SIAM J. SCI. COMPUT.
Vol. 14, No. 3, pp. 585-593, May 1993

© 1993 Society for Industrial and Applied Mathematics
006

PRECONDITIONING THE LANCZOS ALGORITHM FOR SPARSE SYMMETRIC EIGENVALUE PROBLEMS*

RONALD B. MORGAN† AND DAVID S. SCOTT‡

Abstract. A method for computing a few eigenpairs of sparse symmetric matrices is presented and analyzed that combines the power of preconditioning techniques with the efficiency of the Lanczos algorithm. The method is related to Davidson's method and its generalizations, but can be less expensive for matrices that are fairly sparse. A double iteration is used. An effective termination criterion is given for the inner iteration. Quadratic convergence with respect to the outer loop is shown.

Key words. eigenvalues, sparse matrices, Lanczos, preconditioning, Davidson's method

AMS(MOS) subject classifications. 65F15, 15A18

1. Introduction. We consider the problem of computing a few eigenvalues and eigenvectors of a large, sparse, symmetric matrix. It is assumed that factoring the matrix is impractical due to its size and sparsity structure. A method is presented that incorporates both the technique of preconditioning and the Lanczos algorithm. A double iteration scheme is used. The outside loop updates a certain preconditioned matrix; the inside loop applies the Lanczos algorithm. An effective termination criterion is given for the inner loop. This method can be very efficient if the matrix is fairly sparse and an approximate inverse is easily available.

This section briefly discusses background material on eigenvalue techniques and on preconditioning. Section 2 presents the method. Section 3 gives convergence results including asymptotic quadratic convergence with respect to the outer loop. Section 4 gives examples and looks at some implementation details. Comparisons are made with other methods.

Krylov subspace methods are popular for both eigenvalue problems and linear equations problems. Krylov subspaces are used by the Lanczos algorithm [11], [19] for eigenvalues and by the conjugate gradient method [2], [9], [10] for linear equations (see [7, p. 523] and [18] for the relation between the methods). Both methods have convergence problems if the distribution of eigenvalues is unfavorable. The Lanczos algorithm has difficulty if the desired eigenvalues are not well separated from the rest of the spectrum. The conjugate gradient method needs the spectrum to be somewhat separated from zero. Convergence of the conjugate gradient method is often improved by preconditioning (multiplying the matrix equation by an approximate inverse) [1], [3], [8], [13].

It would also be desirable to improve the convergence of the Lanczos algorithm with preconditioning, but this is not straightforward. Preconditioning can be applied indirectly to eigenvalue problems by using the preconditioned conjugate gradient method to solve equations for inverse iteration, the Rayleigh quotient iteration [12], [22], or shift-and-invert Lanczos [6], [20].

Davidson's method [4] and the generated Davidson (GD) method [15], [17] give a more direct approach to preconditioning eigenvalue problems. Suppose eigenvalues are sought for the matrix A. Then these methods generate a subspace with the operator $(M-\rho I)^{-1}(A-\rho I)$, where ρ is the latest approximate eigenvalue obtained by the

* Received by the editors March 23, 1988; accepted for publication (in revised form) April 6, 1992.

† Department of Mathematics, Baylor University, Waco, Texas 76798.

‡ Intel Scientific Computers, 15201 N.W. Greenbrier Parkway, Beaverton, Oregon 97006.

Rayleigh-Ritz procedure [19]. In Davidson's method, the operator M is D, the main diagonal of A. For the GD method, M can be any approximation to A. $M-\rho I$ is in effect a preconditioner for $A-\rho I$. Let (λ, z) be the desired eigenpair. Asymptotically, the subspace produced by these methods resembles a Krylov subspace generated by $(M-\lambda I)^{-1}(A-\lambda I)$. The operator $(M-\lambda I)^{-1}(A-\lambda I)$ has one eigenvalue at 0 with z as the associated eigenvector. The rest of the spectrum tends to be compressed around 1 by the preconditioning. This makes 0 well separated, and convergence is rapid toward z (see [17] for more detail).

