NASA-TM-112076

RNR-88-008

THE IMPLEMENTATION OF A BLOCK LANCZOS ALGORITHM WITH REORTHOGONALIZATION METHODS

ROGER G. GRIMES*, JOHN G. LEWIS† AND HORST D. SIMON‡

Abstract. In this paper we describe a block Lanczos algorithm with partial reorthogonalization and external selective orthogonalization. The block Lanczos recurrence developed here is used as the key building block of a software package for the extraction of eigenvalues and eigenvector of large sparse symmetric generalized eigenproblems, particularly structural engineering applications. This paper describes the algorithmic details of our implementation, which includes a novel combination of several features that have only been investigated independently in the past: a block recurrence, partial reorthogonalization and various other reorthogonalizations to guarantee the orthogonality of the Lanczos vectors, the integration with an automatic shift strategy, and the unified treatment of generalized eigenvalue problems arising both in vibration and buckling problems.

Key Words. Lanczos algorithm, sparse eigenvalue problems, structural analysis, symmetric generalized eigenvalue problem, orthogonalization methods

AMS(MOS) subject classification. 65F15, 15A18, 65F50, 73K99

1. Introduction. The Lanczos algorithm [21] is a very powerful tool for extracting some of the extreme eigenvalues of a real symmetric matrix A, i.e. to find the largest and/or smallest eigenvalues and vectors of the eigenvalue problem

$$Ax = \lambda x.$$

Following Paige's thorough analysis of the roundoff properties of the Lanczos algorithm in finite precision [28], there has been a lively interest in this algorithm in the numerical analysis community and more recently also in various application areas such as structural engineering (see [5,20,30] and the references therein). Over the years several research codes have been published for the solution of sparse eigenvalue problems based on the Lanczos algorithm [6,12,22,27,34]. Each of these codes addresses some of the difficulties in implementing the Lanczos algorithm in finite precision arithmetic. However, none provides a complete solution for the difficult problems encountered in solving structural engineering problems in a production setting.

In order to apply the Lanczos algorithm to vibration and buckling problems in structural analysis, one must account for several difficulties which can arise in these problems:

1. multiple eigenvalues are a common occurrence
2. frequently there are significant input/output costs for accessing the matrices involved,

* Scientific Computing and Analysis Division, Boeing Computer Services, M/S 7L-21, Seattle, WA 98124

† Scientific Computing and Analysis Division, Boeing Computer Services, M/S 7L-21, Seattle, WA 98124

‡ Numerical Aerodynamic Simulation (NAS) Systems Division, NASA Ames Research Center, Mail Stop 258-5, Moffett Field, CA 94035. (The author is an employee of the SCA Division of Boeing Computer Services).

3. in these generalized eigenvalue problems, possibly one or both of the matrices are semidefinite; indefinite matrices also occur
4. usually the required eigenvalues are poorly separated

In order to deal with (1) and (2) efficiently, a *block* Lanczos algorithm is considered here. Block algorithms have been considered by a variety of researchers [4,22,15]; Scott's LASO code is based on a block algorithm as well [34]. A derivation of the block Lanczos algorithm and its general theory are given in §2.

Problems (3) and (4) require the use of a shifted and inverted operator. Scott's code leaves the decision for the shift and invert strategy completely to the user. The STLM code by Ericsson and Ruhe [10] is the first to implement an automatic shifting strategy for the vibration problem combined with the Lanczos algorithm. However, it does not handle multiple eigenvalues well [25], and it does not solve buckling problems. §3 explains how to form the shifted and inverted operators for both vibration and buckling analysis, and what modifications to the Lanczos recurrence result. One of the major problems is the fact that vectors have to be orthonormalized with respect to an inner product defined by a positive definite matrix M. A discussion of the issues associated with the M-orthonormalization of vectors, especially how to implement it in the context of the Lanczos algorithm is given in §4. The further precautions to allow cases where M induces only a semi-norm are also addressed in this section.

The block Lanczos recurrence by itself produces only a block tridiagonal matrix T. §5 describes how eigenvalue and vector approximations are computed from T and the Lanczos vectors. Some error bounds on these approximations are given as well.

In order to deal with the effects of finite precision arithmetic, several reorthogonalization schemes have been implemented for the block Lanczos algorithm. These schemes, together with some theoretical justification, are discussed in §6. §7 deals with the question of deciding when to terminate a Lanczos run. The performance of the block Lanczos recurrence in the context of an automated shift strategy, and coupled with efficient sparse matrix solver is discussed in §8.

The intent of this paper is to present the algorithms used in the implementation of the block Lanczos recurrence. The mathematical software aspects of the block Lanczos solver, such as the automatic shifting strategy, the optimal stopping heuristic for an individual Lanczos recurrence, and the integration with structures packages is discussed in detail in the companion report [18].

2. The Block Lanczos Recurrence for $Hx = \lambda x$. In order to simplify the discussion of the block Lanczos algorithm we consider the simple eigenvalue problem

$$Hx = \lambda x, \tag{1}$$

where H is a real symmetric linear operator. An important fact is that H does not need to be known explicitly for the application of the Lanczos algorithm. Spectral transformations typically give rise to problems where H is a product of sparse matrices and inverses of sparse matrices. However, neither the inverse matrices nor the product itself are actually computed. All that is required is a subroutine that computes Hy for a given vector y. This allows us to discuss the block Lanczos recurrence for a symmetric

operator H in all generality, whereas in the next section we will be more specific about the choice for H in the case of a vibration or buckling analysis.

The block Lanczos iteration with *blocksize* p for an $n \times n$ matrix H is given as follows:

Algorithm 1. Basic Block Lanczos Algorithm.

1) Initialization:
   Set $Q_0 = 0$,
   Set $B_1 = 0$,
   Choose R_1 and orthonormalize the columns of R_1 to obtain Q_1.
2) Lanczos Loop:
   For $j = 1, 2, 3 \ldots$ *maxstp* do

$$\begin{aligned} U_j &= HQ_j - Q_{j-1}B_j^T \\ A_j &= Q_j^T U_j \\ R_{j+1} &= U_j - Q_j A_j \\ Q_{j+1}B_{j+1} &= R_{j+1} \text{ (orthogonal factorization of } R_{j+1}) \end{aligned}$$

   End loop

The matrices Q_j, U_j, R_j for $j = 1, 2, \ldots$ are $n \times p$, A_j and B_j are $p \times p$, A_j is symmetric and B_j is upper triangular. The matrices Q_j have orthonormal columns. The columns of R_{j+1} are orthonormalized in the last step of the Lanczos loop. The orthonormalization procedure is discussed in more detail in §4. For the discussion here, it suffices that there always exists an upper triangular $p \times p$ matrix B_{j+1} and an $n \times p$ matrix Q_{j+1} with orthonormal columns so that $Q_{j+1}B_{j+1} = R_{j+1}$.

This formulation of the Lanczos loop is the one least susceptible to round off errors [29]. The order of the computation is, however, immaterial for the following analysis. Therefore, U_j and R_{j+1} can be eliminated from the Lanczos loop and the recurrence formula becomes

$$Q_{j+1}B_{j+1} = HQ_j - Q_jA_j - Q_{j-1}B_j^T. \tag{2}$$

We now show that the combined column vectors of the matrices $Q_1, Q_2, \ldots Q_j$, the so called *Lanczos vectors*, form an orthonormal set. This is a remarkable result, since the loop of the Lanczos algorithm does not involve explicit orthogonalizations to the earlier Lanczos vectors. This material is quite standard, but included here to make the discussion of reorthogonalization methods easier.

The column vectors of Q_1 are orthonormal by construction, and so are the columns of $Q_2, Q_3 \ldots$ etc. We only have to show that

$$Q_i^T Q_k = 0, \text{for } i \neq k.$$

Let us first show that $Q_j^T Q_{j+1} = 0$ and $Q_{j-1}^T Q_{j+1} = 0$ by induction over j. For $j = 1$:

$$\begin{aligned} Q_1^T Q_2 B_2 &= Q_1^T H Q_1 - Q_1^T Q_1 A_1 \\ &= A_1 - A_1 = 0. \end{aligned}$$

For $j \geq 2$, assume the result holds for all $i < j$:

$$\begin{aligned} Q_j^T Q_{j+1} B_{j+1} &= Q_j^T H Q_j - Q_j^T Q_j A_j - Q_j^T Q_{j-1} B_j^T \\ &= A_j - A_j = 0 \\ Q_{j-1}^T Q_{j+1} B_{j+1} &= Q_{j-1}^T H Q_j - Q_{j-1}^T Q_j A_j - Q_{j-1}^T Q_{j-1} B_j^T \\ &= Q_{j-1}^T H Q_j - B_j^T = 0, \end{aligned}$$

if $B_j = Q_j^T H Q_{j-1}$. But this follows by writing formula (2) for $i = j-1$ and premultiplying it by Q_j^T. This shows that the matrices Q_j are locally orthogonal. As a side result, $B_j = Q_j^T H Q_{j-1}$.

