

## On the Lanczos Method for Solving Symmetric Linear Systems with Several Right-Hand Sides\*

By Youcef Saad

**Abstract.** This paper analyzes a few methods based on the Lanczos algorithm for solving large sparse symmetric linear systems with several right-hand sides. The methods examined are suitable for the situation when the right sides are not too different from one another, as is often the case in time-dependent or parameter-dependent problems. We propose a theoretical error bound for the approximation obtained from a projection process onto a Krylov subspace generated from processing a previous right-hand side.

**1. Introduction.** In many applications we need to solve several symmetric linear systems of the form

$$(1.1) \quad Ax^{(i)} = b^{(i)}, \quad i = 1, 2, \dots, k,$$

involving the same  $N \times N$  coefficient matrix  $A$  but different right-hand sides  $b^{(i)}$ . When all of the right-hand sides are available simultaneously, then block methods such as the block-Lanczos or block-conjugate gradient algorithms [6], [12], and the block-Stiefel method [12], [13], can be successfully applied to solve (1.1).

However, it is often the case in practice that the right-hand sides are not available at the same time, i.e., that a given right-hand side  $b^{(i)}$  depends on the solutions  $x^{(j)}$ ,  $j = 1, \dots, i - 1$ , of the previous linear systems. In this case the block methods are no longer applicable. For this situation, Parlett [9] suggested using the Lanczos algorithm to solve the first system and saving the Lanczos vectors thus generated in order to provide good approximate solutions to the subsequent systems. For example, an approximate solution to the second linear system can be obtained by using a projection (Galerkin) technique onto the Krylov subspace generated when solving the first linear system. We refer to this as the Lanczos-Galerkin projection procedure. If the accuracy obtained from the Lanczos-Galerkin projection process alone is not sufficient, a further refinement is needed. The simplest way of improving the Lanczos-Galerkin approximation is to start a new Lanczos run with this approximation as an initial guess. This will be referred to as the restarted Lanczos-Galerkin procedure. An alternative is to use a special Lanczos process introduced by Parlett [9]. This algorithm consists of orthogonalizing the current Lanczos vector not

---

Received October 15, 1985; revised May 4, 1986 and July 18, 1986.

1980 *Mathematics Subject Classification.* Primary 65F10.

\* This work was supported in part by the Office of Naval Research under contract N00014-82-K-0184, in part by the Department of Energy under contract DE-AC02-81ER0996 and in part by the Army Research Office under contract DAAG-29-83-0177.

©1987 American Mathematical Society  
0025-5718/87 \$1.00 + \$.25 per page

only against the previous two vectors, as is done in the classical Lanczos procedure, but also against the last Lanczos vector of the previous Krylov subspace. We will refer to this technique as the modified Lanczos algorithm.

The purpose of the present paper is to analyze these techniques from a theoretical point of view. We will establish an error bound which shows that the Lanczos-Galerkin procedure provides a good accuracy under the condition that the residual vector of the new system is nearly contained in the previously generated Krylov subspace. We will also show that the restarted Lanczos-Galerkin method is, in some sense, equivalent to a block-Lanczos method.

**2. The Lanczos Algorithm for Solving Linear Systems.** As a background, we present in this section a brief description of the Lanczos method for solving symmetric linear systems. For details consult [9] and the references therein. Consider the (single) linear system

$$(2.1) \quad Ax = b,$$

where  $A$  is  $N \times N$  and symmetric. Suppose that a guess  $x_0$  to the solution is available, and let its residual vector be  $r_0 = b - Ax_0$ . In what follows, the norm  $\|\cdot\|$  denotes the Euclidean norm and  $\tau$  is some tolerance related to the machine precision and some norm of  $A$ . Then the Lanczos algorithm for solving (2.1) can be described as follows:

**THE LANCZOS ALGORITHM FOR SOLVING LINEAR SYSTEMS.**

1. *Start:* Compute  $\beta_1 := \|r_0\|$ , and  $v_1 := r_0/\beta_1$ .

2. *Generate the Lanczos vectors:* For  $j = 1, 2, \dots$ , do

$$(2.2) \quad \begin{aligned} \hat{v}_{j+1} &:= Av_j - \beta_j v_{j-1} \quad (v_0 \equiv 0), \\ \alpha_j &:= (\hat{v}_{j+1}, v_j), \end{aligned}$$

$$(2.3) \quad \hat{v}_{j+1} := \hat{v}_{j+1} - \alpha_j v_j,$$

$$(2.4) \quad \beta_{j+1} := \|\hat{v}_{j+1}\|.$$

If  $\beta_{j+1} < \tau$  then set  $m := j$  and go to 3; else, compute

$$(2.5) \quad v_{j+1} := \hat{v}_{j+1}/\beta_{j+1}.$$

If convergence is reached, then set  $m := j$  and go to 3, else continue.

