_{n-1})$ and $z_n\neq 0$. Then, from Property 2, $(z_0,r_n)=0$ and thus $(r_0,r_n)=0$. Let us proceed by induction and assume that $(r_0,r_n)=\cdots=(r_i,r_n)=0$ for $i\leqslant n-2$. From Property 2, $(z_{i+1},r_n)=0=(a_0r_0+\cdots+a_ir_i+r_{i+1},r_n)=0$ and, by the induction assumption, it follows that $(r_{i+1},r_n)=0$. If $z_n=0$ then r_n is a linear combination of $r_0,\ldots,r_{n-1}$. Since $r_{i+1}=r_i+\lambda_iMz_i$, it follows that $\forall i, r_i-r_0\in\text{span}(Mz_0,\ldots,Mz_{i-1})$. So $z_n=0$ implies that Mz_{n-1} is a linear combination of $Mz_0,\ldots,Mz_{n-2}$. Thus, $(z_{n-1},Mz_{n-1})=0$ which is impossible since the condition $(z_{n-1},Mz_{n-1})\neq 0$ is required for computing r_n. Thus $z_n\neq 0$. Since $i\neq n$, one of the indexes is always greater than the other one and, so, we proved the

Property 3. *If $z_0=r_0$ and if, $\forall n\geqslant 1, z_n-r_n\in\text{span}(r_0,\ldots,r_{n-1})$, then $\forall n\geqslant 1$ and $\forall i\neq n,(r_i,r_n)=0$.*

We also have the

Property 4. *Under the assumptions of Property 3, we have $\forall n\geqslant 2,(Mz_i,r_n)=(Mz_i,Mz_n)=0$ for $i=0,\ldots,n-2$.*

Proof. $Mz_i=(r_{i+1}-r_i)/\lambda_i$. Thus, from Property 3, $(r_{i+1}-r_i,r_n)=(Mz_i,r_n)=0$ for $i=0,\ldots,n-2\geqslant 0$. Thus, it follows $(Mz_i,Mz_n)=(Mz_i,r_{n+1}-r_n)/\lambda_n=0$ for $i=0,\ldots,n-2$. □

It follows from Property 3 that $(z_n,r_n)=(r_n,r_n)$ since $(r_n,r_i)=0$ for $i=0,\ldots,n-1$. Thus

$$\lambda_n=-\frac{(r_n,r_n)}{(z_n,Mz_n)}.$$

This property also shows that the condition $(z_n, r_n) \neq 0$ is satisfied unless $r_n = 0$. We also have $\lambda_n < 0$.

2.1.2. Method of conjugate gradients

The method of conjugate directions contains the main ingredients of the method of conjugate gradients that will now be discussed. The vectors z_n can be computed by various recurrence relationships. For example, we can set

$$z_n = r_n + \beta_n z_{n-1}, \quad n = 1, 2, \ldots, \quad \text{with } z_0 = r_0$$

and choose β_n so that the conjugacy property is satisfied. We have

$$(Mz_i, z_n) = (Mz_i, r_n) + \beta_n (Mz_i, z_{n-1}). \tag{6}$$

For $n = 1$, we obtain

$$(Mz_0, z_1) = (Mz_0, r_1) + \beta_1 (Mz_0, z_0).$$

Thus, taking $\beta_1 = -(Mz_0, r_1)/(z_0, Mz_0)$, we have $(Mz_0, z_1) = 0$. Assuming that $(Mz_i, z_{n-1}) = 0$ for $i = 0, \ldots, n-2$ and using Property 4, the relation (6) shows that $(Mz_i, z_n) = 0$ for $i = 0, \ldots, n-2$. For $i = n-1$, we have

$$(Mz_{n-1}, z_n) = (Mz_{n-1}, r_n) + \beta_n (Mz_{n-1}, z_{n-1}).$$

Thus, $(Mz_{n-1}, z_n) = 0$ if we take

$$\beta_n = -\frac{(Mz_{n-1}, r_n)}{(z_{n-1}, Mz_{n-1})}. \tag{7}$$

Another expression for β_n can be obtained. We have

$$(Mz_{n-1}, r_n) = \frac{1}{\lambda_{n-1}}(r_n - r_{n-1}, r_n) = \frac{1}{\lambda_{n-1}}(r_n, r_n)$$

by Property 3. But $z_{n-1} = (z_n - r_n)/\beta_n$ and thus

$$(Mz_{n-1}, z_{n-1}) = \frac{1}{\beta_n}(Mz_{n-1}, z_n - r_n) = -\frac{1}{\beta_n}(Mz_{n-1}, r_n) = -\frac{1}{\beta_n \lambda_{n-1}}(r_n, r_n).$$