Finally, to prove global orthogonality, we show that

$$Q_i^T Q_{j+1} = 0 \ \ for \ \ i = 1, 2, \ldots j-1.$$

This is again shown by induction on j. First, by local orthogonality, $Q_2^T Q_1 = 0$. Assume then that it has been shown that $Q_i^T Q_j = 0$ for all $i = 1, 2, \ldots j-2$. It remains to be shown that $Q_i^T Q_{j+1} = 0$ for $i = 1, \ldots j-1$. Premultiply (2) by ${Q_i}^T$ to obtain:

$$\begin{aligned} Q_i^T Q_{j+1} B_{j+1} &= Q_i^T H Q_j - Q_i^T Q_j A_j - Q_i^T Q_{j-1} B_j^T \\ &= Q_i^T H Q_j \end{aligned}$$

Now equation (2) is transposed to substitute for $Q_i^T H$:

$$Q_i^T Q_{j+1} B_{j+1} = (Q_{i+1} B_{j+1} + Q_i A_i + Q_{i-1} B_i^T)^T Q_j = 0,$$

where by induction, the terms on the right all drop out.

Hence the Lanczos vectors form an orthonormal set of vectors. The computational efficiency of the Lanczos algorithm rests on the fact that these vectors can be computed with a simple recurrence and with a fixed amount of work per iteration step.

The blocks of Lanczos vectors can now be grouped together as the columns of an $n \times jp$ matrix $\mathcal{Q}_j$, where

$$\mathcal{Q}_j = [Q_1, Q_2, Q_3, \ldots Q_j].$$

Also define the block tridiagonal matrix T_j by

$$T_j = \begin{pmatrix} A_1 & B_2^T & 0 & \ldots & 0 \\ B_2 & A_2 & B_3^T & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & B_{j-1} & A_{j-1} & B_j^T \\ 0 & \ldots & 0 & B_j & A_j \end{pmatrix}. \tag{3}$$

T_j is a $jp \times jp$ matrix. Since the matrices B_j are upper triangular, T_j is a band matrix with half band width $p+1$ (rather than $2p$, if the B_j were full).

The first j instances of formula (2) can now be combined into a single formula:

$$H\mathcal{Q}_j = \mathcal{Q}_j T_j + Q_{j+1} B_{j+1} E_j^T. \tag{4}$$

Here E_j is an $n \times p$ matrix whose last $p \times p$ block is the $p \times p$ identity matrix and which is zero otherwise. Formula (4) is a compact way of expressing the Lanczos recurrence, and will be used throughout this discussion.

By premultiplying (4) by Q_j^T and using the orthogonality of the Lanczos vectors, it follows that

$$Q_j^T H Q_j = T_j.$$

Hence T_j is the orthogonal projection of H onto the subspace spanned by the columns of Q_j. For $p = 1$, this space is called the *Krylov subspace* $K_j(H; q_1)$. It can be shown by induction that

$$span(Q_j) = span(Q_1, HQ_1, H^2Q_2, \ldots H^{j-1}Q_1), \tag{5}$$

where *span (.)* denotes the subspace spanned by the columns of the matrices involved. In particular, it follows from (5) that for $p = 1$

$$K_i(H; q_1) = span(q_1, Hq_1, H^2q_1, \ldots H^{j-1}q_1).$$

From a different perspective, the (block) Lanczos algorithm is a method for constructing an orthonormal basis for the (block) Krylov subspace determined by H and Q_1. This basis of Lanczos vectors is distinguished by the fact that the orthogonal projection of H onto the (block) Krylov subspace is given by a (block) tridiagonal matrix. Hence the eigenvalues of T_j are the Rayleigh-Ritz approximations from $span(Q_j)$ to the eigenvalues of H. In addition, if s is an eigenvector of T_j, the vector $y = Q_j s$ is an approximate eigenvector of H. Viewed in this form, the Lanczos algorithm replaces a large and difficult eigenvalue problem involving H by a small and easy eigenvalue problem involving the block tridiagonal matrix T_j.

How good are the approximations obtained by solving the block tridiagonal eigenvalue problem involving the matrix T_j? An a posteriori bound on the residual can be obtained as follows: Let θ, s be an eigenpair for T_j, i.e.,

$$T_j s = s\theta$$

and let

$$y = Q_j s,$$

then

$$\begin{aligned} \|Hy - y\theta\| &= \|HQ_j s - Q_j s\theta\| \\ &= \|Q_j T_j s + Q_{j+1} B_{j+1} E_j^T s - Q_j s\theta\| \\ &= \|B_{j+1} E_j^T s\| = \|B_{j+1} s_j\|, \end{aligned} \tag{6}$$

where s_j are the last p components of the eigenvector s.

The quantity $\|B_{j+1} s_j\|$ can be computed without computing the approximate eigenvector y. Hence, with some modifications described in §5, (6) provides an inexpensive posteriori error bound.

Formula (6), however, does not guarantee that good approximations to eigenpairs will appear quickly. Such a priori estimates are provided by the Kaniel-Paige-Saad theory. For a detailed discussion, see references [30,33].

3. The Spectral Transformation Block Lanczos Algorithm . The eigenvalue problem in vibration analysis is given as

$$Kx = \lambda Mx, \tag{7}$$

where K is a symmetric matrix, and M is a positive semidefinite matrix. One might be tempted to reduce the generalized eigenvalue problem (7) to a standard eigenvalue problem and then apply the Lanczos algorithm. There are two good reasons not to do this: first, M cannot be factored if it is truly semidefinite, and second, the typical eigenvalue distribution for a vibration problem would make this approach very inefficient. Usually only the smallest eigenvalues of (7) are wanted. Although these lie at one end of the spectrum, they are commonly very poorly separated. A typical distribution of eigenvalues in a vibration problem would have the required eigenvalues of order 1, but the largest eigenvalue of the problem may be of order 10^8. The a priori estimates in [30] give the rate of convergence for computing the smallest eigenvalue to be bounded by $(1 - 10^{-8})$. This predicts very slow convergence which is observed in practice. This slow convergence necessitates the application of a shift and invert strategy.

Consider the problem

$$M\,(K\ -\ \sigma\,M)^{-1}\,Mx = \mu Mx, \tag{8}$$

where σ is a real parameter. Assume for the moment that M is positive definite; the complications introduced by a semidefinite M will be discussed later. It is easy to verify that (λ, x) is an eigenpair of (7) if and only if $(\frac{1}{\lambda-\sigma}, x)$ is an eigenpair of (8). Hence, the transformation of the eigenvalue problem from (7) to (8) does not change the eigenvectors, and the eigenvalues are related by

$$\mu = \frac{1}{\lambda - \sigma}. \tag{9}$$

This is the so called spectral transformation [11,35]. The advantage of applying the Lanczos algorithm to (8) instead to (7) becomes clear when the effect of the spectral transformation on the spectrum is considered. Figure 1 illustrates the transformation, where for simplicity the shift σ is taken at zero.

Note in Figure 1 that the desired eigenvalues λ_4 to λ_8, which are clustered and small, are transformed to μ_4 to μ_8, which are well separated and large. This spread of the eigenvalues ensures rapid convergence.

The primary price for this rapid convergence is the cost of a factorization of $K\ -\ \sigma M$. Of course, we never form the actual transformation $M\,(K\ -\ \sigma\,M)^{-1}\,M$ explicitly, because it is almost certainly a dense matrix. Instead the transformation is realized implicitly as a sequence of operations, computing MQ for a block of vectors Q, or solving the linear systems $(K\ -\ \sigma\,M)X = Q$. In practice, these operations are realized by separate subroutines. This modularity allows tuning the matrix factorization and multiplication routines to the class of problem under consideration.