3. *Form the approximate solution:*

$$(2.6) \quad x_m := x_0 + V_m T_m^{-1}(\beta_1 e_1),$$

where  $V_m$  is the  $N \times m$  matrix

$$(2.7) \quad V_m = [v_1, v_2, \dots, v_m]$$

and  $T_m$  is the tridiagonal matrix

$$(2.8) \quad T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}.$$

The convergence test at the end of step 2 is feasible because there exists a simple formula that can be updated inexpensively at every iteration of step 2, and which allows one to compute the residual norm without actually computing the approximate solution [9]. In exact arithmetic, the vectors  $v_i$  computed from this process form an *orthonormal basis* of the Krylov subspace  $K_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}$ . Note that we have  $V_m^T A V_m = T_m$ , so that  $T_m$  is nothing but the matrix representation of the projection of  $A$  onto the Krylov subspace  $K_m$  with respect to the basis  $V_m$ . Furthermore, it is easily seen that the Lanczos algorithm realizes a projection process, i.e., a Galerkin process, onto the Krylov subspace  $K_m$ ; see, e.g., [9], [10]. Specifically, the approximate solution  $x_m$  can be found by noting that it belongs to the affine subspace  $x_0 + K_m$  and that its residual vector  $b - Ax_m$  is orthogonal to  $K_m$ . If we denote by  $P_m$  the orthogonal projector onto  $K_m$ , this means that the Lanczos method solves the approximate problem:

$$(2.9) \quad \text{Find } x_m \in x_0 + K_m \text{ such that } P_m(b - Ax_m) = 0.$$

In exact arithmetic, the approximation  $x_m$  thus computed is identical with that provided by  $m$  steps of the conjugate gradient (CG) method when  $A$  is positive definite [9]. When  $A$  is not positive definite, the conjugate gradient method may break down or become unstable, but this relationship between the Lanczos and the CG algorithms has been exploited to derive stable generalizations of the conjugate gradient algorithm to indefinite systems [2], [7], [9], [11].

A well-known and troublesome misbehavior of the Lanczos algorithm is the loss of orthogonality of the  $v_i$ 's. Although this does not prevent the method from converging, it often causes significant slowdown. Parlett [9] and Simon [15] have made the important observation that the fast convergence properties can be regained by resorting to different sorts of partial reorthogonalizations. This important matter will be further discussed in the last section.

**3. The Lanczos-Galerkin Projection Method.** Consider now the two linear systems

$$(3.1) \quad Ax^{(i)} = b^{(i)}, \quad i = 1, 2,$$

and assume that  $m$  steps of the Lanczos algorithm described in the previous section have been performed to solve the first system in a first pass. We will use the notation of Section 2 for the resulting variables and definitions associated with this first linear system: Thus,  $P_m$  denotes the orthogonal projector onto the Krylov subspace  $K_m = \text{span}\{v_1, Av_1, \dots, A^{m-1}v_1\}$ , where  $v_1$  is obtained by normalizing the initial residual vector  $r_0 = b^{(1)} - Ax_0^{(1)}$ . We wish to use the information gathered during the solution of the first system to provide an approximation to the second system,

$$(3.2) \quad Ax^{(2)} = b^{(2)}.$$

Clearly, we must assume that the vectors  $v_i$ ,  $i = 1, 2, \dots, m$ , as well as the tridiagonal matrix (2.8), have been saved, possibly in some secondary storage.

Suppose that we know an initial guess  $x_0^{(2)}$  to the solution of (3.2) and let  $r_0^{(2)}$  be the corresponding residual vector, i.e.,  $r_0^{(2)} = b^{(2)} - Ax_0^{(2)}$ . A natural way of improving the approximation  $x_0^{(2)}$  is by means of a Galerkin projection onto the Krylov subspace  $K_m$  generated for the solution of the first system. Such an approximation is obtained by solving the  $m$ -dimensional problem

$$(3.3) \quad P_m(b^{(2)} - Az) = 0$$

for  $z$  in the affine subspace  $x_0^{(2)} + K_m$ , i.e., for  $z = x_0^{(2)} + y$ ,  $y \in K_m$ . For the variable  $y$ , the condition (3.3) translates into the Galerkin problem

$$(3.4) \quad P_m(r_0^{(2)} - Ay) = 0,$$

or equivalently,

$$V_m^T(r_0^{(2)} - Ay) = 0,$$

whose solution is  $\bar{y} = V_m T_m^{-1} V_m^T r_0^{(2)}$ . This yields the desired approximation

$$(3.5) \quad \bar{z} = x_0^{(2)} + \bar{y} = x_0^{(2)} + V_m T_m^{-1} V_m^T r_0^{(2)}.$$

We will refer to the above method as the Lanczos-Galerkin process. This process requires solving a tridiagonal linear system of size  $m$  and forming a linear combination of  $m$  vectors of length  $N$ . The matrix  $A$  is invoked only when computing the residual  $r_0^{(2)}$ . This means that the whole procedure requires only one call to the matrix-by-vector multiplication subroutine.

