MATHEMATICS OF COMPUTATION
VOLUME 48, NUMBER 178
APRIL 1987, PAGES 663-673

How to Implement the Spectral Transformation*

By Bahram Nour-Omid**, Beresford N. Parlett,
Thomas Ericsson**, and Paul S. Jensen

Abstract. The general, linear eigenvalue equations $(\mathbf{H} - \lambda\mathbf{M})\mathbf{z} = 0$, where $\mathbf{H}$ and $\mathbf{M}$ are real symmetric matrices with $\mathbf{M}$ positive semidefinite, must be transformed if the Lanczos algorithm is to be used to compute eigenpairs $(\lambda, \mathbf{z})$. When the matrices are large and sparse (but not diagonal) some factorization must be performed as part of the transformation step. If we are interested in only a few eigenvalues λ near a specified shift, then the spectral transformation of Ericsson and Ruhe [1] proved itself much superior to traditional methods of reduction.

The purpose of this note is to show that a small variant of the spectral transformation is preferable in all respects. Perhaps the lack of symmetry in our formulation deterred previous investigators from choosing it. It arises in the use of inverse iteration.

A second goal is to introduce a systematic modification of the computed Ritz vectors, which improves the accuracy when $\mathbf{M}$ is ill-conditioned or singular.

We confine our attention to the simple Lanczos algorithm, although the first two sections apply directly to the block algorithms as well.

1. Overview. This contribution is an addendum to the paper by Ericsson and Ruhe [1] and also [7]. The value of the spectral transformation is reiterated in a later section. Here we outline our implementation of this transformation.

The equation to be solved, for an eigenvalue λ and eigenvector $\mathbf{z}$, is

$$(\mathbf{H} - \lambda\mathbf{M})\mathbf{z} = 0, \tag{1}$$

$\mathbf{H}$ and $\mathbf{M}$ are real symmetric $n \times n$ matrices, and $\mathbf{M}$ is positive semidefinite. A practical instance of (1) occurs in dynamic analysis of structures, where $\mathbf{H}$ and $\mathbf{M}$ are the stiffness and mass matrices, respectively. We assume that a linear combination of $\mathbf{H}$ and $\mathbf{M}$ is positive definite. It then follows that all eigenvalues λ are real. In addition, one has a real scalar σ, distinct from any eigenvalue, and we seek a few eigenvalues λ close to σ, together with their eigenvectors $\mathbf{z}$. Ericsson and Ruhe replace (1) by a standard eigenvalue equation

$$\left[\mathbf{C}(\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{C}^{\mathrm{T}} - \nu\mathbf{I}\right]\mathbf{y} = 0, \tag{2}$$

where $\mathbf{C}$ is the Choleski factor of $\mathbf{M}$; $\mathbf{M} = \mathbf{C}^{\mathrm{T}}\mathbf{C}$ and $\mathbf{y} = \mathbf{C}\mathbf{z}$. If $\mathbf{M}$ is singular then so is $\mathbf{C}$, but fortunately the eigenvector $\mathbf{z}$ can be recovered from $\mathbf{y}$ via $\mathbf{z} = (\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{y}$. Of course, there is no intention to invert $(\mathbf{H} - \sigma\mathbf{M})$ explicitly. The

Received May 14, 1984; revised December 20, 1985.

1980 *Mathematics Subject Classification*. Primary 65F15.

*This research was supported in part by the AFOSR contract F49620-84-C-0090. The third author was also supported in part by the Swedish Natural Science Research Council.

**The paper was written while this author was visiting the Center for Pure and Applied Mathematics, University of California, Berkeley, California 94720.

operator given to the Lanczos program is $\mathbf{A} = \mathbf{C}(\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{C}^{\mathrm{T}}$. The spectrum of $\mathbf{A}$ is related to the original spectrum by

$$\nu = \frac{1}{\lambda - \sigma}, \tag{3}$$

and so it is the eigenvalues of $\mathbf{A}$ closest to $\pm\infty$ which must be computed.

In contrast to (2), we prefer to change (1) into

$$\left[(\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M} - \nu\mathbf{I}\right]\mathbf{z} = 0. \tag{4}$$

Our operator $\mathbf{B} = (\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}$ is not symmetric, but it is selfadjoint with respect to the semi-inner product defined by $\mathbf{M}$. At first sight it appears to be extravagant to work with the $\mathbf{M}$-inner product, but it is not. Our investigation suggests that there is no trade-off. Reduction (4) is no worse than (2), and is sometimes better, with respect to storage, arithmetic effort, and vectorizability. In fact, $\mathbf{B}$ occurs naturally in the setting of Subspace Iteration methods, see [3]. It is only in the Lanczos context that it has been overlooked.

Section 2 examines the important case of singular $\mathbf{M}$. Sections 3 and 4 look in detail at the two reductions. Section 5 extols the spectral transformation (3), but with more arguments than were given in [1]. Section 6 shows that the tridiagonal $\mathbf{T}$ is not quite the projection that we want. The notation follows Ericsson and Ruhe [1] and Parlett [5]. Some familiarity with the simple Lanczos algorithm is assumed.

2. Singular M. This case is merely the extreme point of the set of problems in which $\mathbf{M}$ becomes increasingly ill-conditioned. There is no sharp break in behavior when $\mathbf{M}$ becomes singular and, in fact, the situation is easier to describe.