An additional cost introduced by this spectral transformation is that, if anything, the problem moved farther from the standard form used in Algorithm 1. It is necessary

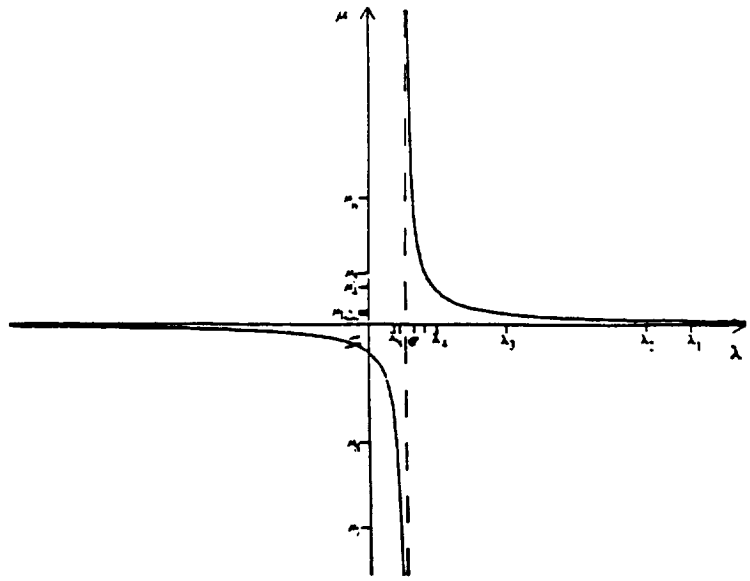

FIG. 1. *Vibration Spectral Transformation.*

now to consider the the Lanczos algorithm applied to a generalized symmetric eigenproblem, say $Hx = \lambda Mx$, where for the moment we assume M positive definite. Were we to reduce the problem to standard form by factoring M, the three term recurrence (2) would become

$$Q_{j+1}B_{j+1} = M^{-1/2}HM^{-1/2}Q_j - Q_jA_j - Q_{j-1}B_j^T. \tag{10}$$

If we premultiply (10) by $M^{1/2}$ and make the transformation of variables $\hat{Q}_j = M^{-1/2}Q_j$, (10) becomes

$$\begin{aligned} M\hat{Q}_{j+1}B_{j+1} &= M^{1/2}\, M^{-1/2}H\hat{Q}_j - M\hat{Q}_jA_j - M\hat{Q}_{j-1}B_j^T \\ &= H\hat{Q}_j - M\hat{Q}_jA_j - M\hat{Q}_{j-1}B_j^T. \end{aligned} \tag{11}$$

Again the value of the transformation may not be clear. The matrices $\hat{Q}_j$ are now M-orthogonal, rather than orthogonal, since $Q_j^TQ_j = I$ implies $\hat{Q}_j^TM\hat{Q}_j = I$. This is also a property of the eigenvectors x of this generalized eigenproblem. Whatever form is used for the Lanczos recurrence the approximate eigenvectors will be eventually be computed in the subspace $span(\hat{Q})$. M-orthogonality will in fact introduce difficulties in implementation, but the advantages of performing the recursion in the correct subspace are well documented in [35]. The Lanczos recurrence in this subspace is:

1) Initialization:
   Set $\hat{Q}_0 = 0$,
   Set $B_1 = 0$,
   Choose R_1 and M-orthonormalize the columns of R_1 to obtain $\hat{Q}_1$.
2) Lanczos Loop:
   For $j = 1, 2, 3 \ldots$ *maxstp* do
   $U_j \quad = H\hat{Q}_j - M\hat{Q}_{j-1}B_j^T$

$A_j \quad = \hat{Q}_j^T M U_j$
$W_{j+1} \quad = U_j - M\hat{Q}_j A_j$
$solve M R_{j+1} = W_{j+1}$
$\hat{Q}_{j+1} B_{j+1} \quad = R_{j+1}$ (M-orthogonal factorization of R_{j+1})
End loop

Here all matrices are dimensioned in the same way as in the basic block Lanczos algorithm given in the previous section. The matrix M is used at several occasions to assure the M-orthogonality of the Lanczos vectors, i.e., to assure that

$$Q_j^T M Q_j = I_{jp}. \tag{12}$$

Any symmetric positive definite matrix M can be used to define an inner product $x^T M y$ on the space of real n-vectors, which has the same properties as the ordinary Euclidean inner product $x^T y$. The Lanczos vectors are now M-orthogonal with respect to this inner product. This can be shown by an induction proof, which is identical to the one in §2, only with the M-inner product used at the corresponding locations.

For the moment it appears that $M^{-1/2}$ has disappeared from the standard recurrence, only to reappear in disguise as a solution operation. However, let us view (11) again, knowing that $H = M\ (K\ -\ \sigma\ M)^{-1}\ M$. Substituting for H gives:

$$M\hat{Q}_{j+1} B_{j+1} = M\ (K\ -\ \sigma\ M)^{-1}\ M\hat{Q}_j - M\hat{Q}_j A_j - M\hat{Q}_{j-1} B_j^T. \tag{13}$$

It is now evident that M appears in *all* of the terms in the recurrence. Formally we can premultiply (13) by M^{-1} to obtain a recurrence

$$\hat{Q}_{j+1} B_{j+1} = (K\ -\ \sigma\ M)^{-1}\ M\hat{Q}_j - \hat{Q}_j A_j - \hat{Q}_{j-1} B_j^T \tag{14}$$

in which only M, not M^{-1} appears. This, of course, has significant advantages in general, but in particular allows us to apply the same recurrence even if M is semi-definite. The justification for doing so appears later in this section.

At this point we shall drop the fiction of $\hat{Q}$. All operations will take place in this space, and we shall no longer bother putting 'hats' on the matrices. The actual Lanczos recurrence for solving (8) then becomes

Algorithm 2. Block Lanczos Algorithm for the Vibration Problem.

1) Initialization:
   Set $Q_0 = 0$,
   Set $B_1 = 0$,
   Choose R_1 and orthonormalize the columns of R_1 to obtain Q_1
   with $Q_1^T(MQ_1) = I_p$.
2) Lanczos Loop:
   For $j = 1, 2, 3 \ldots$ *maxstp* do
   $U_j \quad = (K\ -\ \sigma\ M)^{-1}\ (MQ_j) - Q_{j-1} B_j^T$
   $A_j \quad = U_j^T (MQ_j)$

$R_{j+1} = U_j - Q_j A_j$
Compute Q_{j+1} and (MQ_{j+1}) such that
a) $Q_{j+1} B_{j+1} = R_{j+1}$
b) $Q_{j+1}^T (MQ_{j+1}) = I_p$
End loop

It should be observed that the algorithm as formulated requires only one multiplication by M per step, to obtain (MQ_j). As noted, no factorization of M is required. However, the appropriate implementation of the last step of the Lanczos loop is not as obvious as it may seem. This *M-orthogonalization* of a set of p vectors will be discussed in more detail in §5.

For further reference, some of the key formulas of the basic Lanczos recurrence are listed here again for the spectral transformation block Lanczos algorithm. We have already used the analog of the ordinary three term recurrence in (14). Combining all j instances of (14) into one equation yields

$$(K - \sigma M)^{-1} M Q_j = Q_j T_j + Q_{j+1} B_{j+1} E_j^T, \tag{15}$$

where Q_j, T_j, and E_j are defined as in (4). Premultiplying (15) by $Q_j^T M$ and utilizing the M-orthogonality of the Lanczos vectors it follows that

$$Q_j^T M (K - \sigma M)^{-1} M Q_j = T_j.$$

Hence, T_j is the M-orthogonal projection of $(K - \sigma M)^{-1}$ onto the block Krylov subspace spanned by the columns of Q_j. The eigenvalues of T_j will approximate the eigenvalues of (8). If (s, θ) is an eigenpair of T_j, i.e.,

$$T_j s = s\theta,$$

then define $y = Q_j s$, and (y, θ) will be an approximate eigenpair of

$$M(K - \sigma M)^{-1} My = \mu My.$$

However, we are interested in eigenvalue approximations to the original problem (7) and not to the shifted and inverted problem (8). Formula (9) describes the relationship between the spectra of the two problems; if θ is an approximate eigenvalue of T_j (9) implies that

$$\nu = \sigma + \frac{1}{\theta} \tag{16}$$

is an approximate eigenvalue of (7). Since the spectral transformation does not change the eigenvectors, y is an approximate eigenvector for (7).

The a posteriori residual bound (6) does not generalize quite so cleanly. The computation analogous to (6) results in

$$(K - \sigma M)^{-1} My - y\theta = Q_{j+1} B_{j+1} E_j^T s. \tag{17}$$

For $\theta \neq 0$ it follows that

$$\frac{1}{\theta}\, My - (K - \sigma M)^{-1} y = \frac{1}{\theta}(K - \sigma M) Q_{j+1} B_{j+1} E_j^T s,$$

and

$$(K - \nu M) y = -\frac{1}{\theta}(K - \sigma M) Q_{j+1} B_{j+1} E_j^T. \tag{18}$$

The bounds in §5 will show that in case $\theta = 0$ or for very small θ, we have a very inaccurate approximation to an eigenpair. We therefore never expect to compute any residual bound in that case. The quantity on the right is computable without explicitly computing the eigenvector y, but only at the cost of a multiplication by $K - \sigma M$. This is not desirable because of the expense of such a multiplication, and more importantly, because $K - \sigma M$ is otherwise not used in the recurrence. (Only the factors of $K - \sigma M$ are assumed to be available.) In §5 we present a better way to obtain a residual bound.

The case of a semidefinite M, i.e., when there are vectors $z \neq 0$ such that

$$Mz = 0, \tag{19}$$

remains to be considered. The formulation of the block Lanczos algorithm for the vibration problem does not require the factorization of M. Hence the Lanczos algorithm can be applied in this case without further modifications. However, the eigenproblem (7) now has both finite and infinite eigenvalues. A direct modification leaves unanswered questions about the effect of the infinite eigenvalues on the convergence of the finite eigenvalues. Fortunately, we need only to make the obvious block modification of the analysis in [27] to remove the infinite eigenpairs from the recurrence. Following [27] the starting block for the Lanczos algorithm should computed as follows:

Algorithm 3. Computation of the Starting Block.