Note that since  $P_m$  is a projector, we can rewrite (3.4) as

$$(3.6) \quad P_m(P_m r_0^{(2)} - Ay) = 0,$$

which will be useful shortly.

An important question regarding the approximation obtained from the above Lanczos-Galerkin process is its accuracy. The usual norm used to analyze the rate of convergence of the conjugate gradient type methods, when  $A$  is positive definite, is the  $A^{-1}$ -norm defined by  $\|x\|_{A^{-1}} = (A^{-1}x, x)^{1/2}$ . An important optimality property, which allows one to derive error bounds in projection methods, is that when  $A$  is positive definite the solution of any Galerkin problem of the form (3.3) minimizes the  $A^{-1}$ -norm of the residual vector  $b - Az$  over all vectors  $z$  in  $x_0^{(2)} + K_m$ ; see, e.g., [2], [4], [14]. Alternatively, the vector  $\bar{y}$  as defined by (3.5) is solution of the Galerkin problem (3.6), and therefore it minimizes the  $A^{-1}$ -norm of the residual vector  $P_m r_0^{(2)} - Ay$  over all vectors  $y$  of  $K_m$ . We are now in a position to state the main result of this section.

**THEOREM 3.1.** *Assume that  $A$  is symmetric positive definite with smallest eigenvalue  $\lambda_1$  and largest eigenvalue  $\lambda_N$ . Let the projected residual  $P_m r_0^{(2)}$  be expressed in the Krylov basis as*

$$(3.7) \quad P_m r_0^{(2)} = \sum_{j=1}^m \eta_j A^{j-1} v_1.$$

*Then the approximation  $\bar{z}$  obtained from the Lanczos-Galerkin projection process (3.5) is such that*

$$(3.8) \quad \|b^{(2)} - A\bar{z}\|_{A^{-1}} = \|(I - P_m)r_0^{(2)}\|_{A^{-1}} + \epsilon,$$

where

$$(3.9) \quad |\epsilon| \leq \frac{\|\eta_1 v_1\|_{A^{-1}}}{T_m(\gamma)},$$

in which  $\gamma = (\lambda_N + \lambda_1)/(\lambda_N - \lambda_1)$  and  $T_m$  represents the Chebyshev polynomial of the first kind of degree  $m$ .

*Proof.* The residual vector  $\bar{r} = b^{(2)} - A\bar{z} = b^{(2)} - A(x_0^{(2)} + \bar{y}) = r_0^{(2)} - A\bar{y}$  can be decomposed as

$$(3.10) \quad \bar{r} = (I - P_m)r_0^{(2)} + (P_m r_0^{(2)} - A\bar{y}),$$

from which we get by the second triangle inequality

$$(3.11) \quad \left| \|\bar{r}\|_{A^{-1}} - \|(I - P_m)r_0^{(2)}\|_{A^{-1}} \right| \leq \|P_m r_0^{(2)} - A\bar{y}\|_{A^{-1}},$$

which can be rewritten in the form

$$(3.12) \quad \|\bar{r}\|_{A^{-1}} = \|(I - P_m)r_0^{(2)}\|_{A^{-1}} + \alpha \|P_m r_0^{(2)} - A\bar{y}\|_{A^{-1}} \quad \text{with } -1 \leq \alpha \leq 1.$$

As was indicated earlier,  $\bar{y}$  minimizes  $\|P_m r_0^{(2)} - Ay\|_{A^{-1}}$  over all vectors  $y$  in  $K_m$ . Let  $y$  be an arbitrary vector in  $K_m$  which we express as  $y = s(A)v_1$ , where  $s$  is a polynomial of degree  $\leq m - 1$ , and let us denote by  $q$  the polynomial  $q(t) \equiv \sum_{j=1}^m \eta_j t^{j-1}$  and by  $P_j$  the set of all polynomials  $p$  of degree  $\leq j$ . Then we have

$$(3.13) \quad \begin{aligned} \|P_m r_0^{(2)} - A\bar{y}\|_{A^{-1}} &= \min_{y \in K_m} \|q(A)v_1 - Ay\|_{A^{-1}} \\ &= \min_{s \in P_{m-1}} \|(q(A) - As(A))v_1\|_{A^{-1}} \\ &= \min_{p \in P_m, p(0)=\eta_1} \|p(A)v_1\|_{A^{-1}} = |\eta_1| \min_{p \in P_m, p(0)=1} \|p(A)v_1\|_{A^{-1}}. \end{aligned}$$

