Numer. Math. 67: 1–19 (1994)

Numerische
Mathematik

Hybrid procedures for solving linear systems

C. Brezinski[1] **and M. Redivo-Zaglia**[2]

[1] Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA - M3, Université des Sciences et Technologies de Lille, F-59655 Villeneuve d'Ascq Cedex, France
[2] Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, via Gradenigo 6/a, I-35131 Padova, Italy

Received October 21, 1992/Revised version received May 28, 1993

Summary. In this paper, we introduce the notion of hybrid procedures for solving a system of linear equations. A hybrid procedure consists in a combination of two *arbitrary* approximate solutions with coefficients summing up to one. Thus the combination only depends on one parameter whose value is chosen in order to minimize the Euclidean norm of the residual vector obtained by the hybrid procedure. Properties of such procedures are studied in detail. The two approximate solutions which are combined in a hybrid procedure are usually obtained by two iterative methods. Several strategies for combining these two methods together or with the previous iterate of the hybrid procedure itself are discussed and their properties are analyzed. Numerical experiments illustrate the various procedures.

Mathematics Subject Classification (1991): 65F05, 65B05

1. Introduction

Let us try to explain how we got the idea of the hybrid procedures described in this paper.

Lanczos method [23] (usually implemented via the well known biconjugate gradient algorithm [12]) and the conjugate gradient squared method (CGS) of Sonneveld [30] are two iterative methods with finite convergence, for solving systems of linear equations, which recently received much attention. It was remarked, from the numerical experiments, that, when Lanczos method does not work well, then the CGS is usually worse and that, when Lanczos method works well, then the CGS is still better. Thus we wanted to use both methods simultaneously and, at each iteration, to select the best one according to the norm of their respective residual vectors. But we also did something more: after selecting, at each iteration, the best method, we continued the iterations of both methods from the last residual vector given by the best one. The numerical results were quite interesting! So, we had to try to understand the theory of such a procedure, i.e., writing the iterates of the new method in terms of the orthogonal polynomials which are involved in Lanczos method and the CGS. However,

Correspondence to: C. Brezinski

this seemed to be quite difficult. Thus, the idea emerged of taking a combination of both methods instead of mixing them as we did. This was the starting point of the hybrid procedures discussed in this paper. As we shall see in the next sections, these procedures turn out to be more general than we thought at the beginning.

Let x' and x'' be two approximations of the solution x of a system of linear equations. We shall construct a new approximation y of x by

$$y = \alpha x' + (1 - \alpha)x''$$

where the parameter α is chosen to minimize the Euclidean norm of the residual vector $r = b - Ay$. It will be proved that this residual is not greater than the residuals corresponding to the vectors x' and x''. Such a procedure, called a hybrid procedure since it mixes two different approximations of the solution x, will be studied in details in Sect. 2. Usually, x' and x'' are obtained by two iterative methods for solving a system of linear equations. In Sect. 3, several possible strategies for combining these two underlying iterative methods (or one of them with the previous iterate of the hybrid procedure) will be discussed. Numerical examples will be given in Sect. 4.

2. The procedure

Let us consider the system of linear equations

$$Ax = b.$$

We assume that two approximate solutions x' and x'' are known and we shall construct a better approximate solution y from x' and x''. We shall set

$$r' = b - Ax'$$
$$r'' = b - Ax''$$

and we shall compute

$$r = b - Ay$$

as a combination of r' and r'', that is

$$r = \alpha r' + \beta r''.$$

Once r has been obtained, we need to compute y. We have

$$b - Ay = \alpha(b - Ax') + \beta(b - Ax'').$$

Since y have to be computed without using A^{-1}, b must be eliminated from the preceding relation, that is we must take

$$\beta = 1 - \alpha$$

and thus

$$r = \alpha r' + (1 - \alpha)r''.$$

Because of the simple dependence on α in the above equation, the best possible α, which minimizes (r, r), is easily seen to be given by

$$\alpha = -(r' - r'', r'')/(r' - r'', r' - r''). \tag{1}$$

Of course, if $r' = r''$ we shall take $r = r'$.

Thus, we have the following hybrid procedure

$$r = \frac{(r' - r'', r')r'' - (r' - r'', r'')r'}{(r' - r'', r' - r'')}, \tag{2}$$

which can also be written as

$$r = r' - \frac{(r' - r'', r')}{(r' - r'', r' - r'')}(r' - r''), \tag{3}$$

or as

$$r = r'' - \frac{(r' - r'', r'')}{(r' - r'', r' - r'')}(r' - r''). \tag{4}$$

From (2) we have

$$y = \frac{(r' - r'', r')x'' - (r' - r'', r'')x'}{(r' - r'', r' - r'')}, \tag{5}$$

with similar expressions holding for (3) and (4). From (3) and (4), it is easy to see that $r = 0$ if $r' = 0$ or if $r'' = 0$.

Also, from the Cauchy-Schwarz inequality, we see that

$$(r, r) = \frac{(r', r')(r'', r'') - (r', r'')^2}{(r' - r'', r' - r'')} \geq 0. \tag{6}$$

Moreover, for the minimizing α given by (1), it is evident that

$$(r, r) \leq \min[(r', r'), (r'', r'')].$$

The vectors r' and r'' are of course fundamental to our hybrid procedure. But, some choices of r' and r'' are more suitable than others, for improved behavior of the procedure. Let us now discuss this point. We have

$$(r, r) = (r', r') - \frac{(r' - r'', r')^2}{(r' - r'', r' - r'')}, \tag{7}$$

and

$$(r, r) = (r'', r'') - \frac{(r' - r'', r'')^2}{(r' - r'', r' - r'')}. \tag{8}$$

These two relations show that there is no improvement in the hybrid procedure when $r' - r''$ is orthogonal to r' or r'', i.e., in other words, when (r', r'') is equal either to (r', r') or (r'', r''). Obviously, this is avoided if r' and r'' can be chosen so that

$$(r', r'') \leq 0,$$

i.e., if the angle between r' and r'' is greater or equal to $\pi/2$.