1) Choose $\tilde{R}_1$
2) Compute $R_1 = (K - \sigma M)^{-1} M \tilde{R}_1$
3) M-orthogonalize $R_1 = Q_1 B_0$

Here B_0 is an upper triangular $p \times p$ matrix, and Q_1 is then the M-orthogonal block of starting vectors.

It is shown in [9,18] that the eigenvectors corresponding to finite eigenvalues consist of a component orthogonal to the null vectors of M and a component in the nullspace of M. The second nullspace component is determined by an algebraic constraint from the non-nullspace component. Nour-Omid et. al. [27] show that all of the Lanczos vectors satisfy this constraint if the starting vectors are chosen as above. Hence, the Lanczos recurrence begins and remains entirely in the subspace spanned by the finite eigenvectors. To summarize the result, the effect of Algorithm 3 is such that infinite eigenvalues have no influence whatsoever on the block Lanczos algorithm in infinite precision. However, recently it has been shown, there is the possibility that infinite eigenvalue reappear in spite of Algorithm 3 in finite precision arithmetic [27]. Again, in

§5, we address how a final postprocessing step can purge the approximate eigenvectors of these components.

The final point to be discussed in this section is the implementation of the spectral transformation for the buckling problem

$$Kx = \lambda K_\delta x, \tag{20}$$

where K is the symmetric positive semidefinite stiffness matrix and K_δ is the symmetric differential stiffness matrix. Usually, the eigenvalues closest to 0 are wanted. Thus, a simple approach would be to interchange the roles of K and K_δ and to compute the largest eigenvalues of the problem

$$K_\delta x = \mu K x, \tag{21}$$

with $\mu = 1/\lambda$ by applying the simple Lanczos algorithm without shifts. This approach has three drawbacks: it requires the factorization of the possibly semidefinite matrix K, it does not allow for shifting, and the Lanczos vectors would not be in the same subspace as the approximate eigenvectors. A new spectral transformation avoids these problems. As before, the operator $K - \sigma K_\delta$ is factored, but the Lanczos recurrence is carried out using K orthogonality among the Lanczos vectors. This modification is easy to implement by replacing each multiplication by the mass matrix M in the vibration case with a multiplication by the stiffness matrix K in the buckling case; the rest of the recurrence remaining the same. Hence, in the buckling case, the shifted and inverted problem

$$K(K - \sigma K_\delta)^{-1} K x = \mu K x \tag{22}$$

is solved instead of the original problem (20). It is easy to check that (λ, x) is an eigenpair of (20) if and only if $(\frac{\lambda}{\lambda - \sigma}, x)$ is an eigenpair of (22). Hence the *buckling spectral transformation* does not change the eigenvectors, and the eigenvalues are related by

$$\mu = \frac{\lambda}{\lambda - \sigma}. \tag{23}$$

This buckling spectral transformation has essentially the same advantages as the vibration spectral transformation discussed earlier. Large eigenvalues of (20) are transformed to a cluster of eigenvalues of (22) near 1. Eigenvalues near the shift σ are transformed into large and well separated eigenvalues of (22), which are easily computed by the Lanczos algorithm. The main difference is that a shift at $\sigma = 0$ is not allowed, since all eigenvalues of (20) would be transformed to 1. Figure 2 shows the effect of the buckling spectral transformation.

Note in Figure 2 that the clustered eigenvalues from λ_4 to λ_8 which lie around σ are transformed to the well separated eigenvalues μ_4 to μ_8. This assures rapid convergence to these μ.

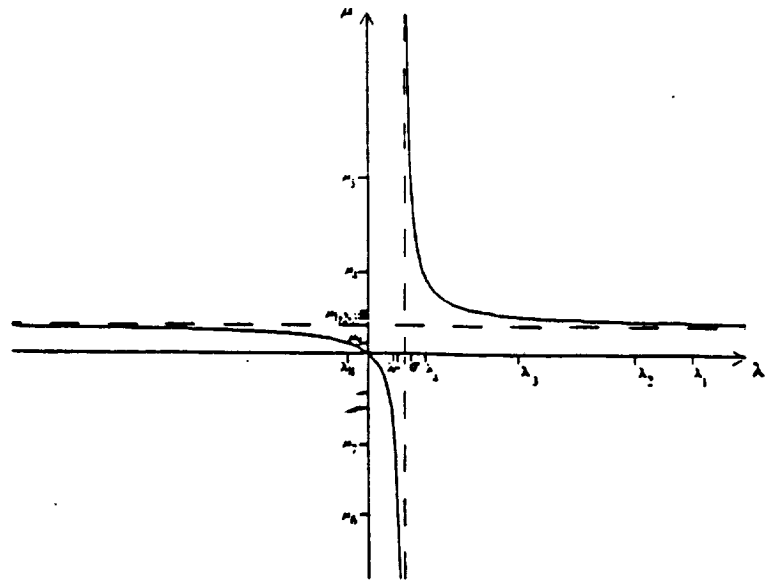

FIG. 2. *Buckling Spectral Transformation.*

Except for this different spectral transformation all other implementation details are the same for vibration and buckling analysis. In particular the issues involving the M-orthogonality of the Lanczos vectors apply equally to the K-orthogonal Lanczos vectors in the buckling case.

The eigenvalues of T_j constructed by the block Lanczos algorithm with buckling spectral transformation approximate the eigenvalues of (22). Hence, if (s, θ) is an eigenpair of T_j, i.e.

$$T_j s = s\theta,$$

then an approximate eigenpair (ν, y) of (20) is defined by

$$\nu = \frac{\sigma\theta}{\theta - 1}$$

and

$$y = Q_j s. \tag{24}$$

The eigenvectors y obtained by (24) form a K-orthonormal set.

Since the stiffness matrix K is used in the initialization phase in the same way as M in the vibration case, the sequence of Lanczos vectors will be orthogonal to the space spanned by the eigenvectors corresponding to zero eigenvalues. Hence T_j will contain no approximations to the exactly zero eigenvalues of K, which are also zero eigenvalues of (20). This is desirable since in the buckling analysis usually the first nonzero eigenvalue of (20) is wanted. The eigenvalues computed by the buckling spectral transformation Lanczos method are truly distinct from zero. Bounds on the residuals of approximate eigenpairs will be derived in §5.

4. The M-Orthogonal QR Factorization. Each step of the block Lanczos recurrence generates an $n \times p$ matrix R, whose column vectors are to be orthogonalized with respect to an inner product defined by a positive definite matrix M. Given R, we must compute Q and B such that

1) $R = QB$,
2) $Q^T MQ = I$,
3) Q is $n \times p$, and
4) B is $p \times p$ and upper triangular.

In the two standard engineering analyses M can be either the mass matrix (vibration analysis) or the stiffness matrix K (buckling analysis). Let M stand generically for the matrix inducing the norm in either of these problems. Although M may be positive semidefinite, it is sufficient here to consider only positive definite M by implicitly assuming that we work with the positive definite restriction of M on the orthogonal complement of the nullspace of M.

The algorithm that we use here to generate the matrices Q and B above is a generalization of the modified Gram-Schmidt process. The particular form we adopt here is chosen so that the matrix M is only accessed to form matrix-block products, never matrix-vector products. This greatly reduces cost in cases where multiplication by M is expensive, either when there are many nonzeros or when M is not stored in main memory. Let $r_1, r_2, \ldots r_p$ be the columns of R, $q_1, q_2, \ldots q_p$ the columns of Q and $\{b_{ij}\}$ be the entries of B, then the algorithm MMGS is:

Algorithm 4. M-orthogonal Modified Gram-Schmidt Orthogonalization (MMGS).

Initialization:

```
Given R = (r_1, r_2, ... r_p)
for    i = 1, 2, ... p do
       (mq)_i ← M r_i
end.
```

MMGS:

```
for i = 1, 2, ... p do
    b_ii ← r_i^T (mq)_i
    q_i ← r_i / b_ii
    (mq)_i ← (mq)_i / b_ii
    for j = i + 1, ... p do
        b_ij    ← q_i^T (mq)_j
        r_j     ← r_j / b_ij q_i
        (mq)_j ← (mq)_j / b_ij (mq)_i
    end
end
```

This algorithm utilizes an additional set of p auxiliary vectors $(mq)_i$, $i = 1, \ldots p$, which are initialized as Mr_i, $i = 1, \ldots p$. During the orthogonalization process these

vectors are updated together with the vectors $r_i = 1, \ldots p$. At completion of MMGS the vectors r_i, $i = 1, \ldots p$ have been overwritten with the desired vectors q_i, $i = 1, \ldots p$; further, the vectors $(mq)_i$ contain the vectors Mq_i. The vectors Mq_i are saved for use in the next iteration of the Lanczos algorithm. As stated only one matrix multiplication by M for a block of p vectors is required for this implementation of MMGS.