The last term of the right-hand side is a classical factor in the theory of the conjugate gradient method, and a well-known upper bound for it yields (e.g., [5])

$$(3.14) \quad \|P_m r_0^{(2)} - A\bar{y}\|_{A^{-1}} \leq |\eta_1| \frac{\|v_1\|_{A^{-1}}}{T_m(\gamma)}$$

with  $\gamma$  defined in the theorem. The result follows by combining (3.12) and (3.14).  $\square$

Before discussing the above result, we will make a few comments. First, we should emphasize that  $\epsilon$  in the theorem is not necessarily positive. We prefer an equality of the form (3.8) rather than a simpler upper bound for the residual norm, because we wish to examine closely the particular cases when one of the terms in the right-hand side of (3.8) is much smaller than the other. Another remark is that the expansion (3.7) can be replaced by a simpler, but equivalent, expansion of the form

$$(3.15) \quad P_m r_0^{(2)} = \eta_1 v_1 + Ag,$$

where  $g$  is a vector of  $K_{m-1}$ . The above expansion is unique and is based on the decomposition of the subspace  $K_m$  into the direct sum  $K_m = \text{span}\{v_1\} \oplus AK_{m-1}$ .

The scalar  $\eta_1$  is neither known a priori nor easy to estimate. However, it is readily computable by noting that  $\eta_1 = q(0)$ , where  $q$  is the polynomial defined in the proof, and then using the available expansion  $P_m r_0^{(2)} = \sum_{i=1}^m \xi_i v_i$ . Since each  $v_i$  is of the form  $v_i = q_{i-1}(A)v_1$ , where  $q_{i-1}$  is the  $i$ th Lanczos polynomial [8], we have  $\eta_1 = \sum_{i=1}^m \xi_i q_{i-1}(0)$ . The sequence  $q_j(0)$  can be obtained inexpensively, thanks to the three-term recurrence of the Lanczos polynomials,

$$\beta_{j+1}q_j(\lambda) = (\lambda - \alpha_j)q_{j-1}(\lambda) - \beta_j q_{j-2}(\lambda), \quad j = 1, 2, \dots,$$

with the convention that  $q_{-1}(\lambda) \equiv 0$ .

The assumption that  $A$  is positive definite is made only to provide the standard tool for analyzing the convergence of the method. It is by no means necessary for the definition of the algorithm.

We now interpret the result of the theorem. Notice that if  $r_0^{(2)}$  belongs to the previous Krylov subspace  $K_m$ , then the term  $\|(I - P_m)r_0^{(2)}\|_{A^{-1}}$  in the right-hand side of (3.8) vanishes. Then the theorem tells us that in this case the method will provide a good accuracy when  $\eta_1$  is not too large. In fact, the accuracy will be of the same order as that obtained from  $m$  steps of the classical conjugate gradient method. Note that if  $\eta_1 = 0$  then the term  $\epsilon$  reduces to zero. As a consequence, an extreme case where the new system can be *exactly* solved by the application of the projection process occurs when  $(I - P_m)r_0^{(2)} = 0$ , i.e.,  $r_0^{(2)}$  belongs to  $K_m$ , and when simultaneously  $\eta_1 = 0$ , i.e.,  $r_0^{(2)}$  has no component in  $v_1$ . These two conditions can be summarized by the single condition that  $r_0^{(2)} \in AK_{m-1}$ .

The opposite extreme case is when the projection process leaves the starting approximate solution  $x_0^{(2)}$  unchanged. This happens when  $P_m r_0^{(2)} = 0$ , i.e., when  $r_0^{(2)}$  is orthogonal to  $K_m$ . In this case  $\bar{y} = 0$ , i.e.,  $\bar{z} = x_0^{(2)}$ , and the theorem yields the obvious result  $\|\bar{r}\|_{A^{-1}} = \|r_0^{(2)}\|_{A^{-1}}$  because  $\eta_1 = 0$ .

More realistic situations arising in practice will lie somewhere between these two extremes. For these general cases, the theorem shows that the error is the sum of a 'small' part  $\epsilon$  and a 'large' part  $\|(I - P_m)r_0^{(2)}\|_{A^{-1}}$ . The 'small' part is usually as small as would be obtained from  $m$  'average' steps of the usual conjugate gradient algorithm. The 'large' part depends essentially on how far the new system is from the previous one, and can be quite large as compared with  $\epsilon$ . Perhaps the most interesting and useful situations arise in time-dependent, or more generally parameter-dependent problems, in which the right-hand sides  $b^{(i)}$ ,  $i = 1, 2, \dots$ , change very little. Then the system can be expected to be solved relatively accurately because the 'large' term  $\|(I - P_m)r_0^{(2)}\|_{A^{-1}}$  becomes small.