Replacing λ_{n-1} by its expression, it follows

$$\beta_n = \frac{(r_n, r_n)}{(r_{n-1}, r_{n-1})}.$$

Thus, we finally obtain the following algorithm

$$z_0 = r_0 = c - Mx_0,$$

$$x_{n+1} = x_n - \lambda_n z_n \qquad \text{with} \quad \lambda_n = -(r_n, r_n)/(z_n, Mz_n),$$

$$r_{n+1} = r_n + \lambda_n Mz_n,$$

$$z_{n+1} = r_{n+1} + \beta_{n+1} z_n \quad \text{with} \quad \beta_{n+1} = (r_{n+1}, r_{n+1})/(r_n, r_n).$$

This method is the conjugate gradient algorithm [27]; see [26] for a detailed discussion. If A is symmetric and positive definite, we can take $M=A$ and $c=b$.

Let us apply the conjugate gradient algorithm to the normal residuals. We have $M=A^{\mathrm{T}}A, r'_n = A^{\mathrm{T}}b-A^{\mathrm{T}}Ax_n=A^{\mathrm{T}}r_n$ with $r_n=b-Ax_n$. Thus, we obtain the following algorithm starting from $z_0=A^{\mathrm{T}}r_0$:

$$
\begin{aligned}
&x_{n+1} = x_n - \lambda_n z_n \qquad \text{with} \quad \lambda_n = -(A^{\mathrm{T}}r_n, A^{\mathrm{T}}r_n)/(Az_n, Az_n),\\
&r_{n+1} = r_n + \lambda_n A z_n,\\
&z_{n+1} = A^{\mathrm{T}}r_{n+1} + \beta_{n+1} z_n \quad \text{with} \quad \beta_{n+1} = (A^{\mathrm{T}}r_{n+1}, A^{\mathrm{T}}r_{n+1})/(A^{\mathrm{T}}r_n, A^{\mathrm{T}}r_n).
\end{aligned}
$$

This algorithm, which is now called CGNR, was proposed in [27]; see also [16, p. 403]. Each iteration needs the computation of two matrix-by-vector products. It is mathematically equivalent to the LSQR [39] which possesses more favorable numerical properties.

Let us now apply the conjugate gradient algorithm to the normal equations $AA^{\mathrm{T}}x'=b$ with $x=A^{\mathrm{T}}x'$. Setting $x_n = A^{\mathrm{T}}x'_n$, $r_n = b - Ax_n = b - AA^{\mathrm{T}}x'_n = r'_n$ and $p_n = A^{\mathrm{T}}z_n$, we obtain, starting from $p_0 = A^T r_0$

$$
\begin{aligned}
&x_{n+1} = x_n - \lambda_n p_n \qquad \text{with} \quad \lambda_n = -(r_n, r_n)/(p_n, p_n),\\
&r_{n+1} = r_n + \lambda_n A p_n,\\
&p_{n+1} = A^{\mathrm{T}}r_{n+1} + \beta_{n+1} p_n \quad \text{with} \quad \beta_{n+1} = (r_{n+1}, r_{n+1})/(r_n, r_n).
\end{aligned}
$$

This algorithm, which is called CGNE, is essentially due to Craig [13]. It can be found in that exact form in [16, p. 405]. Each iteration needs two matrix–by–vector products.

These two algorithms can never break down. However, they can suffer from near–breakdowns. Their convergence factor now depends on the condition number of the matrix AA^{T} instead of that of A. However, there are particular cases where solving the normal equations or the normal residuals is optimal; see [18] for a review and the corresponding references.