The basic algorithm MMGS must be modified to handle all special situations that can occur when orthogonalizing vectors produced by the block Lanczos algorithm. MMGS is actually embedded in an outer loop. The whole orthogonalization procedure is repeated if there is some indication that the vectors are not robustly orthogonal. Following [7] MMGS is repeated up to p times whenever the norm of one of the r_j after the orthogonalization is less than η times the norm of the original r_j . Currently $\eta = \sqrt{2}/2$. In this sense MMGS is an iterative procedure. The choice of η guarantees that an orthonormal set of vectors is obtained.

There are several situations where MMGS may encounter numerical difficulties. The Lanczos algorithm may produce a a rank deficient set of vectors r_j if the space of linearly independent vectors is exhausted, if the starting block contains a dependency by chance, or if the shift is very close to an eigenvalue. All three cases lead to rank deficiencies of different numerical character. Our implementation of MMGS detects these deficiencies and produces the appropriate response. For more details see [18].

5. Analysis of the Block Tridiagonal Matrix T_j. The large generalized eigenvalue problem is reduced by the block Lanczos algorithm to an eigenvalue problem of the form

$$T_j s = s\theta. \tag{25}$$

Here T_j is a block tridiagonal matrix as given in formula (3). Because B_i is an upper triangular matrix, T_j is a symmetric $jp \times jp$ band matrix of bandwidth $2p + 1$.

The eigenvalue problem for T_j is solved by first reducing T_j to tridiagonal form, and then by applying the tridiagonal QL algorithm. For the single vector Lanczos algorithm a very efficient implementation of the analysis of the tridiagonal matrix is given by Parlett and Nour-Omid [32]. Block generalizations of the approach in [32] have yet to be found; as a result we use a more straight forward approach. The implementations used here are slight modifications of corresponding subroutines in EISPACK [38,13]. Only the bottom p entries of the eigenvectors of T_j are needed for the evaluation of the residual bound (18) at any given block Lanczos step. Thus, it is unnecessary to compute and store the whole eigenvector matrix for each T_j. Instead only the last p components of this matrix are computed. This reduces considerably both computation and storage requirements for each Lanczos step. Only $p^2 j$ words are needed as opposed to $(pj)^2$ for the full eigenvector matrix.

The full eigenvectors of (25) are computed only at the very end of the Lanczos run. These are used to obtain approximate eigenvectors for the vibration and buckling analysis according to

$$y = Q_j s. \tag{26}$$

We used unmodified routines from EISPACK to obtain the full eigenvector matrix for T_j.

Eigenvectors are not needed during the Lanczos iteration, since error bounds on the accuracy of the computed eigenvalues can be computed without an explicit computation of the eigenvectors. These bounds are then used to determine which eigenvectors are accurate enough to be computed at the end of the Lanczos iteration. Let us now readdress the issue of evaluating the accuracy of intermediate approximations in the Lanczos algorithm without explicitly computing approximate eigenvectors of (7). Recall that for vibration analysis the following relation (17) holds:

$$(K - \sigma M)^{-1} My - y\theta = Q_{j+1} B_{j+1} E_j^T s$$

Therefore, since $Q_{j+1}^T M Q_{j+1} = I$:

$$\begin{aligned} \|M(K - \sigma M)^{-1} M - My\theta\|_{M^{-1}} &= \|M Q_{j+1} B_{j+1} E_j^T s\|_{M^{-1}} \\ &= \|B_{j+1} E_j^T s\| \equiv \beta_j. \end{aligned}$$

Note that β_j is easily computed for each eigenvector s, as the Euclidean norm of the product of the upper triangular matrix B_{j+1} with the last p components of s. We now apply a theorem on the error in eigenvalue approximations from [30] (pg. 318) to obtain:

$$|\frac{1}{\lambda - \sigma} - \theta| \leq \frac{\|M(K - \sigma M)^{-1} My - My\theta\|_{M^{-1}}}{\|My\|_{M^{-1}}} = \beta_j \tag{27}$$

Formula (27) gives a bound on how well the eigenvalues of T_j approximate the eigenvalues of the shifted and inverted operator. What is really needed is a bound on the error after backtransforming the eigenvalues according to (16), i.e., on $|\lambda - \nu|$.

Following [11], we use (27) to find

$$\begin{aligned} |\lambda - \nu| &= |\lambda - \sigma - \frac{1}{\theta}| \\ &= |\frac{1}{\theta}(\lambda - \sigma)(\frac{1}{\lambda - \sigma} - \theta)| \\ &\leq \frac{1}{|\theta|}|\lambda - \sigma|\beta_j = \frac{\beta_j}{\theta^2}. \end{aligned}$$

Hence

$$|\lambda - \nu| \leq \frac{\beta_j}{\theta^2}, \tag{28}$$

and (28) shows that we require only a moderately small β_j to guarantee a good approximate eigenvalue ν when θ is large, that is, for λ close to the shift σ.

The bound (28) can be improved for well separated eigenvalues. Define the gap γ as follows:

$$\gamma \equiv \min_{\lambda_i \neq \lambda} |\frac{1}{\lambda_i - \sigma} - \frac{1}{\lambda - \sigma}|, \tag{29}$$

Applying the same argument above to another theorem from [30](pg. 222) results in

$$|\lambda - \nu| \leq \frac{\beta_j^2}{\theta^2\gamma}. \tag{30}$$

The gap γ will be large if the eigenvalues are well separated, and (30) will usually be an improvement over (28). For clustered eigenvalues (28) will be better, hence in our implementation we use

$$|\lambda - \nu| \tilde{\leq} \min\{\frac{\beta_j}{\theta^2}, \frac{\beta_j^2}{\theta^2\gamma}\} \tag{31}$$

For multiple eigenvalues the definition of γ in (29) is modified; the gap between sets of multiple eigenvalues is used. In practice only an approximation to γ can be computed, which we derive from the shifted and inverted eigenvalues of T_j.

Similar error bounds can be derived for the buckling analysis. Let β_j be defined as before and define γ_b as

$$\gamma_b \equiv \min_{\lambda_i \neq \lambda} |\frac{\lambda_i}{\lambda_i - \sigma} - \frac{\lambda}{\lambda - \sigma}|, \tag{32}$$

Then one obtains

$$|\lambda - \nu| \leq \frac{|\sigma|\beta_j}{(\theta - 1)^2} \tag{33}$$

as the simple error bound, and

$$|\lambda - \nu| \leq \frac{|\sigma|\beta_j^2}{(\theta - 1)^2\gamma_b} \tag{34}$$

as the refined gap error bound. As in the vibration case, the minimum of (33) and (34) is chosen in the actual implementation.

Because the spectral transformation preserves,the eigenvectors, there is no immediate need to transform the approximate eigenvector y as computed by (26). However, we can find improved eigenvector approximations by using the so-called Ericsson-Ruhe correction [11]. This has the additional benefit [27] of purging from y any components in the directions of the infinite eigenvectors (or eigenvectors corresponding to zero eigenvalues in the buckling analysis). The correction ensures that the approximate eigenvectors are uncontaminated by the effects of a semidefinite M (K).

Let $\nu = \sigma + \frac{1}{\theta}$ be the computed eigenvalue, then we formally apply one step of inverse iteration with y

$$(K - \sigma M)\tilde{z} = My$$

and use (17) to obtain

$$\begin{aligned} \tilde{z} &= (K - \sigma M)^{-1} My \\ &= y\theta + Q_{j+1}B_{j+1}E_j^T s. \end{aligned}$$

The vector

$$z = \frac{1}{\theta}\tilde{z} = y + \frac{1}{\theta}Q_{j+1}B_{j+1}E_j^T s, \tag{35}$$

can be obtained easily by adding a linear combination of the next block of Lanczos vectors. This gives a better approximation to the eigenvector of the vibration problem very cheaply. The corresponding correction for the buckling analysis is given by

$$z = y + \frac{1}{\theta - 1}Q_{j+1}B_{j+1}E_j^T s.$$

6. Global Loss of Orthogonality and Reorthogonalization. So far the block Lanczos algorithm has been discussed in the context of exact arithmetic. Most of the relationships derived here also hold for the corresponding computed quantities of the Lanczos algorithm in finite precision arithmetic up to errors of about the round-off level of the machine. The notable exception is the global loss of orthogonality among the computed Lanczos vectors. According to (12) $Q_j^T M Q_j$ should equal I_{jp}. In reality, using finite precision arithmetic and no reorthogonalization, $Q_j^T M Q_j$ can be quite different from the identity matrix. Consider the finite precision Lanczos recurrence

$$Q_{j+1}B_{j+1} = K_\sigma M Q_j - Q_j A_j - Q_{j-1}B_j^T + F_j, \tag{36}$$

where F_j represents the roundoff error introduced at step j, and $K_\sigma \equiv (K - \sigma M)^{-1}$. Then for $k < j - 1$

$$\begin{aligned} Q_k^T M Q_{j+1} B_{j+1} &= Q_k^T M K_\sigma M Q_j - Q_k^T M Q_j A_j \\ &\quad - Q_k^T M Q_{j-1} B_j^T + Q_k^T M F_j. \end{aligned}$$