When  $\|(I - P_m)r_0^{(2)}\|_{A^{-1}}$  is large, as compared with  $\epsilon$ , then the theorem indicates that the error in the  $A^{-1}$ -norm sense *cannot be decreased below*  $\|(I - P)r_0^{(2)}\|_{A^{-1}}$  *by the projection process (3.5) alone*. This means that we must further improve the solution, and this is the subject of the next section.

**4. Refining the Lanczos-Galerkin Approximation.** We start by summarizing the essential features of the two stages of the process described in the previous section.

1. A first linear system  $Ax^{(1)} = b^{(1)}$  has been solved by the Lanczos method. As a by-product we obtain a Krylov subspace  $K_m$  of dimension  $m$ , an orthonormal basis  $V_m = [v_1, v_2, \dots, v_m]$  of that subspace and a tridiagonal matrix  $T_m$  representing the section of  $A$  in  $K_m$  with respect to this basis.

2. We now want to solve a new linear system  $Ax^{(2)} = b^{(2)}$ . Some initial guess  $x_0^{(2)}$  is improved by means of a Lanczos-Galerkin projection process onto the Krylov subspace generated from the previous linear system. This yields the approximation  $\bar{z}$ .

The accuracy of the approximation  $\bar{z}$  thus obtained may be far from sufficient, as is shown by the comments at the end of the previous section. We are therefore faced with the problem of improving the approximation  $\bar{z}$  still further. We consider two different procedures for achieving this.

4.1. *Starting a New Lanczos Process.* The simplest way of improving the current approximation  $\bar{z}$  obtained from the projection process is to start a fresh Lanczos process from the current approximation  $\bar{z}$ , as described in Section 2. Let  $\bar{v}_i$ ,  $i = 1, 2, \dots, m$ , be the new Lanczos vectors with  $\bar{v}_1 = \bar{r}/\|\bar{r}\|$ . The initial residual  $\bar{r}$  of the new Lanczos iteration, and therefore also the initial vector  $\bar{v}_1$ , has the property of being orthogonal to the subspace  $K_m$  associated with the previous linear system. This is a peculiar property for an initial vector of the Lanczos algorithm, and an important question that arises is whether convergence is likely to be affected by it. As will be seen shortly, the answer is yes. We will need the following lemma.

LEMMA 4.1. *Let  $\bar{v}_1$  be any vector that is orthogonal to some Krylov subspace  $K_m$ . Then for any integer  $k$  not exceeding  $m - 1$  we have*

$$(A^k \bar{v}_1, v_i) = 0, \quad 1 \leq i \leq m - k,$$

where  $\{v_1, v_2, \dots, v_m\}$  is the Lanczos basis of  $K_m$ .

*Proof.* Because of the symmetry of  $A$ , we must show that  $(\bar{v}_1, A^k v_i) = 0$ , for  $1 \leq i \leq m - k$ . For this it suffices to prove that  $A^k v_i$  is a member of  $K_m$  for any integer  $i$  satisfying  $k + i \leq m$ . This property can be established by a simple induction argument on  $k$ , as follows. It is obviously true for  $k = 0$ . Assume that it is true for  $k$ , and consider

$$\begin{aligned} A^{k+1} v_i &= A^k [A v_i] = A^k [\beta_{i+1} v_{i+1} + \alpha_i v_i + \beta_i v_{i-1}] \\ &= \beta_{i+1} A^k v_{i+1} + \alpha_i A^k v_i + \beta_i A^k v_{i-1}, \end{aligned}$$

with the usual convention that  $v_0 \equiv 0$ . By the induction hypothesis, each of the terms in the right-hand side belongs to  $K_m$ , provided that  $k + (i + 1) \leq m$ . The proof is complete.  $\square$