The main point is that there is no intrinsic mathematical difficulty here; no hidden pathology. The troubles that beset certain algorithms arise simply from our yearning for efficiency. We begin by describing the geometry of the situation because, to our knowledge, such a description is not readily available. Next we turn to the Lanczos algorithm and make four points:

(a) The starting vector must be put into the proper subspace.

(b) There is a simple recurrence that governs the angle separating the Lanczos vectors from this subspace. Usually the recurrence is unstable, but the growth in these angles is invisible when the usual $\mathbf{M}$-inner product is used.

(c) The Lanczos vectors can be projected back into the proper subspace, when necessary, but at substantial cost.

(d) There is an inexpensive modification to computed eigenvectors that purges unwanted components in the null space of $\mathbf{M}$.

2.1. *The Geometric Picture*. The pair $(\mathbf{H}, \mathbf{M})$ is assumed to be definite, so there is no loss in generality in taking $\mathbf{H}$ itself to be positive definite. For any matrix $\mathbf{X}$ let $n(\mathbf{X})$ denote its null space and $r(\mathbf{X})$ its range (or column space). Recall that $\mathbf{B} = (\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}$. Clearly, $n(\mathbf{B}) = n(\mathbf{M}) \neq \{0\}$. Now

(1) $r(\mathbf{B})$ and $n(\mathbf{B})$ are each invariant under $\mathbf{B}$, i.e., $\mathbf{B}n(\mathbf{B}) \subset n(\mathbf{B})$, $\mathbf{B}r(\mathbf{B}) \subset r(\mathbf{B})$.

(2) $\mathbf{u} \in r(\mathbf{B})$ and $\mathbf{z} \in n(\mathbf{B})$ implies that $\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{u} = 0$; i.e., $r(\mathbf{B}) \perp_{\mathbf{H}} n(\mathbf{B})$.

Proof. Let $0 \neq \mathbf{z} \in n(\mathbf{B})$ and $\mathbf{u}\ (= \mathbf{B}\mathbf{x}) \in r(\mathbf{B})$. Then, by definition of $\mathbf{B}$, $\mathbf{H}\mathbf{u} = \sigma\mathbf{M}\mathbf{u} + \mathbf{M}\mathbf{x}$. Premultiply by $\mathbf{z}^{\mathrm{T}}$ to find $\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{u} = \sigma\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{u} + \mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{x} = 0$. Here, we use the fact $\mathbf{z}^{\mathrm{T}}\mathbf{M} = 0^{\mathrm{T}}$.

(3) $\mathbf{R}^n = r(\mathbf{B}) \oplus n(\mathbf{B})$. This follows from the fundamental theorem of linear algebra; $n = \text{rank} + \text{nullity}$.

The *oblique* projection of $\mathbf{R}^n$ in (3) is the relevant one for this problem. All the eigenvectors belonging to finite eigenvalues are in $r(\mathbf{B})$. This is the good subspace. Note that $r(\mathbf{B})$ is not invariant under $\mathbf{M}$. $r(\mathbf{B})$ *is not orthogonal to* $n(\mathbf{M})$ (in the Euclidean sense).

Example.

$$\mathbf{H} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \sigma = 0.$$

$$n(\mathbf{M}) = n(\mathbf{B}) = \text{span}\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad r(\mathbf{M}) = \text{span}\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad r(\mathbf{B}) = \text{span}\begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

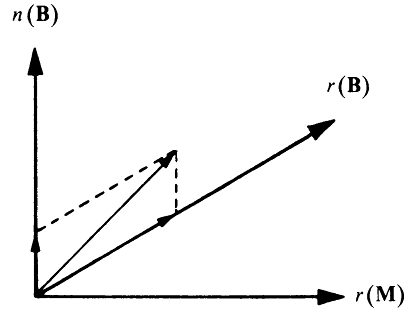

FIGURE 1
Geometric representation of the subspaces

The example confirms that the desired eigenvectors are not orthogonal to $n(\mathbf{M})$ in the Euclidean sense. In general, it is difficult to tell whether or not a vector is in $r(\mathbf{B})$. The next result yields a test.

THEOREM. $\mathbf{H}r(\mathbf{B}) = r(\mathbf{M})$.

Proof. Let $\mathbf{u}$ $(= \mathbf{Bx}) \in r(\mathbf{B})$. From the proof of property (2) above one has $\mathbf{Hu} = \mathbf{M}(\sigma\mathbf{u} + \mathbf{x}) \in r(\mathbf{M})$. Thus $\mathbf{H}r(\mathbf{B}) \subset r(\mathbf{M})$. Since $\mathbf{H} - \sigma\mathbf{M}$ is invertible, $\dim(r(\mathbf{B})) = \dim(r(\mathbf{M})) = \text{rank}(\mathbf{M})$. Finally, since $\mathbf{H}$ is invertible, $\dim(\mathbf{H}r(\mathbf{B})) = \dim(r(\mathbf{B})) = \dim(r(\mathbf{M}))$. Q.E.D.

When $\mathbf{M}$ is diagonal then $n(\mathbf{M})$ is known and $\mathbf{q} \in r(\mathbf{B})$ if $\mathbf{Hq} \perp n(\mathbf{M})$. In other words, $\mathbf{Hq}$ must have zeros in the appropriate elements. Unfortunately, the test is not cheap.