The GD method is more expensive per iteration than the Lanczos algorithm. There is the cost of the preconditioning and also the cost of the Rayleigh-Ritz procedure. Full orthogonalization must be done. Davidson's method is often applied to problems for which the main expense is the matrix-vector products, and thus any reduction in the number of iterations is worthwhile. For other problems, particularly those where A is quite sparse, the orthogonalization costs are significant. Restarting reduces the expense but slows down the convergence.

A method is desired that could be preconditioned and could take advantage of the Lanczos recurrence. One possibility involves using the Lanczos algorithm to build the subspace for GD [16, pp. 71-83]. The Rayleigh-Ritz expense can be reduced by a factor of five. However, a double iteration is used and the cost still grows as the size of the subspace increases. In the next section, another method is presented that uses preconditioning and the Lanczos algorithm. The difference is that in this method the Rayleigh-Ritz procedure is not done with respect to A. A double iteration is again used, but the costs of order n are fixed as the subspace grows.

2. The Preconditioned Lanczos (PL) Algorithm.

Preconditioning the Lanczos algorithm was suggested by Scott [21] as a special case of a method for generalized eigenvalue problems, but it was not investigated, and the algorithm has not been given before for the standard eigenvalue problem. Our purpose here is threefold. We establish important convergence results. Scott's results [21] do not take into account preconditioning or early termination of the inner iteration. Second, we implement the method and show that it is useful. Third, we derive the method showing its connection with the Davidson and GD methods. This is important because it gives insight into why the method is effective and because Davidson's method is well known among quantum chemists.

The GD method uses the operator $N(\rho)\equiv(M-\rho I)^{-1}(A-\rho I)$ to generate a subspace, but it uses the operator A in the Rayleigh-Ritz procedure for extracting approximate eigenvectors from the subspace. However, since $N(\rho)$ has an eigenvector approximating one of A, the Rayleigh-Ritz procedure with a fixed $N(\rho)$ is also useful for computing an eigenvector of A. It is necessary to restart the Rayleigh-Ritz occasionally with a new ρ, because the eigenvector of $N(\rho)$ is only an approximation. We use this idea, but transform $N(\rho)$ to a symmetric operator so that the Lanczos algorithm can be applied. A positive definite preconditioner is required, so we replace $M-\rho I$ with M_k. The algorithm is called the PL Algorithm.

THE PL ALGORITHM. Given a vector x_0, compute $\rho_0 = x_0^T A x_0 / x_0^T x_0$, and FOR $k=0,1,2,\dots,$ DO 1 to 5

1. Choose M_k to be a positive definite matrix approximating $A-\rho_k I$, and let $L_k L_k^T$ be its Cholesky factorization.
2. Define $W_k \equiv L_k^{-1}(A-\rho_k I)L_k^{-T}$.

3. Run the Lanczos iteration with W_k and starting vector $L_k^T x_k$, until the smallest Ritz value is bounded away from zero by the residual norm [19, p. 260]. Letting θ_k be the smallest Ritz value and y_k be the corresponding Ritz vector of unit length, this stopping test is $-\theta_k > \| W_k y_k - \theta_k y_k \|$.
4. Let $x_{k+1} = L_k^{-T} y_k$.
5. Set $\rho_{k+1} = x_{k+1}^T A x_{k+1} / x_{k+1}^T x_{k+1} = \rho_k + \theta_k / x_{k+1}^T x_{k+1}$.

The operator $W_k \equiv L_k^{-1}(A - \rho_k I) L_k^{-T}$ is related to $M_k^{-1}(A - \rho_k I)$ by a similarity transformation. And $M_k^{-1}(A - \rho_k I)$ has an eigenvector approximating an eigenvector of A. By applying the Lanczos iteration to W_k, computing a Ritz vector, and then transforming it back, we get x_{k+1}, an approximate eigenvector of A. The Rayleigh quotient ρ_{k+1} of x_{k+1} is an approximate eigenvalue. The residual norm [19, p. 69] of (ρ_{k+1}, x_{k+1}) with respect to A can be monitored to determine a stopping point for the PL iteration.