A consequence is that for any polynomial  $q_{k-1}$  of degree  $\leq k - 1$  we have

$$(q_{k-1}(A) \bar{v}_1, v_i) = 0, \quad 1 \leq i \leq m - k + 1.$$

In particular, any vector  $\bar{v}_k$  of the new Lanczos sequence can be written in the form  $\bar{v}_k = q_{k-1}(A) \bar{v}_1$  for some polynomial of degree  $k - 1$ , and therefore we have

$$(\bar{v}_k, v_i) = 0, \quad i + k \leq m + 1.$$

Thus, the vectors  $\{\bar{v}_i\}$  and  $\{v_k\}$  do not only form two orthonormal systems, but they are also orthogonal to each other, for  $i + k \leq m + 1$ . In particular, as long as  $i$  does not exceed  $\lfloor m/2 \rfloor$ , the system  $\{v_1, \bar{v}_1, v_2, \bar{v}_2, \dots, v_i, \bar{v}_i\}$  of dimension  $2i$  forms an orthonormal basis of the subspace

$$\begin{aligned} K_i^* &= \text{span}\{v_1, A v_1, \dots, A^{i-1} v_1; \bar{v}_1, A \bar{v}_1, \dots, A^{i-1} \bar{v}_1\} \\ &= \text{span}\{Z_1, A Z_1, \dots, A^{i-1} Z_1\}, \end{aligned}$$

in which  $Z_1 = [v_1, \bar{v}_1]$ . A well-known method that uses a projection process on a subspace of this form is the block-Lanczos method, or its equivalent block-Conjugate-Gradient method [6], [13]. This algorithm is a block generalization of the Lanczos algorithm of Section 1, in which  $v_1$  (and subsequently the  $v_i$ 's) is replaced by a block of several column vectors. Thus, the above subspaces  $K_i^*$  are nothing but those used in the block-Lanczos method with blocksize 2 and initial block  $Z_1$ . The approximate solution generated at the  $i$ th step of the new Lanczos run belongs to

the affine subspace  $\bar{z} + K_i^*$ . Moreover, its residual vector is proportional to the vector  $\bar{v}_{i+1}$  which is orthogonal to the subspace  $K_i^*$ . These two properties characterize the approximate solution obtained by a block-Lanczos algorithm. Thus we have proved the following theorem.

**THEOREM 4.1.** *For  $i \leq m/2$ , the Lanczos process applied to the linear system  $Ax^{(2)} = b^{(2)}$ , started with the initial vector  $\bar{z}$  provided by the Lanczos-Galerkin approximation, is mathematically equivalent to the block-Lanczos method with block dimension of 2 and starting block  $Z_1 = \{v_1, \bar{v}_1\}$ .*

The rate of convergence of the block-Lanczos algorithm for solving linear systems has been studied in [6], [12] and we will not report the results here. Suffice it to say that, not surprisingly, the bounds on the rate of convergence of the block method are superior to those of the single-vector method. Thus, the first steps of the new Lanczos run enjoy the fast convergence of the block-Lanczos method without the usual additional cost.

**4.2. The Modified Lanczos Process.** An alternative to the procedure described above is one that continues naturally the previous Lanczos process by performing a Galerkin method onto a sequence of subspaces containing  $K_m$  and of increasing dimension. One such procedure was first introduced by Parlett [9] and was later rediscovered by Carnoy and Geradin [1] in a different context. The following algorithm, which will be referred to as the modified Lanczos algorithm, differs only in its presentation from Parlett's algorithm and the algorithm in [1]. Its inner loop computes a sequence of vectors  $w_i$ ,  $i = 1, 2, \dots$ , which are orthonormal to each other and also orthogonal to the  $v_i$ 's,  $i = 1, \dots, m$ , generated for the first system.

**THE MODIFIED LANCZOS ALGORITHM.**

1. *Start:* Compute  $\bar{\beta}_1 := \|\bar{r}\|$  and  $w_1 := \bar{r}/\bar{\beta}_1$ .

2. *Iterate:* For  $j = 1, 2, \dots$ , do

$$\hat{w}_{j+1} := Aw_j - \bar{\beta}_j w_{j-1} \quad (w_0 \equiv 0),$$

$$\bar{\alpha}_j := (\hat{w}_{j+1}, w_j),$$

$$\hat{w}_{j+1} := \hat{w}_{j+1} - \bar{\alpha}_j w_j,$$

$$\delta_j = (\hat{w}_{j+1}, v_m),$$

$$\hat{w}_{j+1} := \hat{w}_{j+1} - \delta_j v_m,$$

$$\bar{\beta}_{j+1} := \|\hat{w}_{j+1}\|.$$

If  $\beta_{j+1} < \tau$  then set  $p := j$  and go to 3; else, compute

$$w_{j+1} := \hat{w}_{j+1}/\bar{\beta}_{j+1}.$$

If convergence is reached, then set  $p := j$  and goto 3, else continue.

3. *Form the appropriate solution:*

$$z_p := \bar{z} + W_p T_{m,p}^{-1} (\bar{\beta}_1 e_{m+1}),$$

where  $e_{m+1}$  is the vector of length  $m + p$  whose  $(m + 1)$ st component is 1 and all others are 0,

$$W_p \equiv [v_1, v_2, \dots, v_m, w_1, \dots, w_p],$$



Using again the induction hypothesis, we see that all these terms are zero. This completes the proof of (4.1).