2.2. *The Starting Vector.* It is not appropriate to start the Lanczos process from a random vector in $\mathbf{R}^n$. It should be confined to $r(\mathbf{B})$. In exact arithmetic the *whole* Krylov subspace spanned by $\mathbf{q}_1, \mathbf{Bq}_1, \mathbf{B}^2\mathbf{q}_1, \ldots$ will then be in $r(\mathbf{B})$.

If $\mathbf{q}_1 \notin r(\mathbf{B})$, then all computed Ritz vectors with significant components in $\mathbf{q}_1$ will contain unwanted components in $n(\mathbf{B})$. The usual way to enforce $\mathbf{q}_1 \in r(\mathbf{B})$ is to apply $\mathbf{B}$ to a random vector in $\mathbf{R}^n$. This increases the cost of the starting vector, but this is negligible relative to the total computation. Unfortunately, roundoff error drives later Lanczos vectors away from $r(\mathbf{B})$.

2.3. *The Growth in Unwanted Components.* Let $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots\}$ be the computed Lanczos vectors. Let $\mathbf{z}$ be any fixed vector in $n(\mathbf{M})$ with $\|\mathbf{z}\|_{\mathbf{H}} = 1$. Let $\tau_i = \mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{q}_i$. Since $\|\mathbf{q}_i\|_{\mathbf{H}} \neq \|\mathbf{q}_i\|_{\mathbf{M}} = 1$, the τ_i are not true cosines of the angles between $\mathbf{q}_i$ and $n(\mathbf{M})$. However, $|\tau_i|$ is the length of the projection of $\mathbf{q}_i$ onto $\mathbf{z}$ in the $\mathbf{H}$-norm. Recall that

$$\mathbf{q}_{j+1}\beta_{j+1} = \mathbf{B}\mathbf{q}_j - \mathbf{q}_j\alpha_j - \mathbf{q}_{j-1}\beta_j + \mathbf{f}_j,$$

where $\mathbf{f}_j$ is a roundoff quantity. Premultiply by $\mathbf{z}^{\mathrm{T}}\mathbf{H}$ to find

$$\tau_{j+1}\beta_{j+1} = \mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{B}\mathbf{q}_j - \tau_j\alpha_j - \tau_{j-1}\beta_j + \mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{f}_j.$$

By property (2) $\mathbf{z} \perp_{\mathbf{H}} r(\mathbf{B})$ and so

$$\tau_{j+1} = -\left(\alpha_j\tau_j + \beta_j\tau_{j-1} + \mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{f}_j\right)/\beta_{j+1}.$$

There is nothing to stop the τ_j from growing steadily. However, $\|\mathbf{q}_j + \rho\mathbf{z}\|_{\mathbf{M}} = \|\mathbf{q}_j\|_{\mathbf{M}} = 1$ for all ρ, and so this growth is not visible in the standard implementation of the Lanczos algorithm.

2.4. *Projection of Lanczos Vectors.* The matrix that projects onto $r(\mathbf{B})$ orthogonal to $n(\mathbf{M})$ in the $\mathbf{H}$-norm is $\mathbf{I} - \mathbf{N}(\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N})^{-1}\mathbf{N}^{\mathrm{T}}\mathbf{H}$, where the columns of $\mathbf{N}$ form a basis for $n(\mathbf{B})$. $\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N}$ is invertible since $\mathbf{H}$ is positive definite. It is possible to compute $\mathbf{N}$ before beginning a Lanczos run. When $\mathbf{M}$ is diagonal then $\mathbf{N}$ is composed of certain columns of the identity matrix.

At the end of each step of the Lanczos algorithm one has only to form $\tilde{\mathbf{q}}_{j+1} = \mathbf{q}_{j+1} - \mathbf{N}(\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N})^{-1}\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{q}_{j+1}$. The matrix $\mathbf{G}^{\mathrm{T}} = (\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N})^{-1}\mathbf{N}^{\mathrm{T}}\mathbf{H}$ may be formed *before* the start of a Lanczos run. In this way, the extra cost is l dot products and l vector combinations per step. When $\mathbf{M}$ is diagonal and l is small, this arithmetic cost is modest. The extra storage is less acceptable. We do not use this modification.

2.5. *Purification of Computed Eigenvectors.* A simple way to restore vectors to $r(\mathbf{B})$ is to apply $\mathbf{B}$ to them. However, one goal in using the Lanczos algorithm is to keep down the number of applications of $\mathbf{B}$ to a level near the minimum.

To compute a converged Ritz vector $\mathbf{y}$, the algorithm first finds the eigenvector $\mathbf{s}$ of $\mathbf{T}_j$:

$$\mathbf{T}_j\mathbf{s} = \mathbf{s}\theta, \quad \mathbf{y} = \mathbf{Q}_j\mathbf{s}, \quad \|\mathbf{s}\| = 1.$$

Here $\mathbf{Q}_j = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_j)$.

The famous three-term recurrence can be expressed compactly in matrix form,

$$\mathbf{B}\mathbf{Q}_j = \mathbf{Q}_j\mathbf{T}_j + \mathbf{q}_{j+1}\beta_{j+1}\mathbf{e}_j^{\mathrm{T}} + \mathbf{F}_j,$$

where $\mathbf{F}_j = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_j)$ accounts for local roundoff error. On postmultiplying by $\mathbf{s}$, one finds

$$\mathbf{B}\mathbf{y} = \mathbf{y}\theta + \mathbf{q}_{j+1}\beta_{j+1}\mathbf{s}(j) + \mathbf{F}_j\mathbf{s} = \left[\mathbf{y} + \mathbf{q}_{j+1}\left(\beta_{j+1}\mathbf{s}(j)/\theta\right)\right]\theta + \mathbf{F}_j\mathbf{s}.$$

Note that we have an approximation to $\mathbf{B}\mathbf{y}/\theta$ without the expense of applying $\mathbf{B}$.