In particular when $(r', r'') = 0$, we have

$$r = \frac{(r', r')r'' + (r'', r'')r'}{(r', r') + (r'', r'')}$$

and

$$(r,r) = \frac{(r',r')}{1+(r',r')/(r'',r'')} = \frac{(r'',r'')}{1+(r'',r'')/(r',r')}.$$

Setting $\rho = (r',r')/(r'',r'')$, we have $(r,r)/(r',r') = 1/(1+\rho)$ and $(r,r)/(r'',r'') = \rho/(1+\rho)$ which shows that, if ρ is close to zero, (r,r) is almost equal to (r',r') but much smaller than (r'',r'') and that, if ρ is large, (r,r) is almost equal to (r'',r'') while it is much smaller than (r',r'). Thus the gain is optimal with respect to *both* r' and r'' when $\rho = 1$ that is when $(r',r') = (r'',r'')$. These remarks and formulae will be useful in the next section.

Let us now give some more geometrical considerations about our hybrid procedure. Multiplying scalarly (3) and (4) by r' and r'' respectively and comparing to (7) and (8) shows that

$$(r,r) = (r,r') = (r,r'')$$

or, in other words

$$(r,r-r') = (r,r-r'') = (r,r'-r'') = 0.$$

This means that r is the height through the origin of the triangle with sides r' and r'' as shown in the following figure

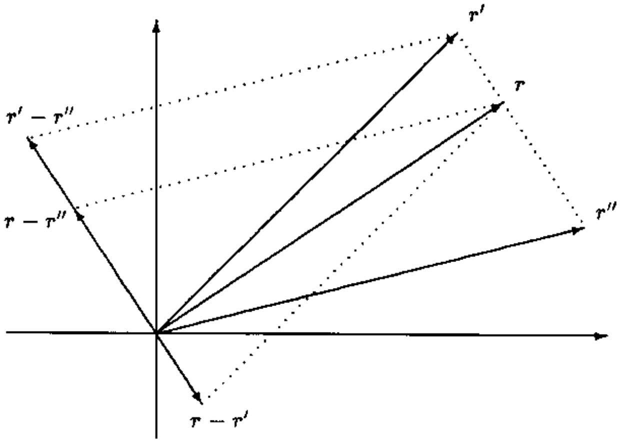

Fig. 1.

From geometrical considerations or from Schwarz inequality, we see that we shall obtain $r = 0$ if and only if $\exists a \neq 1$ such that $r'' = ar'$, that is

$$x'' = ax' + (1-a)x.$$

This choice is optimal but it is unfeasible because x is not known. Thus, we could replace x in the preceding formula by an approximation u, that is we could take

$$x'' = ax' + (1-a)u.$$

Setting $z = b - Au$, we have $r'' = ar' + (1-a)z$. Now, the question is: does it exists a best choice for a? The answer is *no* since, replacing r'' by this expression in (6) gives

$$(r, r) = \frac{(r', r')(z, z) - (r', z)^2}{(r' - z, r' - z)}$$

which shows that (r, r) is independent of a. Thus we can take $a = 0$ which leads to $x'' = u$ and $r'' = z$ and so the introduction of a, u and z was not necessary and we shall no more use them. However, we must remember that the more collinear r' and r'' are, the smaller is (r, r).

Remarks. It must be noticed that

$$r = \begin{vmatrix} r' & r'' \\ (r' - r'', r') & (r' - r'', r'') \end{vmatrix} / \begin{vmatrix} 1 & 1 \\ (r' - r'', r') & (r' - r'', r'') \end{vmatrix}$$

and a similar formula for y by replacing r' and r'' in the first row of the numerator by x' and x'' respectively.

Such a hybrid procedure must be compared to the composite sequence transformations described in [6] where two sequence transformations are composed by a quite similar formula. The property that $r = 0$ if either $r' = 0$ or $r'' = 0$ corresponds to the property that the kernel of a composite transformation of rank 2 contains the kernels of the two underlying transformations involved in its construction.

Our hybrid procedure can also be considered as a generalization of the acceleration scheme, proposed by Gearhart and Koshy [19] and called the *closest point method*, for the method of alternating projections. The hybrid procedure is also a generalization of the β-extrapolation method discussed in [1].

Obviously, instead of chosing α which minimizes (r, r), it is possible to minimize (r, Zr) where Z is a symmetric positive definite matrix.

3. Recursive use of the procedure

Obviously, the approximate solutions x' and x'' can be produced by iterative methods. Thus we shall now denote their values at the k-th iteration by x'_k and x''_k and the corresponding residuals by r'_k and r''_k respectively. We shall also set y_k instead of y, r_k instead of r and α_k instead of α.

From the remark following (4), if one of the methods has a finite convergence (that is $\exists k$ such that $r'_k = 0$ or $r''_k = 0$), then so is the hybrid procedure (that is $r_k = 0$ for the same k).

Some possible strategies for using the hybrid procedure described in the first section are the following

1. compute x'_k and x''_k by two different methods
2. compute x'_k by some method and take $x''_k = x'_{k-1}$
3. compute x'_k by some method and take $x''_k = y_{k-1}$
4. compute x'_k from y_{k-1} and take $x''_k = y_{k-1}$
5. compute x'_k by some method and x''_k from y_{k-1}
6. compute x'_k by some method and x''_k from x'_k
7. compute x'_k and x''_k by the same method but with different starting points
8. . . .

In all the cases where y_{k-1} is used for computing y_k we shall take $y_0 = x'_0$ and $r_0 = r'_0$.

Let us now review these various strategies.

3.1. Case 1

x_k' and x_k'' are computed by two different methods.

This is the most general case which contains all the others. However it is too general and nothing can be said on its algebraic properties without particularizing the methods.

For example, we can built an hybrid procedure from Jacobi and Gauss-Seidel iterations or from Lanczos method and the conjugate gradient squared (CGS) method [30]. This case is an interesting one since we have respectively

$$r_k' = P_k(A)r_0' \quad \text{and} \quad r_k'' = P_k^2(A)r_0'$$

if $r_0'' = r_0'$. Thus $r_k'' = P_k(A)r_k'$ which shows that, usually, if Lanczos method is bad then the CGS is worse and that, if Lanczos method is good, then the CGS is still better. Since the residual vector r_k of the hybrid procedure has a smaller norm than the vectors r_k' and r_k'', the hybrid procedure in fact automatically selects the best method among Lanczos' and the CGS and it can even be better than both of them. A numerical example illustrating this fact will be given in Sect. 4. In this case we have

$$r_k = [\alpha_k + (1 - \alpha_k)P_k(A)]P_k(A)r_0'$$

which shows that the hybrid procedure produces iterates of the CGM class [8].