The main cost of the method is in the Lanczos loop. A matrix-vector product with A is required and systems of linear equations in L_k and L_k^T are solved. So it is important for the Lanczos iteration to converge quickly. The spectrum of W_k, or equivalently of $M_k^{-1}(A - \rho_k I)$, is the determining factor. Here the preconditioning improves the eigenvalue distribution just as it does for the GD method. The early termination test in step 3 of the PL Algorithm also reduces the number of Lanczos iterations. In the following section it is shown that the early termination test does not significantly impair the convergence of the outer iteration.

3. Convergence of PL.

For convergence, the preconditioner does not need to be an accurate approximation. If there is boundedness, then the sequence $\{\rho_k\}$ in the PL Algorithm converges to an eigenvalue of A at an asymptotically quadratic rate. First, convergence of ρ_k to an eigenvalue is shown.

THEOREM 1. *Assume that both M_k and M_k^{-1} are uniformly bounded in norm. Then ρ_k converges to an eigenvalue of A.*

Proof. First we will establish the equality asserted in step 5 of the PL Algorithm.

$$\begin{aligned} \rho_{k+1} &= x_{k+1}^T A x_{k+1} / x_{k+1}^T x_{k+1} \\ &= \rho_k + (L_k^{-T} y_k)^T (A - \rho_k I) L_k^{-T} y_k / x_{k+1}^T x_{k+1} \\ &= \rho_k + y_k^T W_k y_k / x_{k+1}^T x_{k+1} \\ &= \rho_k + \theta_k / x_{k+1}^T x_{k+1}. \end{aligned} \tag{1}$$

Next we will show that θ_k is nonpositive, so that from (1), $\{\rho_k\}$ is a nonincreasing sequence. The (1, 1) element of the tridiagonal matrix T that is generated by the Lanczos iteration is the Rayleigh quotient of the starting vector $L_k^T x_k$ with respect to W_k. It can be seen that this Rayleigh quotient is zero, since $\rho_k = x_k^T A x_k / x_k^T x_k$. Using Cauchy's interlace theorem, θ_k is less than or equal to zero, since θ_k is the smallest eigenvalue of T.

The sequence of ρ_k's is bounded below by the smallest eigenvalue of A. Therefore, ρ_k converges. Let the limit be τ, and let $\varepsilon_k = \rho_k - \tau$.

Because by assumption M_k^{-1} is bounded in norm, so also is L_k^{-1}. The vector y_k is of unit length, so $x_{k+1}^T x_{k+1} = (L_k^{-T} y_k)^T L_k^{-T} y_k$ is bounded. From (1),

$$|\theta_k| = (\rho_k - \rho_{k+1}) x_{k+1}^T x_{k+1} \leqq \varepsilon_k x_{k+1}^T x_{k+1}. \tag{2}$$

Therefore, θ_k converges to zero. But θ_k is within the residual norm bound of some eigenvalue of W_k, say ω_k. Because of the stopping test for the Lanczos iteration in step 3 of the PL algorithm, ω_k is within $|\theta_k|$ of θ_k and so within $2|\theta_k|$ of zero. Therefore, ω_k converges to zero.

Since W_k has an eigenvalue converging to zero, $\|W_k^{-1}\|$ goes to infinity. Since

$$\|W_k^{-1}\| = \|L_k^T(A-\rho_k I)^{-1}L_k\| \leqq \|L_k^T\|\,\|(A-\rho_k I)^{-1}\|\,\|L_k\|,$$

and $\|L_k^T\|$ and $\|L_k\|$ are bounded, ρ_k must be converging to an eigenvalue of A, and the proof is complete. □

The theorem does not say which eigenvalue ρ_k converges to, but it is extremely likely to be the smallest one. For this not to be the case would require that each starting vector of the Lanczos iteration have a very small component in the direction of the eigenvector associated with the smallest eigenvalue of W_k.

Let ρ_k converge to λ. The next theorem shows that this convergence is asymptotically quadratic. The "big O" notation will be used: $\beta = \gamma + O(\varepsilon)$ implies $|\beta-\gamma| \leqq c\varepsilon$ for $c>0$.