In an ideal setting all the quantities $Q_k^T M Q_j$ would be zero and drop out. However, under the influence of roundoff, these quantities cannot be ignored. Therefore we define

$$W_{jk} \equiv Q_k^T M Q_j,$$

and first obtain by premultiplying (36) by Q_k that

$$W_{j+1k}B_{j+1} = Q_k^T M K_\sigma M Q_j - W_{jk}A_j - W_{j-1k}B_j^T + Q_k^T M F_j. \tag{37}$$

Because K_σ is symmetric, we can obtain the transpose of $Q_k^T M K_\sigma M Q_j$ by premultiply the occurrence of (36) with $j = k$ by $Q_j^T M$:

$$Q_j^T M K_\sigma M Q_k = W_{jk+1}B_{k+1} + W_{kj}A_j - W_{k-1j}B_k^T + Q_j^T M F_k. \tag{38}$$

Substitute (38) into (37) to obtain:

$$\begin{aligned} W_{j+1k}B_{j+1} &= B_{k+1}^T W_{jk+1} + A_k W_{jk} + B_k W_{jk-1} \\ &\quad - W_{jk}A_j - W_{j-1k}B_j^T + G_{jk}. \end{aligned} \tag{39}$$

Here $G_{jk} \equiv Q_k^T M F_j - F_k^T M Q_j$ represents the local roundoff error. Formula (39) explains the global growth of the loss of orthogonality. We now use (39) to estimate the loss of orthogonality among the Lanczos vectors. Take norms in (39) to show that

$$\begin{aligned} \|W_{j+1k}\| \leq\ & \|B_{j+1}^{-1}\|(\|B_{k+1}\|\|W_{jk+1}\| \\ & +\|B_k\|\|W_{jk-1}\| + \|B_j\|\|W_{j-1k}\| \\ & +(\|A_j\| + \|A_k\|)\|W_{jk}\| + \|G_{jk}\|). \end{aligned} \tag{40}$$

This bound is simulated by the following recurrence, where ω_{jk} is a bound on $\|W_{jk}\|$:

Algorithm 5. Simulation of Loss of Orthogonality.

Initialize:

$\epsilon_s \equiv \epsilon p\sqrt{n}$, where $\epsilon \equiv$roundoff unit
and n = number of degrees of freedom

$\omega_{21} = \epsilon_s \bar{\beta}_2$

$\omega_{11} = \epsilon_s$

$\omega_{22} = \epsilon_s$

Loop:

for $j = 2, 3, \ldots$ maxstp do

$\omega_{j+1j+1} = \epsilon_s$

$\omega_{j+1j} = \epsilon_s$

for $k = 1, \ldots j-1$ do

$\omega_{j+1k} = \bar{\beta}_{j+1}(\beta_{k+1}\omega_{jk+1} + \beta_k\omega_{jk-1} + \beta_j\omega_{j-1k} + (\alpha_j + \alpha_k)\omega_{jk})$

end

end

In this algorithm

$$\begin{aligned} \alpha_k &\equiv \|A_k\| \\ \beta_k &\equiv \|B_k\| \\ \bar{\beta}_k &\equiv 1/\sigma_p(B_k), \text{where } \sigma_p(B_k) \text{ is the smallest singular value of } B_k. \end{aligned}$$

We take $\omega_{j+10} = 0$ and $\omega_{j+1-1} = 0$ where they occur. The ω_{jk} computed by Algorithm 5 simulate the bound given by (40). Previous results [37] show that such a simulation provides an order of magnitude estimate of growth of the loss of orthogonality.

The estimate of the loss of orthogonality obtained from the ω-recurrence is used to determine when reorthogonalization is necessary. Previous results [30,31,36] indicate that reorthogonalization should occur whenever

$$\max_k \omega_{j+1k} \geq \sqrt{\epsilon}. \tag{41}$$

The reorthogonalization should be carried out with both the block of vectors Q_j and Q_{j+1}. Hence, we have implemented the following algorithm for maintaining orthogonality, called *partial reorthogonalization (PRO)* [37]:

Algorithm 6. Partial Reorthogonalization.

At each Lanczos step, after computing Q_{j+1} and B_{j+1} do:

```
Update the ω-recurrence according to Algorithm 5;
ω_max ≡ max_k ω_{j+1k};
If ω_max ≥ √ε then
  for k = 1,...j − 1 do
      orthogonalize Q_j against Q_k
      orthogonalize Q_{j+1} against Q_k
  end;
  orthogonalize Q_{j+1} against Q_j
  update ω-recurrence:
      ω_{j+1k} = ω_{jk} = ε_s, k = 1,...j;
end if;
```

Note that the orthogonalization of Q_j and Q_{j+1} involves M-inner products. This requires the storage of both the Lanczos vectors and M·Lanczos vectors in secondary storage, or, alternatively, reapplying M to the Lanczos vectors. The appropriate form depends on cost.

A particular artifact of the block Lanczos algorithm appears in the computation of ω_{j+1j-1}. By Algorithm 5,

$$\omega_{j+1j-1} = \bar{\beta}_{j+1}(\beta_j\omega_{jj} + \beta_{j-1}\omega_{jj-2} + \beta_j\omega_{j-1j-1} + (\alpha_j + \alpha_{j-1})\omega_{jj-1}).$$

The leading term inside the parentheses on the right side is

$$\beta_j\bar{\beta}_j\epsilon_s = \kappa(B_j)\epsilon_s,$$

where $\kappa(B_j)$ is the condition number of B_j. The analysis of the ordinary Lanczos algorithm has unity corresponding to the term $\kappa(B_j)$. The loss of orthogonality occurs more rapidly in the block Lanczos algorithm, particularly when $\kappa(B_j)$ is significantly larger than one, but also in general. An inexpensive correction is suggested in [22]: at each step a *local reorthogonalization* between Q_{j+1} and Q_j is performed. This ensures that ϵ_s orthogonality holds between successive blocks of Lanczos vectors. Note that a local orthogonalization step is also performed on completion of a partial reorthogonalization.

The local reorthogonalization of block Q_{j+1} against Q_j assures that the Lanczos vectors locally are orthgononal to working precision. This is reflected in our choice of ϵ_s for the term ω_{j+1j} in Algorithm 5. Without local reorthgonalization this term would have to be set to $\epsilon_s\bar{\beta}j + 1\beta j$. The strong effect of the local orthogonality on the global loss of orthogonality for the block Lanczos recurrence has been observed by Lewis [22].

A different type of loss of orthogonality occurs in the context of the shifted and inverted Lanczos algorithm. It is possible that, after computing some eigenvalues with shift σ_1, the same eigenvalues and vectors are computed again when using σ_2. This presents a severe complication, requiring a mechanism for identifying duplicate copies from different runs. In addition, we waste resources recomputing eigenvectors. In order to avoid this complication and the duplicate computation we have implemented

another reorthogonalization scheme *external selective orthogonalization* (external SO). External SO is an efficient way of keeping the current sequence of Lanczos vectors orthogonal to previously computed eigenvectors, and thereby avoiding the recomputation of eigenvalues that are already known.

In theory it would be sufficient to orthogonalize the starting block against known eigenvectors, because this would guarantee that all subsequent Lanczos vectors are orthogonal as well. In practice, however, this does not hold. A global loss of orthogonality occurs, similar to the one among the Lanczos vectors themselves; in addition, the computed eigenvector is not exact. The contribution of both sources of error, which ultimately may lead to the recomputation of eigenvalues and vectors, is analyzed below.

Let (ν, y) be a computed eigenpair of (7). Premultiply (39) by $y^T M$ to obtain

$$\begin{aligned} y^T M Q_{j+1} B_{j+1} &= y^T M K_\sigma \; M Q_j \; - \; y^T M Q_j A_j \\ &\quad - y^T M Q_{j-1} B_j^T + y^T M F_j, \end{aligned} \tag{42}$$

which includes the effect of the local error term. Since (ν, y) is only an approximate eigenpair, it holds that

$$(K \; - \; \nu M) y = d \tag{43}$$

with d small. From (43) it follows that

$$(K \; - \; \sigma \; M)^{-1} M y = K_\sigma M y = \frac{1}{\nu \; - \; \sigma} y \; - \; K_\sigma d \tag{44}$$

We combine (44) and (42) to find

$$\begin{aligned} y^T M Q_{j+1} B_{j+1} &= \frac{1}{\nu \; - \; \sigma} y^T M Q_j \; - \; y^T M Q_j A_j \\ &\quad - y^T M Q_{j-1} B_j^T \; - \; d^T K_\sigma \; M Q_j + y^T M F_j. \end{aligned} \tag{45}$$

We take norms of both sides of (45) to compute a bound on the loss of orthogonality of the Lanczos vectors with respect to a previously computed eigenvector:

$$\|y^T M Q_{j+1}\| \leq \tilde{\beta}_{j+1} (\|(\nu \; - \; \sigma)^{-1} I - A_j\| \|y^T M Q_j\| + \beta_j \|y^T M Q_{j-1}\| + \delta_j),$$

where δ_j accounts for both the local roundoff term and the residual of the approximate eigenpair (ν, y). As with PRO, we define a recurrence relation for a quantity τ_j that estimates the loss of orthogonality of the Lanczos vectors with respect to y. In the recurrence τ_j is defined be:

$$\tau_{j+1} = \tilde{\beta}_{j+1} (\alpha_{\nu\sigma j} \tau_j + \beta_j \tau_{j-1} + \delta), \tag{46}$$

where we set initially $\tau_0 \equiv 0$ and $\tau_1 = \epsilon\sqrt{n}$. We set δ to the a posteriori error bound (31) or (33) and (34) obtained from the Lanczos run in which (ν, y) was computed. β_j and $\tilde{\beta}_{j+1}$ are defined as in Algorithm 5, and $\alpha_{\nu\sigma j} \equiv \|(\nu \; - \; \sigma)^{-1} I - A_j\|$.