Now consider (4.2):

$$(w_{j+1}, w_i) = [\bar{\beta}_{j+1}]^{-1} [(Aw_j, w_i) - \bar{\alpha}_j(w_j, w_i) - \bar{\beta}_j(w_{j-1}, w_i) - \delta_j(v_m, w_i)].$$

Assume first that  $i < j - 1$ . By the induction hypothesis, the three terms  $(w_j, w_i)$ ,  $(w_{j-1}, w_i)$ , and  $(v_m, w_i)$  in the above expression vanish. Proceeding as before, the remaining term  $(Aw_j, w_i)$  is expanded as follows:

$$\begin{aligned} (Aw_j, w_i) &= (w_j, Aw_i) = (w_j, \bar{\beta}_{i+1}w_{i+1} + \bar{\alpha}_i w_i + \bar{\beta}_i w_{i-1} + \delta_i v_m) \\ &= \bar{\beta}_{i+1}(w_j, w_{i+1}) + \bar{\alpha}_i(w_j, w_i) + \bar{\beta}_i(w_j, w_{i-1}) + \delta_i(w_j, v_m), \end{aligned}$$

which, by a final application of the induction hypothesis, shows that  $(Aw_j, w_i) = 0$ . Hence,  $(w_{j+1}, w_i) = 0$ , for  $i < j - 1$ . For  $i = j$  and  $i = j - 1$ , the scalars  $\bar{\alpha}_j$ ,  $\bar{\beta}_j$ , and  $\delta_j$  have been chosen precisely so that the property is true. This completes the proof.  $\square$

Consider the subspace spanned by the orthonormal system  $W_p$ , which we will denote by  $K_{m,p}$ . The matrix representation  $W_p^T A W_p$  of the section of  $A$  in the subspace  $K_{m,p}$  with respect to the basis  $W_p$  is precisely the  $(m+p) \times (m+p)$  matrix  $T_{m,p}$ . Hence, the new approximate solution obtained at step  $p$  of the projection process onto the subspace  $K_{m,p}$  is given by

$$z_p = \bar{z} + W_p T_{m,p}^{-1} W_p^T \bar{r}.$$

Noticing that  $W_p^T \bar{r} = W_p^T \|\bar{r}\| w_1$ , this simplifies to

$$z_p = \bar{z} + \|\bar{r}\| W_p T_{m,p}^{-1} e_{m+1},$$

which explains the formula used in step 3 of the algorithm.

Note that  $K_{m,p}$  is no longer a Krylov subspace. It contains the subspace  $K_m$ , but the newly introduced  $w_j$  vectors bear no obvious relation with any basis vectors used in standard projection techniques. To be more accurate, a vector  $w_i$  assumes the form

$$w_{i+1} = p_i(A)w_1 + q_{i-1}(A)v_m, \quad i \geq 0,$$

where  $p_i$  and  $q_{i-1}$  are polynomials of degree  $i$  and  $i - 1$ , respectively, with the convention that  $q_{-1} \equiv 0$ . An arbitrary element of  $K_{m,p}$  is a linear combination of  $w_i$ 's and  $v_i$ 's, but this does not seem to lead to a simple expression in terms of subspaces similar to those associated with the Lanczos algorithm or the block-Lanczos algorithm. We can only say that  $K_{m,p} \subset K_m[v_1] + K_p[w_1, v_m]$ , where  $K_i[Z]$  here denotes the  $i$ th Krylov (or block-Krylov) subspace generated from the vector (or block)  $Z$ .

**5. Practical Considerations.** An important feature of both the Lanczos-Galerkin process and the modified Lanczos process is that we must save a large number of vectors in secondary storage. This may seem impractical at first, but there are numerous reasons why it is not always so:

- Once a vector has been computed, it is not needed until the convergence of the process is reached. There exists a simple formula for evaluating the residual norm of the solution without even having to compute the solution [9], [11], thus allowing the determination of the integers  $m$  and  $p$ .

- There are supercomputer systems with very fast auxiliary memories, e.g., the Cray-XMP with Solid-state Storage Device (SSD).

- In many cases the dominant cost is the matrix-by-vector product, and therefore the priority is to economize on the number of matrix-by-vector multiplications. The Lanczos-Galerkin process of Section 3 requires only one matrix-vector product (for computing the initial residual  $r_0^{(2)}$ ).

The Lanczos-Galerkin process was successfully used in the context of stiff ordinary differential equations [3]. There, it often is the case that the cost of a matrix-by-vector multiplication is very high and the Lanczos-Galerkin process becomes attractive.