It turns out that this same modification is proposed by the authors of [1]. However their motivation was quite different. They wanted to improve the Ritz vector approximations to the eigenvectors of (1). Ours is to obtain Ritz vectors in $r(\mathbf{B})$.

In practice, the effect is quite striking. Both $\mathbf{y}$ and $\mathbf{q}_{j+1}$ may have large components along $n(\mathbf{M})$, which are almost parallel. Then a linear combination of $\mathbf{y}$ and $\mathbf{q}_{j+1}$ almost removes the contamination. Notice that there is no analogous simple expression for $\mathbf{B}\mathbf{q}_j$.

There is one further improvement to this modification. One replaces $\mathbf{s}$ by $\frac{1}{\theta}\mathbf{T}_j\mathbf{s}$! Thus one forms the $(j+1)$-vector

$$\mathbf{w} = \frac{1}{\theta}\begin{pmatrix} \mathbf{T}_j\mathbf{s} \\ \mathbf{s}(j)\beta_{j+1} \end{pmatrix}.$$

Then, $\tilde{\mathbf{y}} = \mathbf{Q}_{j+1}\mathbf{w}$ is the approximation to the wanted eigenvector.

In the table below we show the actual growth in the τ's and the values predicted by the recurrence, in a typical Lanczos run. We also show the effect of the modification, giving the size of the unwanted components in $\mathbf{y}$ and $\tilde{\mathbf{y}}$.

TABLE 1

Unwanted components in the dominant Ritz vector $\mathbf{y}_i$ and in $\tilde{\mathbf{y}}_i$, $i = 1, \ldots, 40$. Corresponding growth in the τ estimate and the actual unwanted components in the Lanczos vectors $\mathbf{q}_i$.

Index		Unwanted Components in		
i	τ_i	$\mathbf{q}_i$	$\mathbf{y}_i$	$\tilde{\mathbf{y}}_i$
1	3.600e−16	3.334e−16	2.801e−17	2.785e−17
2	1.704e−15	1.572e−15	3.002e−17	5.040e−17
3	6.180e−15	5.672e−15	6.515e−17	4.318e−17
4	1.493e−14	1.368e−14	4.860e−17	6.820e−17
5	1.909e−14	1.749e−14	3.652e−17	3.607e−17
6	2.575e−14	2.358e−14	6.627e−17	4.538e−17
7	2.920e−14	2.671e−14	2.339e−16	2.561e−16
8	4.426e−14	4.044e−14	5.984e−16	5.551e−16
9	5.993e−14	5.476e−14	2.059e−16	2.208e−16
10	1.066e−13	9.743e−14	3.127e−16	2.800e−16
11	1.570e−13	1.434e−13	7.613e−16	7.300e−16
12	3.199e−13	2.923e−13	1.939e−15	1.945e−15
13	5.790e−13	5.291e−13	1.507e−13	1.507e−13
14	1.233e−12	1.127e−12	8.510e−13	8.510e−13
15	2.978e−12	2.721e−12	5.571e−09	5.571e−09
16	6.082e−12	5.557e−12	4.190e−09	4.188e−09
17	1.487e−11	1.359e−11	1.077e−06	1.074e−06
18	2.846e−11	2.600e−11	6.278e−08	6.257e−08
19	5.618e−11	5.133e−11	3.911e−08	3.827e−08
20	1.417e−10	1.295e−10	7.926e−08	6.911e−08
21	3.838e−10	3.507e−10	4.000e−07	1.652e−07
22	1.081e−09	9.878e−10	2.534e−06	1.483e−07
23	3.535e−09	3.230e−09	1.339e−04	7.981e−08
24	9.715e−09	8.876e−09	1.516e−02	9.076e−08
25	2.031e−08	1.856e−08	1.499e−02	1.740e−07
26	6.921e−08	6.323e−08	2.505e−01	1.836e−07
27	1.888e−07	1.725e−07	8.538e−01	3.055e−07
28	7.584e−07	6.929e−07	6.893e+000	3.811e−07
29	2.656e−06	2.427e−06	1.499e+001	2.272e−07
30	9.846e−06	8.996e−06	4.546e+000	1.012e−07
31	6.434e−05	5.879e−05	1.026e+001	4.450e−07
32	2.773e−04	2.534e−04	3.579e+001	1.426e−07
33	1.183e−03	1.081e−03	3.707e+000	2.537e−07
34	7.386e−03	6.748e−03	6.485e+001	1.162e−07
35	3.552e−02	3.245e−02	5.359e+001	1.257e−07
36	2.992e−01	2.734e−01	5.298e+000	2.198e−07
37	2.027e+000	1.852e+000	3.097e+001	1.359e−07
38	1.172e+001	1.071e+001	3.367e+000	1.379e−07
39	1.090e+002	9.964e+001	1.660e+001	7.653e−08
40	1.306e+003	1.193e+003	2.904e−01	4.561e−08

We recommend using this modification in all cases, whether $\mathbf{M}$ is the identity, ill-conditioned, or singular. It should be noted that the vector $\mathbf{y} + \mathbf{q}_{j+1}\beta_{j+1}s(j)/\theta$ is not optimal in the sense of minimizing some residual. However, given the Ritz vector $\mathbf{y}$, and $\mathbf{q}_{j+1}$, then, in exact arithmetic, if $\mathbf{q}_1 \notin r(\mathbf{B})$, $\tilde{\mathbf{y}}$ is the unique linear combination in $r(\mathbf{B})$.