Of course, when the methods producing r_k' and r_k'' are completely independent one from each other, the cost of one iteration of the hybrid procedure is the sum of the individual costs of each of the underlying methods. However this case is very much convenient for a parallel computation of r_k' and r_k''. If both methods are not independent, then the cost of one iteration of the hybrid procedure can be lowered. This is, in particular, the case for the biconjugate gradient algorithm (BCG) [12] and the CGS [30] since, in both methods, the constants appearing in the recurrence relations are the same and thus they have to be computed only once. Then, a coupled implementation of the BCG and the CGS only requires 3 matrix by vector multiplications (instead of 4) and, moreover, A^* is no more needed (as in the BCG alone). Let us mention that a transpose-free implementation of the BCG (at the same cost) cal also be performed via the topological ε-algorithm and some relations derived from it [5]. Another possible coupled implementation with no extra cost is that of the quasi-minimal residual (QMR) of Freund [13] and Freund and Nachtigal [16] and of the BCG as explained in [17] (see also [15]). An even simpler coupled implementation of these methods was recently derived by Zhou and Walker [39]. Let us remark that the hybrid procedure requires no further matrix by vector multiplication.

3.2. Case 2

x_k' is computed by some method and we take $x_k'' = x_{k-1}'$.

This case corresponds to a semi-iterative method where two successive iterates of a method are combined to obtain a better result.

If the vectors x_k' are constructed by an iterative method of the form

$$x_{k+1}' = Bx_k' + b$$

and if the matrix $A = I - B$ is regular, then

$$r'_k = M^k r'_0$$

where $M = ABA^{-1}$. If the eigenvalues of B satisfy

$$\lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n$$

and if its eigenvectors are linearly independent, then it is well known that r'_k behaves like λ_1^k. After some calculations it can be proved that r_k obtained by the hybrid procedure will behave like λ_2^k, not a surprising result.

Our procedure could be compared, in this case, with Lyusternik acceleration method [2, 10, 11] which also makes use of two consecutive iterates but needs the knowledge of λ_1 [25].

If the matrix A is symmetric positive definite and if the basic method used is the conjugate gradient, then $(r'_k, r''_k) = 0$ and, from the results of the preceding section, the gain is assured since

$$\|r_k\| \leq \min(\|r'_k\|, \|r'_{k-1}\|).$$

Moreover, the hybrid procedure has also a finite convergence.

3.3. Case 3

x'_k is computed by some method and we take $x''_k = y_{k-1}$.

The idea here is cycling with the iterates obtained by the hybrid procedure and those given by an arbitrary method. Thanks to the minimization property of the hybrid procedure, the norm of the residual r_k decreases at each iteration since we have

$$\|r_k\| \leq \min(\|r'_k\|, \|r_{k-1}\|)$$

and thus the convergence of the process is smoothed. Moreover, if the basic method has a finite convergence (such as Lanczos method or the CGS) then so has the hybrid procedure.

It seems that this method was first introduced by Schönauer, Müller and Schnepf [29] (see also [28], pp. 261-262) for smoothing the convergence of the BCG. It was called the residual smoothing and was successfully used for many years. The terminology “minimal residual smoothing” (MRS) was coined, for this procedure, by Zhou and Walker [39] to distinguish it from other possible smootings. A complete theory of the MRS in the case where x'_k is computed by any iterative method (and not only the BCG) was given by Weiss in his thesis [35]. In particular, he proved that the MRS transforms generalized conjugate gradient methods which minimize the pseudo-residual into methods where the true residual is minimum. Thus it is not useful to apply the MRS to methods that minimize the true residual. A survey of these results can be found in [38]. It was recently proved by Zhou and Walker [39] that a smoothing algorithm (but with a different choice for α_k) transforms the BCG into the QMR and the CGS into the TFQMR [14]. They gave relations between the iterates of these two pairs of methods and extend the procedure to a quasi-minimal residual smoothing (QMRS) which can be applied to any iterative method. Other results relating to the smoothing technique can be found in [36] and [37].

3.4. Case 4

x'_k is computed from y_{k-1} and we take $x''_k = y_{k-1}$.

This case covers the method SOR if x'_k is given by the Gauss-Seidel iteration and Richardson's method [32]. More generally, it covers all the so-called *extrapolation* methods as described, for example, in [22]. Let us consider the regular splitting

$$A = M - N.$$

Taking $x''_k = y_{k-1}$ and $x'_k = M^{-1}Ny_{k-1} + M^{-1}b$ and setting

$$\alpha_k = \frac{(AM^{-1}r_{k-1}, r_{k-1})}{(AM^{-1}r_{k-1}, AM^{-1}r_{k-1})}$$

we obtain

$$\begin{aligned} y_k &= (1-\alpha_k)y_{k-1} + \alpha_k(M^{-1}Ny_{k-1} + M^{-1}b) \\ r_k &= (I - \alpha_k AM^{-1})r_{k-1}. \end{aligned}$$

In our case, α_k is chosen (by (1)) in order to minimize (r_k, r_k) and not, as usual, in order to minimize the spectral radius of the matrix $\alpha_k M^{-1}N + (1-\alpha_k)I$. Several choices of the matrix M are of interest

1. for the choice $M = I$, Richardson's method is recovered
2. the choice $M = D$, where D is the diagonal part of A, corresponds to Jacobi method for x''_k
3. the choice $M = D - E$, where $-E$ is the strictly lower part of A, leads to SOR. However in this case, α_k, which plays the rôle of ω, is chosen along (1) instead of ω_{opt}.
4. $M = (D - \omega E)/\omega$ corresponds to SOR for x''_k
5. $M = (D-\omega E)D^{-1}(D-\omega F)/[\omega(2-\omega)]$, where $-F$ is the strictly upper part of A, corresponds to SSOR for x''_k.

From (8), we have

$$(r_k, r_k) = (r_{k-1}, r_{k-1}) - \frac{(AM^{-1}r_{k-1}, r_{k-1})^2}{(AM^{-1}r_{k-1}, AM^{-1}r_{k-1})}.$$

Thus, from Schwarz inequality, $(r_k, r_k) = 0$ if and only if $AM^{-1} = aI$ where a is a non-zero scalar. This shows that a good choice for M is analogous to a good choice for a preconditioner, a name often given to M.