THEOREM 2. *Assume that both M_k and M_k^{-1} are uniformly bounded in norm. Then ρ_k converges at an asymptotically quadratic rate.*

Proof. Using the definitions of W_k and y_k,

$$\begin{aligned}(A-\rho_k I)x_{k+1} &= L_k W_k L_k^T x_{k+1} = L_k W_k y_k \\ &= L_k[(W_k y_k - \theta_k y_k) + \theta_k y_k].\end{aligned}$$

Because of the stopping test in step 3 of the PL Algorithm and the fact that y_k is a unit vector,

$$\begin{aligned}\|(A-\rho_k I)x_{k+1}\| &\leqq \|L_k\|[\|W_k y_k - \theta_k y_k\| + |\theta_k|\,\|y_k\|] \\ &\leqq 2\|L_k\|\,|\theta_k|.\end{aligned} \tag{3}$$

Let the residual vector of x_{k+1} with respect to A be r_{k+1}:

$$\|r_{k+1}\| = \|(A-\rho_{k+1}I)x_{k+1}\| / \|x_{k+1}\|.$$

Using first the minimal residual property [19, p. 12], then (3),

$$\begin{aligned}\|r_{k+1}\| &\leqq \|(A-\rho_k I)x_{k+1}\| / \|x_{k+1}\| \\ &\leqq 2\|L_k\|\,|\theta_k| / \|x_{k+1}\|.\end{aligned}$$

With (2) and the fact that $\|L_k\|$ is bounded and $\|x_{k+1}\|$ is bounded away from zero, we have

$$\|r_{k+1}\| = O(\varepsilon_k). \tag{4}$$

Let γ_k be the gap between ρ_k and the nearest eigenvalue of A other than λ. This gap approaches a nonzero constant because ρ_k converges to λ. By the Kato-Temple inequality [19, p. 222],

$$|\rho_{k+1} - \lambda| \leqq \|r_{k+1}\|^2 / \gamma_{k+1}.$$

With (4), this becomes

$$\rho_{k+1} = \lambda + O(\varepsilon_k^2),$$

and we have quadratic convergence. □

4. Implementation and examples. Here examples are given and comparisons are made with other methods. First, some implementation details are discussed.

On the choice of preconditioner, one possibility is to pick M to be a portion of A. Then factor $M-\rho_k I$ in LDL^T form, and let M_k equal $L|D|L^T$. Small pivot elements during the factorization should be changed. Another approach is to let M_k be $M-\sigma I$, where σ is an estimate of the smallest eigenvalue of A and is below the spectrum of M. A powerful preconditioning approach is to use incomplete factorization [8], [13] of $A-\sigma I$. Here it is important that $A-\sigma I$ be positive definite. Even then, adjustment of small or negative pivot elements may be necessary for a stable factorization. The symmetric successive overrelaxation (SSOR) [9] preconditioner is another possibility.

If more than one eigenvalue is desired, a form of deflation can be used. The eigenvalues that are already computed can be shifted out of the way. For example, once (λ_1, z_1) has been computed, $A-\rho_k I$ can be replaced by $A-\rho_k I+\gamma z_1 z_1^T$, where γ is large enough to shift λ_1 away from the other desired eigenvalues. The shifting should be kept as small as possible to comfortably achieve this, because a large shift and the effect of M_k^{-1} could possibly produce a large eigenvalue for W_k. This would slow the convergence of the Lanczos loop. The starting vector for finding the second eigenvalue can be determined while the first eigenvalue is being computed. At some point, the second Ritz vector of W_k can be computed and multiplied by L_k^{-T}. In the examples that follow, the starting vectors for interior eigenpairs are calculated once the residual norm for the previous eigenpair reached two-thirds in orders of magnitude of the desired accuracy.

An additional stopping test is applied in the Lanczos loop to terminate early if it is likely that the desired accuracy has been attained. This is done by comparing the improvement in the residual norm with respect to W_k of the Ritz vector in the Lanczos loop with the improvement needed in the residual norm with respect to A of the approximate eigenvector. (The Lanczos loop is terminated when the log base 10 of the ratio of the residual norms of Ritz vectors at the beginning and current point of Lanczos is less than the log of the ratio of the residual norm of x_k to the specified residual tolerance, divided by a safety factor of 0.9 times the ratio of logs of improvements in residual norms for the previous k.) In our testing, this check for early termination helps the method avoid extra computation.