If $\tau_{j+1} \geq \sqrt{\epsilon}$ then an external selective orthogonalization is performed in order to ensure that the sequence of Lanczos vectors remains orthogonal to working precision to the computed eigenvectors. External selective orthogonalization is implemented as follows:

Algorithm 7. External Selective Orthogonalization.
before the Lanczos iteration:
    determine the set of SO-vectors (eigenvectors for selective
        orthogonalization)
    orthogonalize Q_1 against the SO-vectors.

at each Lanczos step j do:
    update the τ-recurrence according to (46) for each SO-vector;
    if ($j = 2$ or τ_j has been greater than $\sqrt{\epsilon}$ or $\tau_{j+1} \geq \sqrt{\epsilon}$) then
        orthogonalize Q_{j+1} against y ;
        set $\tau_{j+1} = \epsilon_s$;
    end if

It is unnecessary to perform external SO against *all* previously computed eigenvectors. From (45) it is evident that one of the main driving forces in the loss of orthogonality is $(\nu - \sigma)^{-1}$. Loss of orthogonality will mostly occur in the direction of eigenvectors corresponding to eigenvalues close to the new shift. Other eigenvectors are unlikely to have any considerable impact on orthogonality. Further, only a few eigenvectors, again usually those close to the new shift, need be considered in order to avoid confusing new eigenvectors with old. In our implementation of external SO we choose a set of *sentinel eigenvalues* such that we can assume that all eigenvalues appear beyond the sentinels are already known. Such eigenvalues beyond the sentinels are discarded in the analysis of the block tridiagonal system. More details on the choice of sentinels are presented in [18]. The set of eigenvectors used for external SO becomes essentially the eigenvectors corresponding to the sentinels and those corresponding to any other known eigenvalues closer to the shift.

The orthogonalizations involve again both y and My. In order to avoid the repeated computation of My, all SO vectors are premultiplied by M and the result is stored on the same random access file as the eigenvectors y. This computation is performed before the actual Lanczos run begins. The actual interplay among the various orthogonalization schemes is discussed in detail in [18].

7. Cost Analysis and Termination of a Lanczos Run. The block Lanczos algorithm described above is part of a code which includes a shifting strategy [18], which chooses a sequence of shifts σ_i to efficiently compute the desired eigenvalues. In this environment we expect to make a number of Lanczos runs with different shifts. There are four ways in which a given Lanczos run can terminate without error. These are:

1. All required eigenvalues have converged. There is no need to compute further and control is returned to the shifting strategy.
2. Eigenvalues farther from the shift appear to be converging slowly. The estimated cost for computing them in the current Lanczos run is great enough that a new shift should be chosen.
3. Resources allocated to this run, either space or time, have been exhausted.
4. The B_{j+1}-block is ill-conditioned or singular. In this case a continuation of the Lanczos run is either numerically difficult or impossible. Singular or ill-conditioned B_{j+1}-blocks can be encountered for the following reasons:
   - The shift is very close to an eigenvalue.
   - The effective space of Lanczos vectors is exhausted — we cannot compute more orthogonal vectors than the problem has finite eigenvalues.
   - Dependencies within the starting block cause a singular B_{j+1} at some later stage.

All these conditions can be identified, and the appropriate measures taken to assure an orderly termination of a given Lanczos run.

The most common reason for termination is that computing more eigenvalues in the current run is inefficient. This decision to is based on a cost analysis that is carried out at each Lanczos step. The cost analysis assumes that a measure of the real user cost is available, which is used to monitor the cost of the various operations in the algorithm. This is used in a model of the Lanczos algorithm, where the observed changes in unconverged eigenvalues are used to estimate future convergence. Normally, eigenvalues far from the shift converge slowly and require a large number of steps. At the same time, partial reorthogonalization occurs primarily as a function of the nearness of other eigenvalues to the shift. Thus, PRO occurs in a predictable manner, with a cost increasing like $\mathcal{O}(j^3)$. Both the cost of PRO and of the matrix-block solves and multiplies will be large for distant eigenvalues. Our model attempts to locate a point in an individual run where the average cost per eigenvalue is minimized. This is a heuristic attempt to minimize the average cost for all eigenvalues. The effectiveness of the heuristic is demonstrated in [18], where more details are given.

8. Numerical Results. The intent of this section to provide some indication of the size and type of problems that can be solved with an effective block Lanczos code. Our emphasis here is on sparse eigenproblems, although we should note that our code has been used to solve large dense eigenvalue problems from quantum mechanics. Its effectiveness, as documented in [16], extends and verifies predictions by Paige [28] on the use of the Lanczos algorithm for dense eigenproblems.

Our code has been integrated into the structural analysis package MSC/NASTRAN and ported to a variety of computing environments using the NASTRAN internal linear equation solver. Some results specific to this environment are published elsewhere [17]. The results reported here were obtained using the SPARSPAK [14] general sparse linear equation solver and the multiple minimum degree reordering [26] on the Cray X-MP/24 at Boeing Computer Services. SPARSPAK on this machine has been modified for improved vectorization as described in [24]. Similar results using the same code and an

TABLE 1
Numerical Results on the Cray X-MP (execution times in seconds).

Problem Title	Order	Eigenproblem	Time
Platzman's small problem	362	164 e.v. in interval	6.85
Platzman's large problem	1919	648 e.v. in interval	83.10
Reactor Containment Floor	1922	200 smallest e.v.	19.45
Sports Arena	3562	10 smallest e.v.	8.43
767 Bulkhead	13992	one buckling mode	268.3
Columbia Center	15439	10 smallest e.v in a large cluster	242.8

SCS-40 are reported in [1,19].

Table 1 lists the performance of the block Lanczos code on the Cray X-MP/24 on several symmetric generalized eigenproblems. These problems are available as part of the Harwell-Boeing sparse matrix collection [8]. The numbers in Table 1 are most important when placed in historical perspective. Platzman's problems are finite difference discretizations of an oceanographic model for tidal movements. They were formulated in the mid 1970s by G. W. Platzman [3] and proved to very difficult eigenproblems at the time. One of their characteristics is that all of the eigenvalues appear as pairs; the other important characteristic is that the desired eigenvalues are not at either end of the spectrum. In [22] in 1976 the eigenvalues in the interval $[.0001, .24]$ were computed with great difficulty. The eigenvalues in the interval $[.000025, .0001]$ were also of interest but impossible to compute. The spectral transformation provides a powerful mechanism for extracting these interior eigenvalues. However, the STLM code published by Ericsson and Ruhe [10] is unable to cope with the multiplicities [25]. The block shifted Lanczos algorithm used only a few seconds of computer time and required no user intervention.

The next three problems are vibration analyses from structural engineering. The first of these demonstrates the value of shifting. The reactor containment floor problem was first solved in 1981 using an unshifted single Lanczos code [23]. This required over 900 seconds on a Cray-1S computer. The speed-up from hardware improvements is negligible compared to the effect of shifting. The Sports Arena also was solved in 1981 with the above mentioned code in 27 seconds. Shifting does not come into play because the number of eigenvalues required is small. Hence, more of the speed-up seen here can be attributed to hardware improvements, but there is still a clear effect from algorithmic improvements. The Columbia Center problem arose from a full scale finite element model of a 76 story skyscraper in Seattle. The model was abandoned in 1981 because its memory requirements were too large for the machines then available. It was solved for the first time in 1986 using the new algorithm on the X-MP. Unfortunately the finite element model was never completed by the engineers and the eigenvalue problem had 115 zero eigenvalues (rigid body modes). This modeling error proved to a remarkable demonstration of the robustness of the block shifted Lanczos code, which was able to compute all of the 115 zero eigenvalues.

TABLE 2
Historical Perspective on the Results in Table 1.