Iterations of the preceding type were considered by several authors. Mann [26] used them for solving the nonlinear equation $x = F(x)$ with $x'_k = F(y_{k-1})$. He took arbitrary α_k's in $]0, 1]$. The same technique was then discussed by Wegstein [33] but with a value of α_k which related his method to Aitken's Δ^2 process. The case of a system of nonlinear equations was treated similarly by Lemaréchal [24] who obtained a generalization of Aitken's process when α_k is chosen to minimize the Euclidean norm $\|x_k - F(x_k)\|$. His method reduces to our hybrid procedure when $F(x) = Ax+b$ and thus it appears as its generalization to the nonlinear case. A method where x'_k is computed from y_{k-1} and y_{k-2} was given by Weinitschke [34]. In his case, α_k is a constant independent of k. Mann iterations are the subject of an important literature. The interested reader is refered to [21] for a survey and to [3] and [4] for more recent results.

In the case of a system of linear equations, the paper of Dax [9] presents a procedure where x'_k is obtained by an arbitrary iterative method using y_{k-1} as a starting point and where α_k is chosen as above to minimize (r_k, r_k).

3.5. Case 5

x'_k is computed by some method and x''_k is computed from y_{k-1}.

This is a variant of the Case 3, where now x''_k is not taken directly as y_{k-1} but computed from it by any procedure.

In particular, one can think of setting

$$x''_k = -a_k r_{k-1} + y_{k-1} \quad \text{and} \quad r''_k = (1 + a_k A) r_{k-1}$$

and choosing a_k in order to minimize (r''_k, r''_k) or such that $(r'_k, r''_k) \leq 0$ since, as explained in the previous section, the gain brought by the hybrid procedure will be better in this case. However, it is easy to see that (r_k, r_k) is independent of a_k and, thus, such a choice is unuseful.

3.6. Case 6

x'_k is computed by some method and x''_k is computed from x'_k.

This is a variant of the Case 1, where now x''_k is not obtained by an arbitrary method but computed from x'_k by any procedure. Using Lanczos method and the CGS (or any other method of the CGM class [8]) can be considered as entering also into this case since $r''_k = Q_k(A) r'_k, Q_k$ being an arbitrary polynomial with $Q_k(0) = 1$.

In particular, one can think of setting

$$x''_k = -b_k r'_k + x'_k \quad \text{and} \quad r''_k = (1 + b_k A) r'_k$$

and choosing b_k in order to minimize (r''_k, r''_k). This idea is similar to the idea used for constructing the method called Bi-CGSTAB [31] and its variant [8]. These methods can be exactly recovered in our framework. One can also choose b_k such that $(r'_k, r''_k) \leq 0$. However, as in the preceding case, (r_k, r_k) will be independent of b_k and the procedure unuseful.

3.7. Case 7

x'_k and x''_k can be computed by the same method but with two different starting points x'_0 and x''_0.

This is only possible if the iterations have the form $x_{k+1} = F(x_k)$ where F does not depend on the iteration k (such as, for example, in Lanczos method and the CGS).

4. Applications

The linear systems we solved were mostly taken from [20]. Their dimension was always 50 (unless specified) and the right hand side was chosen so that the solution is $x = (1, \ldots, 1)^{\mathrm{T}}$.

In the methods where the residuals are computed by the given iterative algorithm (such as in the BCG, the CGS and the QMR), we used the residual estimates and not the actual residuals.

4.1. Example 1

Let us consider the $n \times n$ matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ a_1 & 1 & 1 & \cdots & 1 & 1 \\ a_1 & a_2 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & 1 \end{pmatrix}$$

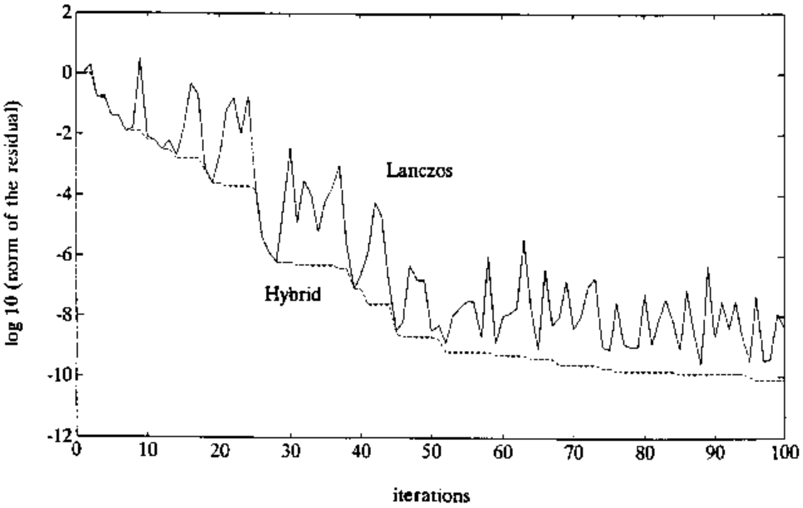

Fig. 2.

Its determinant is equal to $(1 - a_1)(1 - a_2) \cdots (1 - a_{n-1})$. Thus the matrix A is singular if and only if $a_i = 1$ for some i. With $a_i = 1 + i\varepsilon$ and $\varepsilon = 10^{-2}$, we obtain with the hybrid procedure and Lanczos method (Case 3, Subsect. 3.3) with $x_0 = 0$ the results of Fig. 2.

With $\varepsilon = 10^{-5}$, we have the results of Fig. 3.

Let us now consider two cases where the dimension of the system is $n = 1000$. With $\varepsilon = 10^{-4}$ we have the results of Fig. 4 and with $\varepsilon = 10^{-5}$ we have the results of Fig. 5.

In these four examples we see that, due to rounding errors, Lanczos method does not terminate after n iterations as expected from its theory but we also see (specially in Fig. 5) that the hybrid procedure has a smoothing effect on these rounding errors even when they destroy the convergence of Lanczos method. When the convergence is preserved (as in Fig. 4) the hybrid procedure provides an improvement.

Our four curves above are very similar to those given in [35] and [38].

4.2. Example 2

Combining now the hybrid procedure with Jacobi method we obtain for the matrix with elements $a_{ij} = i - j + 1$ and $x_0 = 0$ the results of Fig. 6.