When $\mathbf{q}_1 \in r(\mathbf{B})$, and assuming the Lanczos algorithm has been run in exact arithmetic, other choices are possible, since all linear combinations of the Lanczos vectors lie in the right space. In Section 6 we examine how to construct the best of these other approximations. When roundoff is present, this "best" approximation will not in general lie in $r(\mathbf{B})$. That is why we recommend the use of $\tilde{\mathbf{y}}$.

3. The Algorithms. The advantages of working with the matrix of (4) are twofold. First, the Choleski factors of $\mathbf{M}$ are not needed. When $\mathbf{M}$ is diagonal the computational advantages are small, but for a more general case such as a consistent mass matrix in the dynamic analysis of structures, where $\mathbf{M}$ has the same zero structure as $\mathbf{H}$, substantial saving in both cost and storage can be achieved. Second, the computed eigenvectors are those of (1) and there is no need to recover the eigenvectors of (1) from those of (2). When the mass matrix is either singular or nondiagonal, which is the majority of cases, then this post transformation of the eigenvectors can increase the overall cost of the analysis by as much as 25%.

In a typical step, j, the generalized Lanczos process computes in order, α_j, β_{j+1}, and $\mathbf{q}_{j+1}$, to satisfy

$$(\mathbf{q}_{j+1}, \mathbf{q}_j)_{\mathbf{M}} = 0, \quad (\mathbf{q}_{j+1}, \mathbf{q}_{j-1})_{\mathbf{M}} = 0, \quad \|\mathbf{q}_{j+1}\|_{\mathbf{M}} = 1,$$

and

$$\mathbf{q}_{j+1}\beta_{j+1} = (\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{q}_j - \mathbf{q}_j\alpha_j - \mathbf{q}_{j-1}\beta_j,$$

where $(\mathbf{u}, \mathbf{v})_{\mathbf{M}} = \mathbf{u}^{\mathrm{T}}\mathbf{M}\mathbf{v}$. In exact arithmetic, $\mathbf{M}$-orthogonality is preserved against all the previous Lanczos vectors; that is, $(\mathbf{q}_i, \mathbf{q}_j)_{\mathbf{M}} = 0$ for $i \leqslant j - 1$. However, in practice some reorthogonalizations must be performed to maintain semiorthogonality (see [5]). In matrix form, the above relations read

$$(5) \qquad (\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{Q}_j - \mathbf{Q}_j\mathbf{T}_j = \mathbf{q}_{j+1}\beta_{j+1}\mathbf{e}_j^{\mathrm{T}}$$

and

$$\mathbf{Q}_j^{\mathrm{T}}\mathbf{M}\mathbf{Q}_j = \mathbf{I}_j,$$

where $\mathbf{Q}_j = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j]$, and $\mathbf{T}_j$ is a tridiagonal matrix with elements α_i and off-diagonal elements β_i;

$$\mathbf{T}_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \beta_m \\ & & & & & \beta_m & \alpha_m \end{bmatrix}.$$

Algorithm for a Lanczos Run. Pick a shift σ and factor $(\mathbf{H} - \sigma\mathbf{M}) = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$. Choose a starting vector $\mathbf{r}_0$ (in $r(\mathbf{B})$), set $\mathbf{q}_0 = 0$ and form $\mathbf{p}_1 = \mathbf{M}\mathbf{r}_0$ and $\beta_1 = (\mathbf{p}_1^{\mathrm{T}}\mathbf{r}_0)^{1/2}$.

```
For $j = 1, 2, \ldots, \text{lanmax}$ do
    (a) if so indicated then $\mathbf{M}$-orthogonalize $\mathbf{r}_{j-1}$ and $\mathbf{q}_{j-1}$ against a computed Ritz
        vector or previous Lanczos vectors, reset $\mathbf{p}_j = \mathbf{M}\mathbf{r}_{j-1}$ and $\beta_j = (\mathbf{p}_j^{\mathrm{T}}\mathbf{r}_{j-1})^{1/2}$,
        and compute any converged Ritz vectors.
    (b) $\mathbf{q}_j \leftarrow \mathbf{r}_{j-1}/\beta_j$
    (c) $\mathbf{p}_j \leftarrow \mathbf{p}_j/\beta_j$
    (d) solve $(\mathbf{H} - \sigma\mathbf{M})\mathbf{r}_j = \mathbf{p}_j$
    (e) $\mathbf{r}_j \leftarrow \mathbf{r}_j - \mathbf{q}_{j-1}\beta_j$
    (f) put $\mathbf{q}_{j-1}$ out to secondary store
    (g) $\alpha_j \leftarrow \mathbf{r}_j^{\mathrm{T}}\mathbf{p}_j$
    (h) $\mathbf{r}_j \leftarrow \mathbf{r}_j - \mathbf{q}_j\alpha_j$
    (i) form $\mathbf{p}_{j+1} = \mathbf{M}\mathbf{r}_j$
    (j) $\beta_{j+1} \leftarrow \mathbf{r}_j^{\mathrm{T}}\mathbf{p}_{j+1}$
    (k) update certain eigenvalues of $\mathbf{T}_j$ and their corresponding error bounds. Exit
        if satisfied. See [6] for more details on this.
```

Although it appears from the above implementation that we require only four vectors, $\mathbf{r}_j$, $\mathbf{q}_j$, $\mathbf{q}_{j-1}$ and $\mathbf{p}_j$, reorthogonalization forces us to use two additional vectors; one to hold each old Lanczos vector that is brought back, and the other for $\mathbf{p}_{j-1}$ if we reorthogonalize $\mathbf{q}_j$ at the same time as $\mathbf{r}_j$. See [4] for more details.