Let us now apply the conjugate gradient algorithm to the expanded system

$$
\begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} x' \\ x \end{pmatrix} = \begin{pmatrix} b \\ b' \end{pmatrix}.
$$

Since the dimension of this system is $2p$, the number of iterations will be roughly doubled. In order to avoid this drawback, let us have a closer look at the conjugate gradient algorithm. As before, we set

$$
\tilde{x}_n = \begin{pmatrix} x'_n \\ x_n \end{pmatrix}, \quad \tilde{z}_n = \begin{pmatrix} z'_n \\ z_n \end{pmatrix}, \quad \tilde{r}_n = \begin{pmatrix} r'_n = b' - A^{\mathrm{T}}x'_n \\ r_n = b - Ax_n \end{pmatrix}, \quad r(\tilde{x}_n) = c - M\tilde{x}_n = \begin{pmatrix} r_n \\ r'_n \end{pmatrix}.
$$

Again, it follows that $(\tilde{z}_n, M\tilde{z}_n) = 2(z'_n, Az_n)$. Moreover, $\tilde{r}_n$ must be replaced by $r(\tilde{x}_n)$ in the relations (1) and (2), and (5) holds. In the proof of Property 2, $\tilde{r}_n$ has to be replaced by $r(\tilde{x}_n)$ and thus we now have

$$
(z'_i, r_n) + (z_i, r'_n) = 0 \quad \text{for} \quad i = 0, \ldots, n-1.
$$

Similarly, the conjugacy property becomes

$$
(z'_i, Az_n) + (z_i, A^{\mathrm{T}}z'_n) = 0 \qquad \text{for} \quad i = 0, \ldots, n-1.
$$

Property 3 is now $(\widetilde{r}_i, r(\widetilde{x}_n)) = 0$ for $i = 0, \ldots, n-1$, i.e.,

$$(r'_i, r_n) + (r_i, r'_n) = 0 \quad \text{for} \quad i = 0, \ldots, n-1.$$

Similarly, Property 4 becomes

$$(z'_i, Ar_n) + (z_i, A^{\mathrm{T}} r'_n) = 0 \quad \text{for} \quad i = 0, \ldots, n-2.$$

It can be proved by induction (see [7] for details) that it holds

Property 5. $(Az_i, z'_j) = (z_i, r'_j) = (z'_i, r_j) = (r'_i, r_j) = 0, \forall i \neq j$ *and* $(Ar_n, z'_i) = (r'_n, Az_i) = 0$, *for* $i = 0, \ldots, n-2 \geqslant 0$. *Moreover,* $\exists n \leqslant p$ *such that* $x_n = x, r_n = 0$ *and* $x'_n = x', r'_n = 0$.

Families of vectors satisfying orthogonality relations for all $i \neq j$ are said to be biorthogonal with respect to the corresponding inner product [4].

Let us give other expressions for λ_n and β_{n+1}. We have

$$(Az_n, r'_{n+1}) = \frac{1}{\lambda_n}(r_{n+1} - r_n, r'_{n+1}) = \frac{1}{\lambda_n}(r_{n+1}, r'_{n+1}).$$

But $z'_n = (z'_{n+1} - r'_{n+1})/\beta_{n+1}$ and thus

$$(Az_n, z'_n) = \frac{1}{\beta_{n+1}}(Az_n, z'_{n+1} - r'_{n+1}) = -\frac{1}{\beta_{n+1}}(Az_n, r'_{n+1}) = -\frac{1}{\lambda_n \beta_{n+1}}(r_{n+1}, r'_{n+1}).$$

Replacing λ_n by its expression gives $\beta_{n+1} = (r_{n+1}, r'_{n+1})/(r_n, r'_n)$ since $(z'_n, r_n) = (r'_n, r_n)$. Using this relation into the expression of λ_n gives $\lambda_n = -(r'_n, r_n)/(z'_n, Az'_n)$.