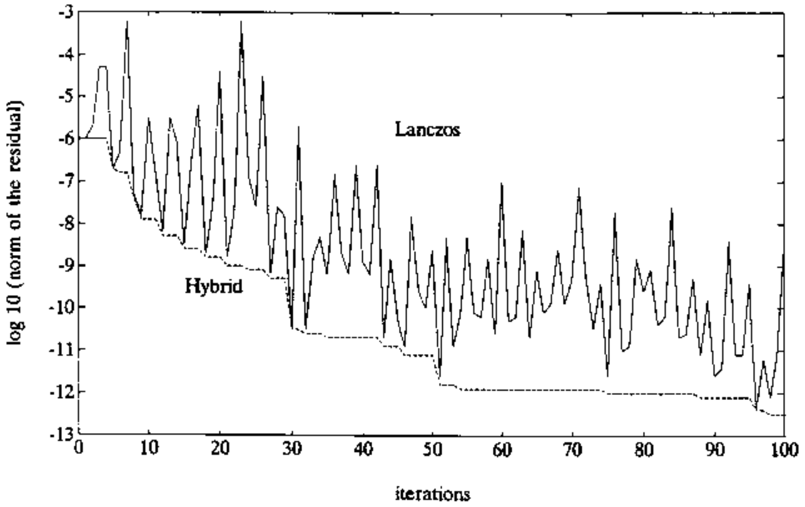
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
0
10
20
30
40
50
60
70
80
90
100
log 10 (norm of the residual)
iterations
Lanczos
Hybrid

Fig. 3.

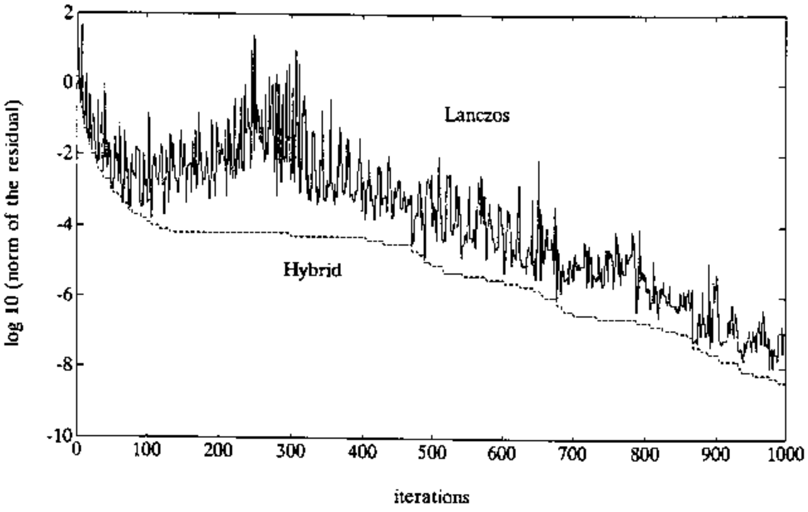
2
0
-2
-4
-6
-8
-10
0
100
200
300
400
500
600
700
800
900
1000
log 10 (norm of the residual)
iterations
Lanczos
Hybrid

Fig. 4.

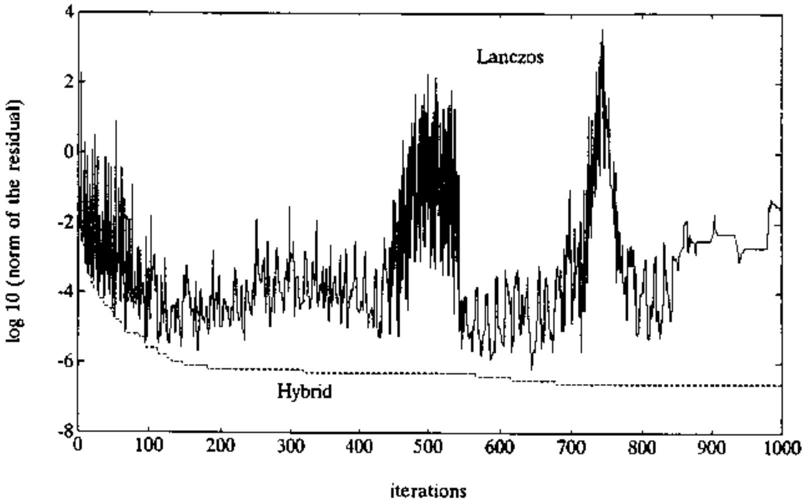

Fig. 5.

In this example, the method of Jacobi diverges while the hybrid procedure converges.

4.3. Example 3

Combining the hybrid procedure with the method of norm decomposition (for the Euclidean norm) due to Gastinel [18] and which is always convergent, we obtain for the matrix whose elements are $a_{ij} = a^i + b^j, a = 0.95, b = 0.90$ and for $x_0 = 0$ the results of Fig. 7.

4.4. Example 4

Let us now use Lanczos method together with Gauss-Seidel iterations. We take the matrix

$$A = \begin{pmatrix} -3 & 1 & & & & \\ 1 & -2 & 1 & & 0 & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & 1 & -2 & 1 \\ & & & & 1 & -1 \end{pmatrix}$$

Starting both methods from $x_0 = 0$, we obtain the results of Fig. 8.

As well known, rounding errors in Lanczos method strongly depend on the way it is implemented. For example, implementing the CGS by the subroutine BSMRZS given in [7] (without making any jump) we obtained the exact solution of this system after 50 iterations as expected from the theory.

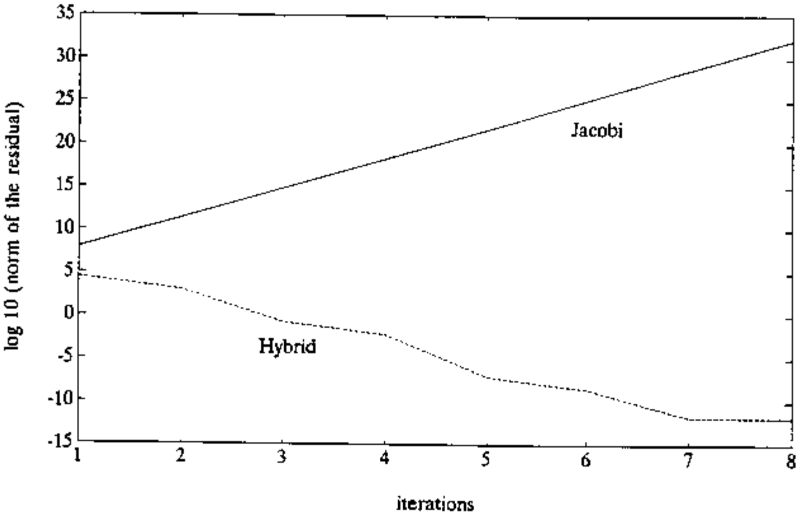
35
30
25
20
15
10
5
0
-5
-10
-15
1
2
3
4
5
6
7
8
log 10 (norm of the residual)
iterations
Jacobi
Hybrid

Fig. 6.