Below we compare the above unsymmetric transformation Lanczos algorithm (UTLA) to the one obtained when applying the Lanczos algorithm to the transformed problem of [1] (STLA). The current algorithm, STLA, by Ericsson and Ruhe, does not use the transformation described in the 1980 paper, but is essentially equivalent to UTLA.

In our comparison, we concentrate on the use of these algorithms for the solution of two different types of eigenproblem that commonly occur in practice. Operation counts and storage requirements for each algorithm are included in tables below.

(i) *Diagonal but singular* $\mathbf{M}$. In this case $\mathbf{M}^{1/2}$ can be computed at a cost of only n square roots, which is negligible compared to the total cost (in Table 2 we assume a square root is simply one operation). Then the cost of running Lanczos is $(2b + 7)n$ operations per step for UTLA, and $(2b + 7r)n$ for STLA. b is the average half-bandwidth of the factored $\mathbf{H}$, and $r = \text{rank}(\mathbf{M})/n$. Typically, $\frac{1}{2} \leqslant r \leqslant 1$. However, STLA must recover the eigenvectors of Eq. (1) and is therefore more expensive than UTLA by this amount. An alternative implementation of STLA that trades space for time keeps the vectors, $(\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{q}$, computed as part of the matrix-vector multiplication with the matrix of (2), in secondary store. A linear combination of these vectors can then be formed to obtain the eigenvectors of (1) directly, thus avoiding any further operations with the factored matrix. However, the need for secondary storage is doubled as compared with UTLA.

(ii) *Sparse positive definite* $\mathbf{M}$. In many applications, $\mathbf{M}$ has the same zero structure as $\mathbf{H}$. In this case two factorizations are performed by STLA. This doubles the cost of the initial step. We should mention that if a series of shifts is performed, then $\mathbf{M}$ need not be factored again. The cost of each step of STLA is more than that for UTLA by an amount which is precisely the fill-in resulting from the Choleski factorization of $\mathbf{M}$. The storage space for STLA is also more, because the factors of

TABLE 2

Operation counts for the case of a singular diagonal **M**.

	STLA	UTLA
Initial cost	$(\frac{1}{2}b^2 + 2)n$	$(\frac{1}{2}b^2 + 2)n$
One step of Simple Lanczos	$(7r + 2b)n$	$(7 + 2b)n$
j Reorthogonalizations	$2jrn$	$(2j + 1)n$
Computing an eigenvector that converged at step j	$(j + 2b + 1)n$	jn

M must be kept. There is also a further cost in STLA when transforming the eigenvectors back to those of Eq. (1).

The FORTRAN implementation of the two versions of the Lanczos algorithm mentioned above are about the same length. UTLA requires no post transformation of the computed eigenvectors that STLA must perform. On the other hand, the inner loop of a Lanczos step and the reorthogonalization step are slightly longer in UTLA because of the **M**-inner product.

TABLE 3

Operation counts for the case of a positive definite, sparse **M**. *Here mn is the cost of applying* **M** *to an n-vector.*

	STLA	UTLA
Initial cost	$(b^2 + 1)n$	$(\frac{1}{2}b^2 + m + 1)n$
One step of Simple Lanczos	$(5 + 4b)n$	$(7 + m + 2b)n$
j Reorthogonalizations	$2jn$	$(2j + m)n$
Computing an eigenvector that converged at step j	$(j + 2b + 1)n$	jn

TABLE 4

Storage demands of the two methods for the cases under consideration. Here mn is the cost of applying **M** *to an n-vector.*

Case	STLA	UTLA
Diagonal **M**	$(b + 6r)n$	$(b + 7)n$
Consistent **M**	$(5 + 2b)n$	$(6 + m + b)n$

4. Accuracy. We turn now to the accuracy of the eigenvalues computed by means of a spectral transformation.

In [1] it is pointed out that for those λ_i very close to σ the situation is most satisfactory. Their results show

$$\left|\lambda_i - \left(\sigma + \frac{1}{\theta_i}\right)\right| \leqslant \frac{\beta_{j+1}}{\theta_i^2}|s_i(j)|,$$

where θ_i is an eigenvalue of $\mathbf{T}_j$ and $\mathbf{s}_i(j)$ is the bottom element of the normalized eigenvector of $\mathbf{T}_j$ corresponding to θ_i. Suppose that $|\lambda_1 - \sigma| < |\sigma|/100$. Then, after a few steps, $\|\mathbf{T}_j\| = |\theta_1^{(j)}| > 100/|\sigma|$, and

$$\left|\lambda_1 - \left(\sigma + \frac{1}{\theta_1^{(j)}}\right)\right| \leqslant \frac{\beta_{j+1}}{\theta_1^{(j)}}|\mathbf{s}_1(j)|\frac{|\sigma|}{100}.$$

Normally, $\beta_{j+1} \leqslant \|\mathbf{T}_j\|/10$. Indeed, $\beta_j \approx \|\mathbf{T}_j\|/100$ is typical. In these circumstances, the relative error in $(\sigma + 1/\theta_1^{(j)})$ is four orders of magnitude less than $|\mathbf{s}_1(j)|$. This is a bonus arising from the fact that $1/\theta_1^{(j)}$ is a small correction to σ.