Thus, the conjugate gradient algorithm applied to the expanded system leads to the following algorithm:

$$\begin{aligned}
&z_0 = r_0 = b - Ax_0, && z'_0 = r'_0,\\
&x_{n+1} = x_n - \lambda_n z_n, && \lambda_n = -(r'_n, r_n)/(z'_n, Az_n),\\
&r_{n+1} = r_n + \lambda_n Az_n, && r'_{n+1} = r'_n + \lambda_n A^{\mathrm{T}} z'_n,\\
&z_{n+1} = r_{n+1} + \beta_{n+1} z_n, && \beta_{n+1} = (r_{n+1}, r'_{n+1})/(r_n, r'_n),\\
&z'_{n+1} = r'_{n+1} + \beta_{n+1} z'_n.
\end{aligned}$$

This algorithm is the biconjugate gradient algorithm (BCG) due to Fletcher [17]. It is also called Lanczos/Orthomin [47]; see also [30]. It is essentially identical to the complete algorithm for minimized iterations given in [35]. The vectors computed by Lanczos' algorithm are related to ours by $r_n = a_n p_n, r'_n = a_n p^*_n, z_n = a_n q_n, z'_n = a_n q^*_n$ where a_n is such that $a_{n+1}/a_n = -(r_n, r'_n)/(z'_n, Az_n)$.

It must be noticed that, unless the solution x' of the system $A^{\mathrm{T}} x' = b'$ is needed, the vector r'_0 can be arbitrarily chosen except that it must not be orthogonal to r_0. If the solution of this system is also wanted, then we must choose x'_0, set $r'_0 = b' - A^{\mathrm{T}} x'_0$ and perform the iterations $x'_{n+1} = x'_n - \lambda_n z'_n$.

An algorithm, inspired by the BCG and where z_{n+1} is computed from z_n and z'_n and where z'_{n+1} is obtained from z_{n+1} and z'_n, is given in [2].

The vectors z_n appearing in the conjugate gradient algorithm can also be computed by other recurrence relationships. We have $z_n = r_n + \beta_n z_{n-1}$. But $r_n = r_{n+1} - \lambda_n M z_n$. Thus, $z_n = r_{n+1} - \lambda_n M z_n + \beta_n z_{n-1}$. Replacing r_{n+1} by $z_{n+1} - \beta_{n+1} z_n$ leads to

$$z_{n+1} = (1 + \beta_{n+1}) z_n + \lambda_n M z_n - \beta_n z_{n-1}. \tag{8}$$

It is possible to give expressions for β_n and λ_n which do not involve the residual vectors. Let us first remark that if z_n is replaced by $c_n z_n$ where c_n is an arbitrary nonzero number, then the formulae $x_{n+1} = x_n - \lambda_n z_n$ and $r_{n+1} = r_n + \lambda_n M z_n$ remain unchanged if λ_n is computed by (3). The same is true for the relation (8) if β_n is computed by (7). Thus, in these two relations, z_n can be replaced by $c_n z_n$ which means that the relation (8) can be replaced by the simpler one

$$z_{n+1} = M z_n - a_{n+1} z_n - b_{n+1} z_{n-1}.$$

We have

$$(z_i, M z_{n+1}) = (M z_i, M z_n) - a_{n+1}(z_i, M z_n) - b_{n+1}(z_i, M z_{n-1}).$$

The second term on the right-hand side is zero for $i = 0, \ldots, n-1$ and the third one for $i = 0, \ldots, n-2$ by the conjugacy property. The first term is zero for $i = 0, \ldots, n-2$ by the symmetry of M and Property 4. Thus, $(z_i, M z_{n+1}) = 0$ for $i = 0, \ldots, n-2$. Imposing also this quantity to be zero for $i = n-1$ and $i = n$ leads to

$$a_{n+1} = (M z_n, M z_n)/(z_n, M z_n)$$
$$b_{n+1} = (M z_{n-1}, M z_n)/(z_{n-1}, M z_{n-1}).$$

If A is symmetric and positive definite, these relations can be used with $M = A$ and $c = b$. If not, they can be used on the normal equations or on the normal residuals. In both cases, they lead to the same recurrence relationship

$$z_{n+1} = (1 + \beta_{n+1}) z_n + \lambda_n A^{\mathrm{T}} A z_n - \beta_n z_{n-1},$$

where λ_n and β_n are given by their expressions in the CGNR or in the CGNE, respectively (in this last case, z_n has to be replaced by p_n). When applied to the expanded system, this algorithm is known as Lanczos/Orthodir [47]; see also [30].