Problem Title	Year	History
Platzman's small problem	1976	all eigenvalues computed semiautomatically with extremely long runs[22]
	1983	STLM fails on multiple eigenvalues [25]
Platzman's large problem	1976	some eigenvalues computed semiautomatically
	1986	eigenvalues in [.000025, .0001] found for first time with new code
Reactor	1981	solved in $\approx$ 900 sec. on Cray-1S using code in [23]
Sports Arena	1981	solved in 27.1 sec. on Cray-1S using code in [23]
767 Bulkhead	1985	new code uncovered modeling error
Columbia Center	1981	large scale model abandoned
	1986	abandoned model solved, multiplicity of 115 for zero eigenvalue correctly determined

The 767 bulkhead problem is an example of a buckling analysis. Only one eigenpair, the fundamental mode, was needed. The eigenanalysis was used successfully to find an error in a finite element model, even though the problem posed extreme numerical difficulties, with the fundamental eigenvalue at rounding level compared to the largest eigenvalue of the problem. The alternative mechanisms for locating the engineering error in this model would have required a difficult and painstaking engineering review, and a great deal of engineering time.

Three of these problems are used to illustrate the distribution of computational resources. The computationally intensive parts of the algorithm can be any of: the sparse matrix reordering; matrix factorization, solution, and multiplication; the block tridiagonal matrix analyze step; and the orthogonalization steps. Except for the block tridiagonal matrix analyze step, our code is based on computational kernels in VectorPak [2], a package of highly optimized computational kernels for scientific and engineering applications on Cray computers.

A breakdown of the percentage of execution time spent in various parts of the code is given in Table 3. Entries of less than 5% are not included in the table. These results are typical of large problems in that most of the execution time is spent in sparse matrix factorization and solve operations. Orthogonalization and block tridiagonal analysis are only significant if the individual Lanczos runs are long and many eigenvalues are required as in the reactor problem (which is also relatively small). The 767 bulkhead problem illustrates that matrix multiplication can play a major role in buckling problems. The buckling spectral transformation requires repeated multiplications with the stiffness matrix, which is generally less sparse than the mass matrices of vibration analysis. We conclude that the current version of the block Lanczos algorithm has reached a very satisfactory performance, and that further algorithmic performance improvements for the dynamic analysis of large structures will be derived from more efficient sparse

TABLE 3
Distribution of Execution Time

Problem	Order	Factor	Solve	Mult.	Eigen Analysis	Orthogo-nalization
Reactor		10%	37%		30%	5%
767	10%	52%		23%		
Columbia	5%	42%	40%			

matrix algorithms for ordering, factoring, and solving large linear systems on advanced architecture computers.

9. Summary. Several important results about the Lanczos algorithm have been combined for the first time in a production code for solving large sparse eigenvalue problems. The synergism of these results leads to an efficient and robust implementation, which can solve previously intractable problems. The block shifted Lanczos algorithm, coupled with the speed of a supercomputer such as the Cray X-MP, provides a very effective solution of large symmetric generalized eigenproblems for sparse matrices.

REFERENCES

[1] R. ANDERSON, R. GRIMES, R. RIEBMAN, AND H. SIMON, *Early experience with the SCS-40*, Supercomputer, 22 (1987), pp. 34 – 44.

[2] BOEING COMPUTER SERVICES, *VectorPak Users Manual*, 1987. Document No. 20460-0501-R1.

[3] A. CLINE, G. GOLUB, AND G. PLATZMAN, *Calculations of normal modes of oceans using a Lanczos method*, in Sparse Matrix Computations, J. Bunch and D. Rose, eds., Academic Press, 1976, pp. 409 – 426.

[4] J. CULLUM AND W. E. DONATH, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large sparse real symmetric matrices*, in Proc. of the 1974 IEEE Conference on Decision and Control, Phoenix, 1974.

[5] J. CULLUM AND R. WILLOUGHBY, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 1 Theory*, Birkhäuser, Boston, 1985.

[6] ——, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 2 Users Guide*, Birkhäuser, Boston, 1985.

[7] J. W. DAVID, W. B. GRAGG, L. KAUFMAN, AND G. W. STEWART, *Reorthogonalizations and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772 – 795.

[8] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse Matrix Test Problems*, Tech. Rep. CSS 191, Harwell Laboratory, Didcot, Oxon, England, 1987.

[9] T. ERICSSON, *A generalized eigenvalue problem and the Lanczos algorithm*, in Large Scale Eigenvalue Problems, J. Cullum and R. Willoughby, eds., North-Holland, 1986.

[10] ——, *User Guide for STLM*, Tech. Rep. UMINF-96.82, University of Umea, 1982.

[11] T. ERICSSON AND A. RUHE, *The spectral transformation Lanczos method*, Math. Comp., 34 (1980), pp. 1251 – 1268.

[12] ——, *STLM - a Software Package for the Spectral Transformation Lanczos Algorithm*, Tech. Rep. UMINF-101.82, University of Umea, Sweden, 1982.

[13] B. S. GARBOW, J. M. BOYLE, J. J. DONGARRA, AND C. B. MOLER, *Matrix Eigensystem Routines - EISPACK Guide Extension*, vol. 51 of Lecture Notes in Computer Sciences, Springer-Verlag, Berlin, 1977.

[14] A. George and J. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, Englewood Cliffs, 1981.

[15] G. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. Rice, ed., Academic Press, New York, 1977.

[16] R. Grimes, H. Krakauer, J. Lewis, H. Simon, and S. Wei, *The Solution of Large Dense Generalized Eigenvalue Problems on the Cray X-MP/24 with SSD*, J. Comp. Physics, 69 (1987), pp. 471 – 481.

[17] R. Grimes, J. Lewis, L. Komzsik, D. Scott, and H. Simon, *Shifted block Lanczos algorithm in MSC/NASTRAN*, in Proceedings of the MSC/NASTRAN User's Conference, Los Angeles, 1985.

[18] R. Grimes, J. Lewis, and H. Simon, *The Implementation of a Block Shifted and Inverted Lanczos Algorithm for Eigenvalue Problems in Structural Engineering*, Tech. Rep. ETA-TR-39, Boeing Computer Services, August 1986.

[19] R. Grimes and H. Simon, *Dynamic analysis with the Lanczos algorithm on the SCS-40*, in Proceedings of the Second International Conference on Supercomputing, Santa Clara, California, 1987, pp. 110 – 118.

[20] T. Hughes, *The Finite Element Method*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987, ch. The Lanczos Algorithm for Solution of Large Generalized Eigenproblems. chapter authored by Bahram Nour-Omid.

[21] C. Lanczos, *An Iteration Method for the Solution of Eigenvalue Problem of Linear Differential and Integral Operators*, J. Res. Nat. Bur. Stand., 45 (1950), p. 255.

[22] J. Lewis, *Algorithms for Sparse Matrix Eigenvalue Problems*, PhD thesis, Stanford University, Dept. of Computer Science, 1977.

[23] J. Lewis and R. Grimes, *Practical Lanczos algorithms for solving structural engineering eigenvalue problems*, in Sparse Matrices and their Uses, I. Duff, ed., Academic Press, London, 1981.

[24] J. Lewis and H. Simon, *The impact of hardware gather/scatter on sparse Gaussian elimination*, SIAM J. Sci. Stat. Comp., 9 (1988), pp. 304 – 311.

[25] J. Lewis and H. Simon, *Numerical Experience with the Spectral Transformation Lanczos Method*, Tech. Rep. ETA-TR-16, Boeing Computer Services, 1983.

[26] J. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Transactions on Mathematical Software, 11 (1985), pp. 141 – 153.

[27] B. Nour-Omid, B. N. Parlett, T. Ericson, and P. S. Jensen, *How to implement the spectral transformation*, Math. Comp., 48 (1987), pp. 663 – 673.

[28] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, PhD thesis, University of London, 1971.

[29] ——, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341 – 349.

[30] B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, New Jersey, 1980.

[31] B. Parlett and D. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217 – 238.

[32] B. N. Parlett and B. Nour-Omid, *The use of refined error bounds when updating eigenvalues of a growing symmetric tridiagonal matrix*, Linear Algebra and its Applications, 68 (1985), pp. 179 – 219.

[33] Y. Saad, *On the rates of convergence of the Lanczos and block-Lanczos methods*, SIAM J. Num. Anal., 17 (1980), pp. 687 – 706.

[34] D. Scott, *Block Lanczos Software for Symmetric Eigenvalue Problems*, Tech. Rep. ORNL/CSD 48, Oak Ridge Nat. Lab., 1979.

[35] D. S. Scott, *The advantages of inverted operators in Rayleigh-Ritz approximations*, SIAM J. Sci. Stat. Computing, 3 (1982), pp. 68 – 75.

[36] H. Simon, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Lin. Alg. Appl., 61 (1984), pp. 101 – 131.

[37] ——, *The Lanczos algorithm with partial reorthogonalization*, Mathematics of Computation, 42 (1984), pp. 115 – 136.

[38] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines - EISPACK Guide*, vol. 6 of Lecture Notes in Computer Sciences, Springer-Verlag, Berlin, 1976.