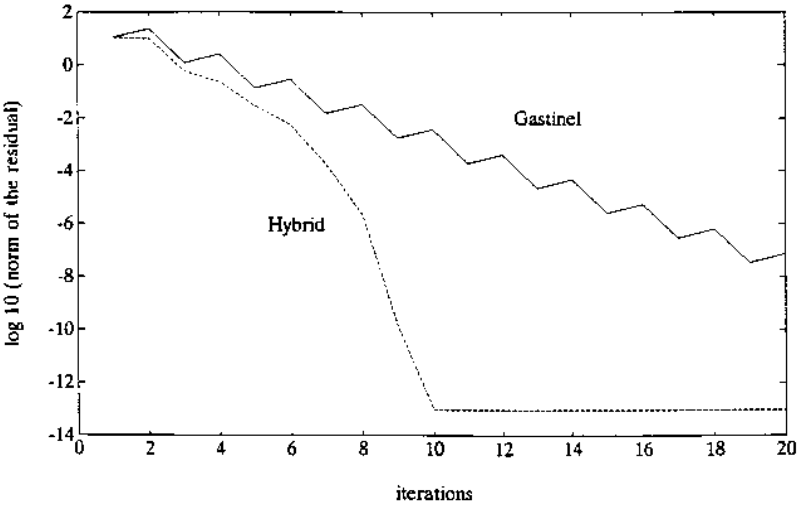
2
0
-2
-4
-6
-8
-10
-12
-14
0
2
4
6
8
10
12
14
16
18
20
log 10 (norm of the residual)
iterations
Gastinel
Hybrid

Fig. 7.

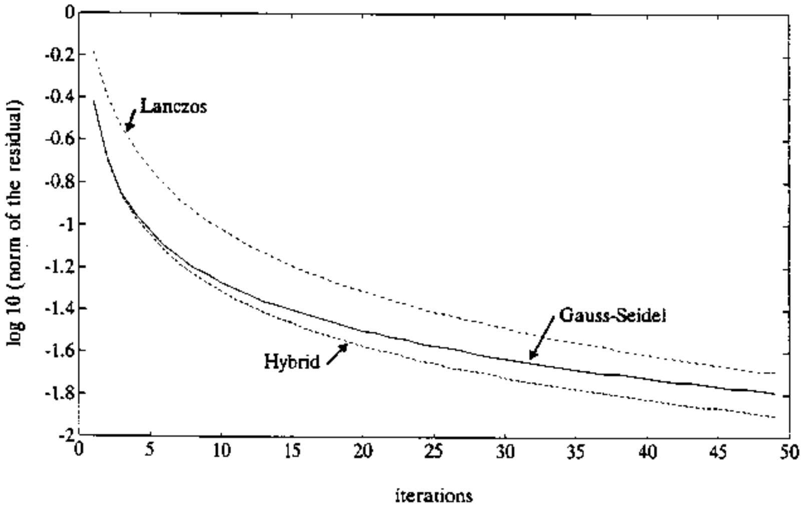

Fig. 8.

4.5. Example 5

Let us now consider the matrix given by $a_{ij} = a_{ji} = 51 - i$ if $i \geq j$ and combine the methods of Gastinel and Gauss-Seidel. With $x_0 = 0$ we obtain the results of Fig. 9.

We shall now try the same methods on the 2×2 block diagonal matrix whose blocks are

$$\begin{pmatrix} 1 & 1 \\ a & -1 \end{pmatrix}.$$

With $a = 1.4$ and $x_0 = 0$, the method of Gauss-Seidel diverges and we get the results of Fig. 10.

4.6. Example 6

Using the methods of Gauss-Seidel and Jacobi for the matrix

$$A = \begin{pmatrix} 1 & a & & & & \\ b & 1 & a & & 0 & \\ & b & 1 & a & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & b & 1 & a \\ & & & & b & 1 \end{pmatrix}$$

with $a = 0.7, b = 10^{-2}$ and $x_0 = 0$, we obtain the results of Fig. 11.

4.7. Example 7

Using the coupled implementation of the QMR and the BCG described in [17] and [15] we obtained, for the matrix of the first example with $\varepsilon = 10^{-7}$ and $\tilde{r}_0 = (1, \ldots, n)^{\mathrm{T}}$, the results of Fig. 12.

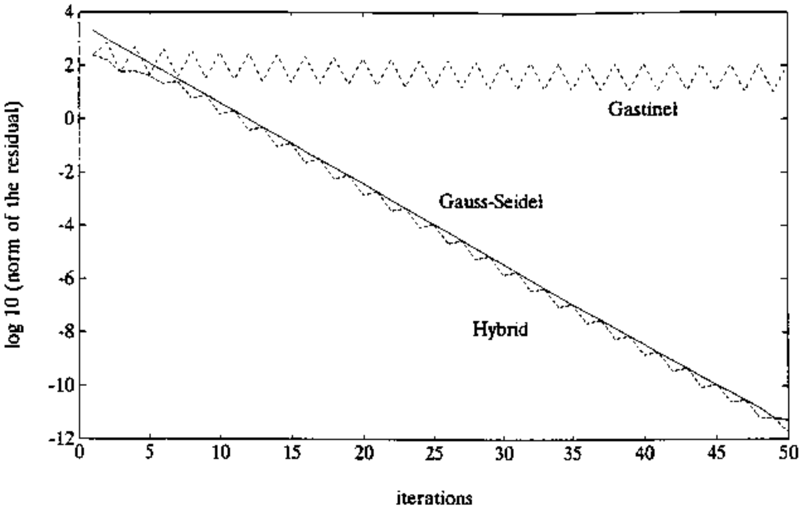
4
2
0
-2
-4
-6
-8
-10
-12
0
5
10
15
20
25
30
35
40
45
50
log 10 (norm of the residual)
iterations
Gastinel
Gauss-Seidel
Hybrid

Fig. 9.