The auxiliary vectors z_n can also be eliminated from the whole process and we obtain the recurrence relationships

$$r_{n+1} = (1 + \gamma_n) r_n + \lambda_n M r_n - \gamma_n r_{n-1}$$
$$x_{n+1} = (1 + \gamma_n) x_n - \lambda_n r_n - \gamma_n x_{n-1}$$

with $\gamma_n = \lambda_n \beta_n / \lambda_{n-1}$. Indeed $r_{n+1} = r_n + \lambda_n M z_n$. But $M z_n = M r_n + \beta_n M z_{n-1}$ and thus $r_{n+1} = r_n + \lambda_n M r_n + \lambda_n \beta_n M z_{n-1}$. Now, $M z_{n-1} = (r_n - r_{n-1})/\lambda_{n-1}$ and it follows $r_{n+1} = (1 + \lambda_n \beta_n / \lambda_{n-1}) r_n + \lambda_n M r_n - \lambda_n \beta_n / \lambda_{n-1} r_{n-1}$ which is the preceding relation. In this algorithm, the direction vectors z_n are no longer needed. Thus, since the expression of λ_n still involved them, it has to be modified. First, we write the relation (9) as

$$r_{n+1} = \lambda_n (M r_n + \alpha_n r_n + \tau_n r_{n-1}). \tag{9}$$

Using Property 3, we have

$$(r_{n-1}, r_{n+1}) = \lambda_n [(r_{n-1}, Mr_n) + \tau_n (r_{n-1}, r_{n-1})] = 0,$$
$$(r_n, r_{n+1}) = \lambda_n [(r_n, Mr_n) + \alpha_n (r_n, r_n)] = 0.$$

These two relations give α_n and τ_n. Moreover, from (9), we have $\lambda_n(\alpha_n + \tau_n) = 1$ and, thus, the three coefficients of (9) are obtained without using the direction vectors. If A is symmetric and positive definite, these relations can be used with $M = A$ and $c = b$. If not, they can be used on the normal equations or on the normal residuals. In both cases, they lead to the same recurrence relationship

$$r_{n+1} = \lambda_n \left(AA^{\mathrm{T}} r_n + \alpha_n r_n + \tau_n r_{n-1}\right),$$

where λ_n, α_n and τ_n are computed as above with $M = A^{\mathrm{T}}A$ in the case of the normal residuals and $M = A^{\mathrm{T}}A$ for the normal equations. When applied to the expanded system, this algorithm is known as Lanczos/Orthores [47]; see also [30].

Other related algorithms can be found in [26, 25]. A derivation of all the possible algorithms, based on orthogonal polynomials, is given in [9]. These algorithms are discussed in detail in [3]. Complex systems can be treated similarly; see, e.g., [29].

2.2. *Projection acceleration procedures*

Let now (x_n) be a sequence of iterates obtained by an arbitrary iterative method and let (z_n) be a sequence of arbitrary nonzero vectors. We shall define a new sequence (y_n) of iterates by

$$y_n = x_n - \lambda_n z_n,$$
$$\rho_n = r_n + \lambda_n M z_n,$$

with

$$\lambda_n = -\frac{(z_n, r_n)}{(z_n, Mz_n)}$$

where $r_n = c - Mx_n$ and $\rho_n = c - My_n$.

The choice $z_n = -\nabla J(x_n) = r_n$ can be considered as a steepest descent acceleration while a choice of z_n such that $\forall n \geqslant 1, (z_n, Mz_i) = 0$ for $i = 0, \ldots, n-1$ is a conjugate gradient acceleration.

Let us now have a look at the various preceding possibilities.

(i) *Symmetric positive definite case*: We have

$$y_n = x_n - \lambda_n z_n \quad \text{with} \quad \lambda_n = -\frac{(z_n, r_n)}{(z_n, Az_n)}$$

and $r_n = b - Ax_n$. As explained in Section 2, this choice minimizes the norm $(y_n - x, A(y_n - x))$ and we have

$$\|y_n - x\|_A^2 = \|x_n - x\|_A^2 - \frac{(z_n, r_n)^2}{(z_n, Az_n)},$$

where $\|u\|_A^2 = (u, Au)$. Thus $\|y_n - x\|_A \leqslant \|x_n - x\|_A$. When $z_n = r_n$, this procedure can be considered as a steepest descent acceleration.