Unfortunately, the term θ_i^{-2} on the right of the error bound works against us when determining eigenvalues much smaller than σ. In these circumstances, say

$$|\lambda_9| < |\sigma|/100,$$

there must be two decimal digits of cancellation in the final formation of $\sigma + 1/\theta_9^{(j)}$. Whatever the relative accuracy of θ_9, two digits will be lost in this way. Moreover, $|\theta_9|$ will be small relative to $\|\mathbf{T}_j\|$, and so it appears to be more difficult to attain high relative accuracy in θ_9. However, appearances are deceptive here. When σ is very close to an eigenvalue then $\mathbf{T}_j$ will be a graded matrix (the first few rows of $\mathbf{T}_j$ will be much larger in norm than the rest). With graded matrices it is possible to compute eigenvalues to high relative accuracy. It is necessary that the criterion for acceptance of small eigenvalues be proportional to the magnitude of that eigenvalue and not $\|\mathbf{T}_j\|$. Unfortunately, some codes always use $\|\mathbf{T}_j\|$.

It is important to consider these aspects of the algorithm, because when factorization of $\mathbf{H} - \sigma\mathbf{M}$ is expensive relative to a Lanczos step then it is efficient to prolong Lanczos runs. Long runs will produce eigenvalues quite far from σ. Our remarks show that the only severe degradation in accuracy would arise in computing eigenvalues less than 10^{-2} from a shift σ exceeding 10^2. Shifts should be selected to avoid such bizarre configurations.

5. The Case for Spectral Transformation. The case is not self-evident. If $\mathbf{M}$ is diagonal and positive definite then the operator $\mathbf{M}^{-1/2}\mathbf{H}\mathbf{M}^{-1/2}$ (or $\mathbf{H}\mathbf{M}^{-1}$) is readily available without the need to factor $\mathbf{H} - \sigma\mathbf{M}$ (or solve a system of equations). In a number of applications, $\mathbf{H} - \sigma\mathbf{M}$ cannot be factored entirely within primary store of the computer, and expensive transfer operations may dominate the process. If the factorization takes as long as n Lanczos steps then we might ask whether the spectral transformation approach is really warranted.

The answer is no. One of the original attractions of the Lanczos algorithm was that it gave a way to find the small eigenvalues of a matrix without any factorization at all. The price paid for this feature is that more Lanczos steps will be required than with the inverted operator.

The trade-off is affected by n, and this is the point we wish to emphasize. The rate of convergence (more precisely, the rate of emergence) of an eigenvalue does not depend solely on its separation δ from its neighbors, but on the ratio of δ to the total spread of all the eigenvalues, δ/spread. Moreover, δ/spread only governs the rate in the early stages of the algorithm. When the number of steps j exceeds $n/3$ then these estimates become too weak to be useful.

For the majority of applications using $\mathbf{M}^{-1/2}\mathbf{H}\mathbf{M}^{-1/2}$, the ratio δ/spread decreases faster than $1/n$ (more like $1/n^2$). For large enough n it will be necessary to take essentially n (actually $> n$) Lanczos steps to find the smallest few eigenvalues. If reorthogonalization is used, the cost of each of the later steps is $O(n^2)$. If no reorthogonalization is used, $2n$ or $3n$ Lanczos steps will be needed.

On the other hand, the inverted operator with a good choice of σ permits 20 or 30 eigenvalues to be computed in 40 or 60 steps almost independent of n. Consequently, factorization of $\mathbf{H} - \sigma\mathbf{M}$ should be avoided only if it costs more than $n/2$ matrix-vector products of the form $\mathbf{H}\mathbf{M}^{-1}\mathbf{q}$. See [7] for more details.

6. Projection of H. From the relations governing the Lanczos process described in Section 4, one can deduce that $\mathbf{T}_j$ is the projection of $\mathbf{M}(\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}$ on the space spanned by the columns of $\mathbf{Q}_j$; that is, $\mathbf{T}_j = \mathbf{Q}_j^T\mathbf{M}(\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{Q}_j$. However, it seems more natural to seek approximations to the eigenvalues of the original problem (with shift σ) by using the projection of $\mathbf{H} - \sigma\mathbf{M}$ onto this space. That is, one would consider $\mathbf{W}_j = \mathbf{Q}_j^T(\mathbf{H} - \sigma\mathbf{M})\mathbf{Q}_j$. Indeed, the Rayleigh-Ritz approximations are different. To establish the relation between $\mathbf{W}_j$ and $\mathbf{T}_j$, we need some extra notation. Let

$$\mathbf{W}_{j+1} = \begin{bmatrix} \mathbf{W}_j & \mathbf{w}_j \\ \mathbf{w}_j^T & \omega_{j+1} \end{bmatrix}.$$