Example 1. The first test matrix is $A=\mathrm{Diag}\,(1, 2, 3, \ldots, 1000)$. For the preconditioner, let $M=\mathrm{Diag}\,(10.1, 10.2, 10.3, \ldots, 110)$ and $M_k=M-\rho_k I$. The starting vector is $(1, 1/2, 1/3, \ldots, 1/1000)^T$. The smallest eigenvalue and eigenvector are computed. The requested residual tolerance is 1×10^{-8}. Table 1 gives the results of the PL method. It lists the number of iterations in the Lanczos loop (this is the size of the Krylov subspace generated and also the number of matrix-vector products with A), the new approximate eigenvalue ρ_{k+1}, and the residual norm. It shows the expected quadratic convergence. The total number of iterations is 88. The standard Lanczos algorithm requires 194 iterations for the same task. So the preconditioning cuts the number of

TABLE 1
PL *for Example* 1.

k	Iterations	ρ_{k+1}	Residual norm
		4.55	24.2
0	3	1.67	4.5
1	8	1.064	1.0
2	13	1.00035	0.101
3	25	1.0000000058	0.56E−3
4	39	1.0000000000	0.76E−8

iterations in half. With better preconditioning, this can be improved. For example, with $M = \mathrm{Diag}(1.1, 1.2, 1.3, \ldots, 101)$, the PL Algorithm converges in only 30 total iterations.

Next we compare three methods that use preconditioning: the PL Algorithm, the GD method, and the Rayleigh quotient iteration (RQI) [12], [22] with preconditioned SYMMLQ [18] in the inner loop. The GD method has a limit of 40 on the size of the subspace because of the orthogonalization costs. The RQI uses termination criterion in SYMMLQ of $\mathrm{RTOL} = \mathrm{Min}\,\{0.01 * \|r\|, \|r\|^3\}$, where $\|r\|$ is the residual norm at the previous step of RQI. The choice of termination criterion can affect the convergence a great deal, especially when the starting vector for RQI is not accurate. Three matrices with increasing difficulty are used. Let $A = \mathrm{Diag}\,(1, 1+\delta, 1+2\delta, \ldots, 1+99\delta, 2+99\delta, 3+99\delta, \ldots, 901+99\delta)$, for $\delta = 1$, 0.1, and 0.01. M is as originally defined. Table 2 gives the total number of matrix-vector products used for each of these methods. For the toughest problem ($\delta = 0.01$), the PL Algorithm is much better than the GD method. Not only are fewer iterations required, but each iteration is considerably less expensive. The PL Algorithm has the advantage of being able to cheaply generate large subspaces, and for difficult problems, large subspaces are needed. The PL Algorithm also has an advantage when storage is limited and the vectors spanning the subspace must be stored on disk, because the GD method accesses these vectors at every iteration.

TABLE 2
Preconditioning methods.

	Number of matrix-vector products		
	Matrix	Matrix	Matrix
Method	$\delta = 1$	$\delta = 0.1$	$\delta = 0.01$
PL	88	247	555
GD	69	309	1584
RQI	99	333	582

The PL Algorithm and RQI are fairly comparable in this example. This is not surprising, since the two methods do resemble each other. The main difference is that the PL Algorithm has an eigenvalue problem in the inner loop instead of a linear equations problem. RQI does not require storage of the vectors spanning the subspace and has easier deflation. But the PL Algorithm has the advantage of being more reliable in converging to the smallest eigenvalue. To illustrate this with the first matrix ($\delta = 1.0$) and starting vector $(5, 5, 5, 5, 5, 1/6, 1/7, 1/8, \ldots, 1/1000)^T$, RQI converges to the fifth eigenvalue. The PL Algorithm still converges to the first eigenvalue. Of course, it is well known that to insure convergence to a particular eigenvalue, RQI needs a good starting vector or an implementation that combines it with inverse iteration.