(ii) *Normal residuals*: With $M = A^{\mathrm{T}}A$ and $c = A^{\mathrm{T}}b$, we have

$$y_n = x_n - \lambda_n z_n \quad \text{with} \quad \lambda_n = -\frac{(z_n, A^{\mathrm{T}}b - A^{\mathrm{T}}Ax_n)}{(z_n, A^{\mathrm{T}}Az_n)} = -\frac{(Az_n, r_n)}{(Az_n, Az_n)}$$

and $r_n = b - Ax_n$. This choice of λ_n minimizes (ρ_n, ρ_n) where now $\rho_n = b - Ay_n$. We have

$$\rho_n = r_n + \lambda_n Az_n,$$
$$\|\rho_n\|^2 = \|r_n\|^2 - \frac{(r_n, Az_n)^2}{(Az_n, Az_n)} = \|r_n\|^2 \sin^2 \theta_n,$$

where θ_n is the angle between the vectors r_n and Az_n. Thus, obviously, $\|\rho_n\| \leqslant \|r_n\|$. Defining x'_n by $x'_n = x_n - z_n$, this projection acceleration procedure is equivalent to applying the hybrid procedure introduced in [8] to the sequences (x_n) and (x'_n). Acceleration results can be found in [1]. In particular, we have the

Property 6. *If* $\exists \theta \neq \pi/2$ *such that* $\lim_{n\to\infty} \theta_n = \theta$, *then* $\lim_{n\to\infty} \|\rho_n\|/\|r_n\| = |\sin\theta| < 1$. *A necessary and sufficient condition that* $\lim_{n\to\infty} \|\rho_n\|/\|r_n\| = 0$ *is that* (θ_n) *tends to* 0 *or* π *when* n *tends to infinity.*

When $z_n = r_n$, we obtain the Richardson acceleration procedure which is equivalent to applying the steepest descent acceleration to the normal residuals. An optimal choice (in some sense) for z_n is given in [6] where acceleration results can also be found.

Extensions of the hybrid procedure are discussed in [5].

(iii) *Normal equations*: We have $M = AA^{\mathrm{T}}$ and $c = b$. Since u is now an approximation of the solution x' of the system $AA^{\mathrm{T}}x' = b$, we shall assume that our iterative method produces iterates (x'_n) that are approximations of x'. We shall denote by y'_n the corresponding iterates of the projection acceleration procedure. We have $y'_n = x'_n - \lambda_n z_n$ with

$$\lambda_n = -\frac{(z_n, r'_n)}{(z_n, Mz_n)} = -\frac{(z_n, r'_n)}{(A^{\mathrm{T}}z_n, A^{\mathrm{T}}z_n)}$$

and $r'_n = b - AA^{\mathrm{T}}x'_n$. Multiplying both sides by A^{T} and setting $x_n = A^{\mathrm{T}}x'_n$, $y_n = A^{\mathrm{T}}y'_n$ and $r_n = b - Ax_n = b - AA^{\mathrm{T}}x'_n = r'_n$, we obtain

$$y_n = x_n - \lambda_n A^{\mathrm{T}}z_n \quad \text{with} \quad \lambda_n = -\frac{(z_n, r_n)}{(A^{\mathrm{T}}z_n, A^{\mathrm{T}}z_n)}.$$

This choice minimizes $(y_n - x, y_n - x)$. We have

$$\|y_n - x\|^2 = \|x_n - x\|^2 - \frac{(z_n, r_n)^2}{(A^{\mathrm{T}}z_n, A^{\mathrm{T}}z_n)}.$$

Since $(z_n, r_n) = (A^{\mathrm{T}}z_n, x - x_n)$, we have

$$\|y_n - x\|^2 = \|x_n - x\|^2 \sin^2 \varphi_n$$

where φ_n is the angle between the vectors $A^T z_n$ and $x - x_n$ and, thus, $\|y_n - x\| \leqslant \|x_n - x\|$. We obviously have the

Property 7. *If* $\exists \varphi \neq \pi/2$ *such that* $\lim_{n\to\infty} \varphi_n = \varphi$, *then* $\lim_{n\to\infty} \|y_n - x\|/\|x_n - x\| = |\sin\varphi| < 1$. *A necessary and sufficient condition that* $\lim_{n\to\infty} \|y_n - x\|/\|x_n - x\| = 0$ *is that* (φ_n) *tends to* 0 *or* π *when* n *tends to infinity.*

For the choice $z_n = r_n$, this procedure can be considered as an extension of the steepest descent acceleration to an arbitrary matrix.