We premultiply the Lanczos equation (5) by $\mathbf{Q}_j^T(\mathbf{H} - \sigma\mathbf{M})$. Note the $\mathbf{M}$-orthogonality of the Lanczos vectors to obtain

$$\mathbf{W}_j T_j = \mathbf{I}_j - \beta_{j+1}\mathbf{w}_j\mathbf{e}_j^T. \tag{6}$$

Similarly, premultiply Eq. (5) by $\mathbf{q}_{j+1}^T(\mathbf{H} - \sigma\mathbf{M})$ and again use the $\mathbf{M}$-orthogonality property to find

$$\mathbf{T}_j\mathbf{w}_j = -\beta_{j+1}\omega_{j+1}\mathbf{e}_j.$$

The eigenvalues of $\mathbf{W}_j$ are the Ritz value approximations to $(\lambda_i - \sigma)$, $i = 1, \ldots, j$ from span $\mathbf{Q}_j$. But it is just as convenient to determine the eigenvalues of $\mathbf{W}_j^{-1}$. From (6),

$$\begin{aligned} \mathbf{W}_j^{-1} &= \mathbf{T}_j\left(\mathbf{I}_j + \rho_j\mathbf{w}_j\mathbf{e}_j^T\right), \qquad \rho_j = \beta_{j+1}/\left(1 - \beta_{j+1}\mathbf{e}_j^T\mathbf{w}_j\right) \\ &= \mathbf{T}_j + \rho_j\mathbf{T}_j\mathbf{w}_j\mathbf{e}_j^T \\ &= \mathbf{T}_j + \mu_j\mathbf{e}_j\mathbf{e}_j^T. \end{aligned} \tag{7}$$

Here,

$$\mu_j = -\frac{\beta_{j+1}^2\omega_{j+1}}{1 - \beta_{j+1}\mathbf{e}_j^T\mathbf{w}_j}. \tag{8}$$

Equation (7) shows that $\mathbf{W}_j$ is the inverse of a tridiagonal matrix that differs from $\mathbf{T}_j$ only in the last diagonal entry.

To evaluate μ_j, equate the $(j + 1, j + 1)$ elements on each side of

$$\mathbf{W}_{j+1}\left(\mathbf{T}_{j+1} + \mu_{j+1}\mathbf{e}_{j+1}\mathbf{e}_{j+1}^T\right) = \mathbf{I}_{j+1}$$

to find

$$\beta_{j+1}\mathbf{w}_j^T\mathbf{e}_j + \omega_{j+1}\left(\alpha_{j+1} + \mu_{j+1}\right) = 1. \tag{9}$$

Now eliminate ω from (8) and (9) to find

$$\mu_{j+1} = -\alpha_{j+1} - \frac{\beta_{j+1}^2}{\mu_j}.$$

Next we examine how well the eigenpairs, $(\theta, \mathbf{s})$, of $\mathbf{T}_j$ approximate those of $\mathbf{W}_j^{-1}$. The norm of the residual vectors is easily computed,

$$\left\|\left(\mathbf{W}_j^{-1} - \theta\right)\mathbf{s}\right\| = \left\|\mathbf{T}_j\mathbf{s} - \theta\mathbf{s} + \mu_j\mathbf{e}_j\mathbf{e}_j^{\mathrm{T}}\mathbf{s}\right\| = \left|\mu_j s(j)\right|.$$

Clearly, those Ritz vectors that have stabilized in the Lanczos process are least affected by the change of projection. Except for the occurrence of large μ, it is not clear that, in practice, it is worth computing these modifications.

The starting value μ_1 is obtained directly from $\omega_1 = \mathbf{q}_1^{\mathrm{T}}(\mathbf{H} - \sigma\mathbf{M})\mathbf{q}_1$. That is, $\mu_1 = 1/\omega_1 - \alpha_1$. When the starting vector for the Lanczos algorithm is $\mathbf{r} = (\mathbf{H} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{u}$ then $\omega_1 = \mathbf{r}^{\mathrm{T}}\mathbf{M}\mathbf{u}/\beta_1^2$.

Acknowledgment. The authors express their thanks to the referee for comments that led to significant improvements in the section on singular **M**.

Lockheed Palo Alto Research Laboratory
3251 Hanover Street
Palo Alto, California 94304

Department of Mathematics
University of California
Berkeley, California 94720

Department of Mathematics
Chalmers University of Technology
and the University of Göteborg
S-412 96 Göteborg, Sweden

Lockheed Space System Division
Sunnyvale, California 94088

1. T. Ericsson & A. Ruhe, "The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems," *Math. Comp.*, v. 35, 1980, pp. 1251–1268.

2. C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Standards*, v. 45, 1950, pp. 255–281.

3. B. Nour-Omid, B. N. Parlett & R. L. Taylor, "Lanczos versus subspace iteration for solution of eigenvalue problems," *Internat. J. Numer. Methods Engrg.*, v. 19, 1983, pp. 859–871.

4. B. Nour-Omid, *The Lanczos Algorithm for the Solution of Large Generalized Eigenproblems*, Tech. Rep. UCB/SESM-84/04, Dept. of Civil Engineering, University of California, Berkeley, 1984.

5. B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N. J., 1980.

6. B. N. Parlett & B. Nour-Omid, "The use of refined error bounds when updating eigenvalues of tridiagonals," *Linear Algebra Appl.*, v. 68, 1985, pp. 179–219.

7. D. S. Scott, "The advantages of inverted operators in Rayleigh-Ritz approximations," *SIAM J. Sci. Statist. Comput.*, v. 3, 1982, pp. 68–75.