It is important that the vectors  $v_i$ 's saved from the solution of the first linear system be orthogonal, since the Lanczos-Galerkin process is essentially based on the orthogonality of these vectors. Selective Orthogonalization [9] or Partial Orthogonalization [15] can both be used for that purpose alternatively to performing a full reorthogonalization at every step. Simon [15] has shown that any partial reorthogonalization that guarantees semiorthogonality, i.e., orthogonality within the square root of the machine unit round-off  $\epsilon_{\text{mach}}$ , will also deliver an approximate solution vector that is within  $\sqrt{\epsilon_{\text{mach}}}$  of the ideal solution vector from the Krylov subspace. These orthogonalization schemes can also be extended to the modified Lanczos method.

Numerical experiments with the Lanczos-Galerkin process for solving successive linear systems with slightly varying right-hand sides are described in [15]. It is shown there that the Lanczos-Galerkin procedure can be very effective in those situations. More recently, van der Vorst [16] took a different approach: The usual conjugate gradient formulation is adopted instead of the Lanczos formulation, and no reorthogonalization of any form is used. However, in order to account for the loss of orthogonality of the previous residual vectors, a modified formula is proposed for projecting the new residual on the previous Krylov subspace. The numerical experiments seem to indicate that this modified formula is less prone to numerical instability than the original one. There has been no experimentation of the modified Lanczos process for solving linear systems, but Carnoy and Geradin [1] reported successful tests with a variant of this method for solving generalized eigenvalue problems.

**Acknowledgments.** The author is indebted to Dr. H. D. Simon and to Professor B. N. Parlett for offering their helpful comments and suggestions and for having directed his attention to the topic of this paper.

Department of Computer Science  
Yale University  
New Haven, Connecticut 06510

1. E. CARNOY & M. GERADIN, *On the Practical Use of the Lanczos Algorithm in Finite Element Applications to Vibration and Bifurcation Problems*, Proc. Conf. on Matrix Pencils, held at Lulea, Sweden, March 1982 (Axel Ruhe, ed.), University of Umea, Springer-Verlag, New York, 1982, pp. 156–176.

2. R. CHANDRA, *Conjugate Gradient Methods for Partial Differential Equations*, Ph.D. Thesis, Computer Science Dept., Yale University, 1978.

3. W. C. GEAR & Y. SAAD, "Iterative solution of linear equations in ODE codes," *SIAM J. Sci. Statist. Comput.*, v. 4, 1983, pp. 583–601.

4. A. S. HOUSEHOLDER, *Theory of Matrices in Numerical Analysis*, Blaisdell, Johnson, Colo., 1964.

5. D. G. LUENBERGER, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass., 1965.
6. D. O'LEARY, "The block conjugate gradient algorithm and related methods," *Linear Algebra Appl.*, v. 29, 1980, pp. 243–322.
7. C. C. PAIGE & M. A. SAUNDERS, "Solution of sparse indefinite systems of linear equations," *SIAM J. Numer. Anal.*, v. 12, 1975, pp. 617–624.
8. B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N. J., 1980.
9. B. N. PARLETT, "A new look at the Lanczos algorithm for solving symmetric systems of linear equations," *Linear Algebra Appl.*, v. 29, 1980, pp. 323–346.
10. Y. SAAD, "Krylov subspace methods for solving large unsymmetric linear systems," *Math. Comp.*, v. 37, 1981, pp. 105–126.
11. Y. SAAD, "Practical use of some Krylov subspace methods for solving indefinite and unsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, v. 5, 1984, pp. 203–228.
12. Y. SAAD & A. SAMEH, *A Parallel Block Stiefel Method for Solving Positive Definite Systems*, Proceedings of the Elliptic Problem Solver Conference (M. H. Schultz, ed.), Los Alamos Scientific Laboratory, Academic Press, New York, 1980, pp. 405–412.
13. Y. SAAD & A. SAMEH, *Iterative Methods for the Solution of Elliptic Differential Equations on Multiprocessors*, Proceedings of the CONPAR 81 Conference (Wolfgang Handler, ed.), Springer-Verlag, New York, 1981, pp. 395–411.
14. Y. SAAD & M. H. SCHULTZ, "Conjugate gradient-like algorithms for solving nonsymmetric linear systems," *Math. Comp.*, v. 44, 1985, pp. 417–424.
15. H. D. SIMON, "The Lanczos algorithm with partial reorthogonalization," *Math. Comp.*, v. 42, 1984, pp. 115–142.
16. H. A. VAN DER VORST, *An Iterative Method for Solving  $f(A)x = b$  Using cg-Information Obtained for the Symmetric Positive Definite Matrix  $A$* , Technical Report 85-32, Delft University of Technology, Mathematics and Informatics, 1985.