Defining x'_n by $x'_n = x_n - z_n$, the projection acceleration procedure is equivalent to applying the hybrid procedure to the sequences (x_n) and (x'_n) but with a choice of λ_n minimizing the error $y_n - x$ instead of the residual ρ_n as proposed in [8]. However, it must be noticed that such a procedure needs using A^T.

(iv) *Expanded system*: As in the case of the projection iterative method applied to the expanded system, we still have

$$\lambda_n = -\frac{(r_n, z'_n) + (r'_n, z_n)}{2(z'_n, Az_n)}$$

and the projection acceleration procedure becomes

$$y_n = x_n - \lambda_n z_n, \quad y'_n = x'_n - \lambda_n z'_n,$$
$$\rho_n = r_n + \lambda_n A z_n, \quad \rho'_n = r'_n + \lambda_n A^T z'_n.$$

It must be noticed that these acceleration procedures need an additional matrix–by–vector product at each iteration. Such a remark must be taken into account when estimating the gain.

Acknowledgements

This work was supported by The Human Capital and Mobility Programme of the European Community, Project ROLLS, Contract CHRX-CT93-0416 (DG 12 COMA).

References

[1] A. Abkowicz and C. Brezinski, Acceleration properties of the hybrid procedure for solving linear systems, *Appl. Math.*, **23** (1996) 417–432.
[2] M. Amara and J.-C. Nedelec, Résolution de système matriciel indéfini par une décomposition sur une double suite orthogonale, *C.R. Acad. Sci. Paris Sér. I* **295** (1982) 309–312.
[3] C. Baheux, New implementations of the Lanczos method, *J. Comput. Appl. Math.* **57** (1995) 3–15.
[4] C. Brezinski, *Biorthogonality and its Applications to Numerical Analysis* (Marcel Dekker, New York, 1992).
[5] C. Brezinski, Hybrid procedures and semi-iterative methods for linear systems, submitted.
[6] C. Brezinski, Variations on Richardson's method and acceleration, *Bull. Soc. Math. Belg.*, to appear.
[7] C. Brezinski, *Projection methods for Systems of Equations* (North-Holland, Amsterdam), to appear.
[8] C. Brezinski and M. Redivo Zaglia, Hybrid procedures for solving systems of linear equations, *Numer. Math.* **67** (1994) 1–19.