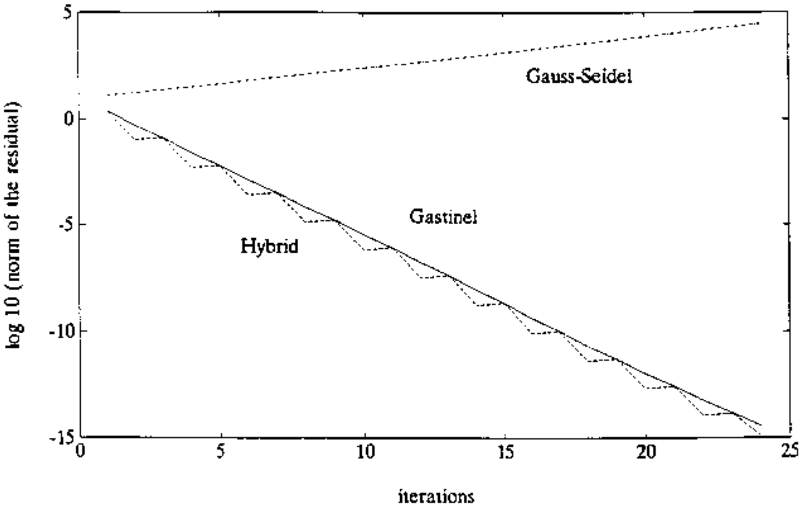
5
0
-5
-10
-15
0
5
10
15
20
25
log 10 (norm of the residual)
iterations
Gauss-Seidel
Gastinel
Hybrid

Fig. 10.

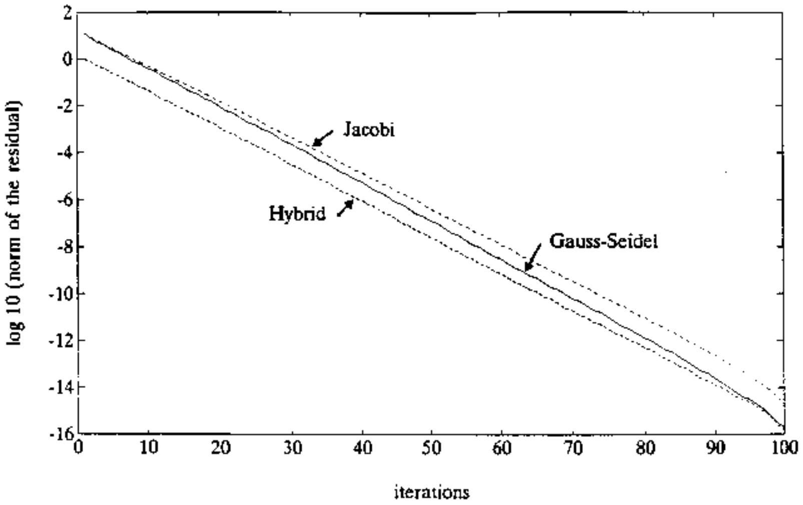
log 10 (norm of the residual)
2
0
-2
-4
-6
-8
-10
-12
-14
-16
0
10
20
30
40
50
60
70
80
90
100
iterations
Jacobi
Hybrid
Gauss-Seidel

Fig. 11.

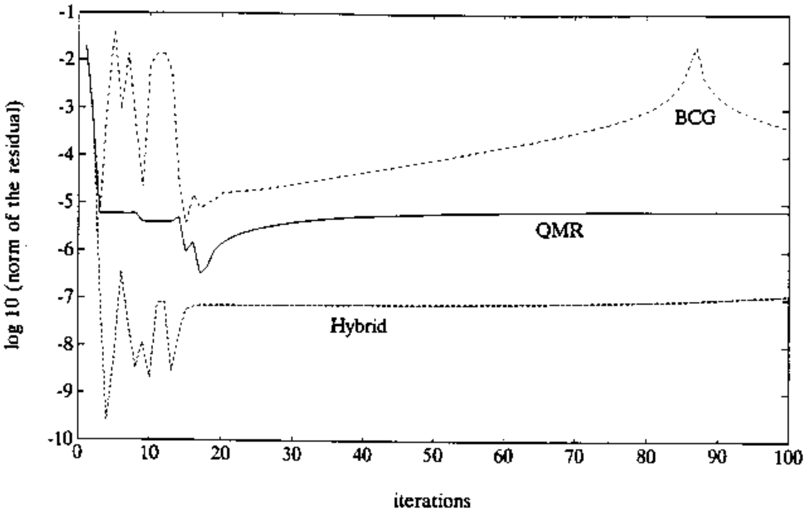
log 10 (norm of the residual)
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
0
10
20
30
40
50
60
70
80
90
100
iterations
BCG
QMR
Hybrid

Fig. 12.

4.8. Example 8

We have also programmed the coupled implementation of the BCG and the CGS and tried it on several systems. When both methods converge, the number of exact digits provided by the CGS is approximately the double of that given by the BCG. The best gain brought by the hybrid procedure was obtained for the example B_κ (5.9) of [27] with various values of ε. In that case, however, the gain was less than half a digit which can possibly be explained by the result of [39] stating that the CGS is transformed into the TFQMR by a smoothing technique quite close to ours.

5. Ideas for further research

There are several other possibilities for extending or using differently the hybrid procedure described in this paper. Here are some of them which are currently under consideration

- Restarting
  In the cases above where only one single method is involved in the hybrid procedure, it is possible to restart the method, at each step, with the vector obtained by the hybrid procedure. Since the residual given by the hybrid procedure has a smaller norm, such a restarting strategy could improve the convergence properties of the whole process.
- Cascade
  It is possible, with the hybrid procedure, to combine more than two different methods as follows: after combining the first two methods, one can combine the result obtained with a third method, then the result with a fourth one, and so on. Such a process could be called a hybrid procedure in cascade or a staircase hybrid procedure.
- Multiple hybrid procedure
  For combining k methods together, it is also possible to solve the problem of minimizing (r, r) with respect to $\alpha_1, \alpha_2, \ldots, \alpha_k$ where

  $$\alpha_1 + \alpha_2 + \cdots + \alpha_k = 1$$

  and

  $$r = \alpha_1 r_1 + \alpha_2 r_2 + \cdots + \alpha_k r_k$$

  $r_1, r_2, \ldots, r_k$ being the residuals corresponding to the k methods to be combined.
- Two stage hybrid procedure
  When an arbitrary iterative method is combined with the hybrid procedure (Case 3) we saw that it has a smoothing effect. This is no more the case when two different methods are combined (Case 1) and the convergence of the hybrid procedure can be quite irregular in this case. To avoid such a situation a two stage hybrid procedure can be introduced. It consists in considering the residual obtained by the hybrid procedure as an intermediate stage and then to use the hybrid procedure a second time for smoothing this intermediate residual. Thus, one iteration is as follows

  $$\begin{aligned} s_k &= \alpha_k r'_k + (1 - \alpha_k) r''_k \\ r_k &= \beta_k s_k + (1 - \beta_k) r_{k-1} \end{aligned}$$

  where α_k is chosen to minimize the norm of the intermediate residual s_k and β_k that of r_k.