[9] C. Brezinski and H. Sadok, Lanczos-type algorithms for solving systems of linear equations, *Appl. Numer. Math.* **11** (1993) 443–473.
[10] A. Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, *C.R. Acad. Sci. Paris* **25** (1847) 536–538.
[11] P.G. Ciarlet, *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation* (Masson, Paris, 1982).
[12] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, *Ricerca Sci.* II **9** (1938) 326–333.
[13] E.J. Craig, The N-step iteration procedures, *J. Math. Phys.* **34** (1955) 64–73.
[14] A. Dax, Line search acceleration of iterative methods, *Linear Algebra Appl.* **130** (1990) 43–63.
[15] H.C. Elman, Iterative methods for large, sparse, nonsymmetric systems of linear equations, Ph.D. Thesis, Yale University, 1982.
[16] D.K. Faddeev and V.N. Fadeeva, *Computational Methods of Linear Algebra* (W.H. Freeman and Co., San Francisco, 1963).
[17] R. Fletcher, Conjugate gradient methods for indefinite systems, in: G.A. Watson, Ed., *Numerical Analysis, Dundee* 1975, Lecture Notes in Maths vol. **506** (Springer, Berlin, 1976) 73–89.
[18] R.W. Freund, G.H. Golub and N.M. Nachtigal, Iterative solution of linear systems, in: A. Iserles, Ed., *Acta Numerica*, (Cambridge Univ. Press, Cambridge, 1992) 57–100.
[19] N. Gastinel, Procédé itératif pour la résolution numérique d'un système d'équations linéaires, *C.R. Acad. Sci. Paris* **246** (1958) 2571–2574.
[20] N. Gastinel, Sur-décomposition de normes générales et procédés itératifs, *Numer. Math.* **5** (1963) 142–151.
[21] N. Gastinel, *Analyse Numérique Linéaire* (Hermann, Paris, 1966).
[22] G.H. Golub and C.F. Van Loan, *Matrix Computations* (The Johns Hopkins Univ. Press, Baltimore, MD, 2nd ed., 1989).
[23] M.H. Gutknecht, Changing the norm in conjugate gradient type algorithms, *SIAM J. Numer. Anal.* **30** (1993) 40–56.
[24] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations* (Springer, New York, 1994).
[25] M.R. Hestenes, The conjugate-gradient method for solving linear systems, in: J. Curtiss, Ed., *Proc. 6th Symp. in Applied Mathematics* (American Mathematical Society, Providence, RI, 1956) 83–102.
[26] M.R. Hestenes, *Conjugate Direction Methods in Optimization* (Springer, Berlin, 1980).
[27] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Stand.* **49** (1952) 409–436.
[28] A.S. Householder, *The Theory of Matrices in Numerical Analysis* (Blaisdell, New York, 1964).
[29] P. Joly and G. Meurant, Complex conjugate gradient methods, *Numer. Algorithms* **4** (1993) 379–406.
[30] W. Joubert, Generalized conjugate gradient and Lanczos methods for the solution of nonsymmetric systems of linear equations, Ph.D. Thesis, The University of Texas at Austin, 1990.
[31] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, *Bull. Acad. Polon. Sci.* **A35** (1937) 355–357.
[32] D.R. Kincaid, On complex second-degree iterative methods, *SIAM J. Numer. Anal.* **11** (1974) 211–218.
[33] D.R. Kincaid, Stationary second-degree iterative methods, *Appl. Numer. Math.* **16** (1994) 227–237.
[34] D.R. Kincaid and D.M. Young, Stationary second-degree iterative methods and recurrences, in: R. Beauwens and P. de Groen, Eds., *Iterative Methods in Linear Algebra* (North-Holland, Amsterdam, 1992) 27–47.
[35] C. Lanczos, Solution of systems of linear equations by minimized iterations, *J. Res. Natl. Bur. Stand.* **49** (1952) 33–53.
[36] V.I. Lebedev and V.V. Zabelin, Combined trinomial iterative methods with Chebyshev parameters, *East–West J. Numer. Math.* **3** (1995) 145–162.
[37] P. Linz, *Theoretical Numerical Analysis. An Introduction to Advanced Techniques* (Wiley, New York, 1979).
[38] N.M. Nachtigal, S.C. Reddy and L.N. Trefethen, How fast are nonsymmetric matrix iterations? *SIAM J. Matrix Anal. Appl.* **13** (1992) 778–795.
[39] 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.
[40] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations* (Springer, Berlin, 1994).
[41] L.F. Richardson, The approximate arithmetical solution by finite differences of physical problems involving differential equations, with application to the stress in a masonry dam, *Philos. Trans. Roy. Soc. London Ser. A* **226** (1910) 307–357.

[42] W. Schönauer, *Scientific Computing on Vector Computers* (North-Holland, Amsterdam, 1987).
[43] W. Schönauer, H. Müller and E. Schnepf, Numerical tests with biconjugate gradient type methods, *Z. Angew. Math. Mech.* **65** (1985) T400–T402.
[44] R.V. Southwell, Stress–calculation in frameworks by the method of "systematic relaxation of constraints", *Proc. Roy. Soc. London* A **151** (1935) 56–95; **153** (1935) 41–76.
[45] G. Temple, The general theory of relaxation methods applied to linear systems, *Proc. Roy. Soc. London* A **169** (1939) 476–500.
[46] D.M. Young, A historical review of iterative methods, in: S.G. Nash, Ed., *A History of Scientific Computing* (Addison-Wesley, Reading, MA, 1990) 180–194.
[47] D.M. Young and K.C. Jea, Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods, *Linear Algebra Appl.* **34** (1980) 159–194.