We intend to come back to these questions in subsequent publications where the acceleration properties of the hybrid procedure will also be discussed.

Acknowledgements. We would like to thank the anonymous referee who pointed out the references [35] and [39] thus helping us connecting our hybrid procedure with other existing methods. A. Abkowicz, B. Germain-Bonne, K. Jbilou and H. Sadok are also thanked for helpful discussions and H.F Walker for several interesting e-mail messages. We also would like to thank R.S. Varga for many improvements.

References

1. Albrecht, P., Klein, M.P. (1984): Extrapolated iterative methods for linear systems. SIAM J. Numer. Anal. **21**, 192–201
2. Berezin, I.S., Zhidkov, N.P. (1965): Computing Methods. 2 volumes. Pergamon Press, Oxford
3. Borwein, D., Borwein, J. (1991): Fixed point iterations for real functions. J. Math. Anal. Appl. **157**, 112–126
4. Borwein, J., Reich, S., Shafrir, I. (1992): Krasnoselski-Mann iterations in normed spaces. Canad. Math. Bull. **35**, 21–28
5. Brezinski, C. (1980): Padé-Type Approximation and General Orthogonal Polynomials. ISNM **50**, Birkhäuser, Basel
6. Brezinski, C., Redivo-Zaglia, M. (1991): Extrapolation Methods. Theory and Practice. North-Holland, Amsterdam
7. Brezinski, C., Redivo-Zaglia, M., Treatment of near-breakdown in the CGS algorithm. (submitted)
8. Brezinski, C., Redivo-Zaglia, M., Look-ahead in Bi-CGSTAB and other methods for linear systems. (submitted)
9. Dax, A. (1990): Line search acceleration of iterative methods. Linear Alg. Appl. **130**, 43–63
10. Faddeeva, V.N. (1959): Computational Methods of Linear Algebra. Dover, New York
11. Faddeev, D.K., Faddeeva, V.N. (1963): Computational Methods of Linear Algebra. W.H. Freeman and Co., San Francisco
12. Fletcher, R. (1976): Conjugate gradient methods for indefinite systems. In: G.A. Watson (ed.) Numerical Analysis. Dundee 1975, LNM **506**, pp. 73–89, Springer, Berlin Heidelberg New York
13. Freund, R.W. (1992): Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. SIAM J. Sci. Stat. Comput. **13**, 425–448
14. Freund, R.W. (1993): A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. SIAM J. Sci. Stat. Comput. **14**, 470–482
15. Freund, R.W., Golub, G.H., Nachtigal, N.M. (1992): Recent advances in Lanczos-based iterative methods for nonsymmetric linear systems. Numerical Analysis Project, Manuscript NA-92-02. Computer Science Department, Stanford University
16. Freund, R.W., Nachtigal, N.M. (1991): QMR: a quasi-minimal residual method for non-Hermitian linear systems. Numer. Math. **60**, 315–339
17. Freund, R.W., Szeto, T. (1991): A quasi-minimal residual squared algorithm for non-Hermitian linear systems. Technical Report 91.26. RIACS NASA Ames Research Center, Moffett Field, CA
18. Gastinel, N. (1958): Procédé itératif pour la résolution numérique d'un système d'équations linéaires. C.R. Acad. Sci. Paris **246**, 2571–2574
19. Gearhart, W.B., Koshy, M. (1989): Acceleration schemes for the method of alternating projections. J. Comput. Appl. Math. **26**, 235–249
20. Gregory, R.T., Karney, D.L. (1969): A Collection of Matrices for Testing Computational Algorithms. Wiley, New York
21. Groetsch, C.W. (1977): Some aspects of Mann's iterative method for approximating fixed points. In: S. Karamardian (ed.) Fixed Points. Algorithms and Applications, pp. 349–365. Academic Press, New York
22. Kincaid, D., Cheney, W. (1991): Numerical Analysis. Brooks/Cole Publ. Co., Pacific Grove
23. Lanczos, C. (1952): Solution of systems of linear equations by minimized iterations. J. Res. Nat. Bur. Standards **49**, 33–53
24. Lemaréchal, C. (1971): Une méthode de résolution de certains systèmes non linéaires bien posés. C.R. Acad. Sci. Paris, sér. A, **272**, 605–607

25. Lyusternik, L.A. (1947): Remarks on the numerical solution of boundary value problems for Laplace's equation and the calculation of eigenvalues by the method of nets (in Russian). **20**, pp. 49–64, Trudy Matem. Inst. Im. Steklov
26. Mann, W.R. (1953): Mean value methods in iteration. Proc. Amer. Math. Soc. **4**, 506–510
27. Nachtigal, N.M., Reddy, S.C., Trefethen, L.N. (1992): How fast are nonsymmetric matrix iterations? SIAM J. Matrix Anal. Appl. **13**, 778–795
28. Schönauer, W. (1987): Scientific Computing on Vector Computers. North-Holland, Amsterdam
29. Schönauer, W., Müller, H., Schnepf, E. (1985): Numerical tests with biconjugate gradient type methods. Z. Angew. Math. Mech. **65**, T400–T402
30. Sonneveld, P. (1989): CGS, a fast Lanczos-type solver for nonsymmetric linear systems. SIAM J. Sci. Stat. Comp. **10**, 36–52
31. Van der Vorst, H.A. (1992): Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **13**, 631–644
32. Varga, R.S. (1962): Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs
33. Wegstein, J.H. (1958): Accelerating convergence of iterative processes. Comm. ACM **1** (#6), 9–13
34. Weinitschke, H.J. (1964): Über eine Klasse von Iterationsverfahren. Numer. Math. **6**, 395–404
35. Weiss, R. (1990): Convergence Behavior of Generalized Conjugate Gradient Methods. PhD Thesis, University of Karlsruhe
36. Weiss, R., Error-minimizing Krylov subspace methods. SIAM J. Sci. Stat. Comput. (to appear)
37. Weiss, R., Properties of generalized conjugate gradient methods. Linear Alg. Appl. (to appear)
38. Weiss, R., Schönauer, W. (1992): Accelerating generalized conjugate gradient methods by smoothing. In: R. Beauwens, P. de Groen (eds.) Iterative Methods in Linear Algebra, pp. 283–292. North-Holland, Amsterdam
39. Zhou, L., Walker, H.F., Residual smoothing techniques for iterative methods. SIAM J. Sci. Stat. Comput. (to appear)

This article was processed by the author using the LaTeX style file *pljour1* from Springer-Verlag.