The shift-and-invert Lanczos method with the preconditioned conjugate gradient method can also be used, but we have not found it to be competitive. For one implementation with early termination of the inner conjugate gradient iteration, 893 iterations were required with the matrix $\delta = 0.1$. However, there is probably a better implementation.

Example 2. This and the next example use matrices from the Harwell-Boeing sparse matrix collection [5]. The matrices selected are quite sparse and have small eigenvalues that are not well separated. The first matrix is SHERMAN1, from oil reservoir simulation. The dimension is 1000, and it has seven nonzero bands with half-bandwidth of 100. The six smallest eigenvalues are 0.000323, 0.00102, 0.00111, 0.00151, 0.00192, and 0.00205, and the largest is 5.04.

The elements of the starting vector are selected randomly on the interval (0, 1). The computations are performed on an IBM 3090-170J computer using double precision arithmetic. The Lanczos algorithm is restarted every 250 steps and the GD method every 40. The small eigenvalue problems generated by these methods are solved with EISPACK routines, though an iterative method such as subspace iteration [19] might be better. The small problem is solved at the end of each 250 iterations of Lanczos, at every step for GD (as is required), and for PL at each of the first seven iterations and then at multiples of five. A few steps of inverse iteration are needed at the start of RQI. We let $\rho = 0$ for the first three outer iterations, then switch to regular RQI. The preconditioner for all methods is from incomplete factorization [13] of A, with no fill-in allowed.

First, the smallest eigenpair of the matrix is computed with residual norm tolerance of 1×10^{-8}. Table 3 lists the total number of matrix-vector products (MVPs) with A, along with the time in CPU seconds. The preconditioning methods need far fewer iterations than the Lanczos algorithm. The PL Algorithm is the fastest method, even though the GD method requires fewer iterations.

TABLE 3
Harwell-Boeing matrices.

	SHERMAN1		SHERMAN1		NOS6	
	One eigenpair		Five eigenpairs		One eigenpair	
Method	MVPs	Time	MVPs	Time	MVPs	Time
Lanczos	750	7.6	—		—	
PL	61	1.2	487	9.3	987	16.7
GD	39	2.2	149	9.7	6920	346
RQI	221	4.2	—		—	

Next the five smallest eigenpairs are computed. PL uses $\gamma = 0.01$ to shift computed eigenvalues. The results are not very sensitive to the choice of γ. (For example, with $\gamma = 0.002$, 445 matrix-vector products are needed, while PL with $\gamma = 0.1$ uses 533.) The PL and GD methods require about the same amount of time. The simple Lanczos algorithm with restarting has difficulty calculating these eigenvalues. A deflation scheme is possible, but it might be better to use a block Lanczos method [6] or out-of-core storage to avoid restarting. RQI is implemented with $\rho = 0$ for the first three outer iterations for each eigenvalue and with a small Raleigh-Ritz procedure [19] applied to the last five approximate eigenvectors to get each new starting vector. RQI finds the first, third, fifth, second, and ninth eigenvalues and uses 2001 total matrix-vector products. A different implementation might help, but this appears to be a difficult problem for RQI.

Example 3. Here we use a structural engineering matrix of dimension 675 called NOS6 in Simon's LANPRO collection [5]. The starting vector has entries distributed randomly on the interval $(-1, 1)$, and the residual tolerance is 1×10^{-4}. The two smallest eigenvalues are 1.0000153 and 1.0000254, which are extremely close considering that the matrix has elements of size 4×10^{6}. The maximum size subspace for the Lanczos algorithm and for the Lanczos iteration of PL is 350, while GD again uses dimension 40. The results for computing one eigenvalue are given in Table 3. The PL Algorithm is more than 20 times faster than GD. The Lanczos algorithm shows little sign of ever finding the smallest eigenvalue. RQI is implemented with initial steps of inverse iteration using $\rho = 0$ until the Rayleigh quotient is less than 1.00002, but it also cannot find this eigenvalue (it never switches from inverse iteration to RQI). In another test with $\rho = 1$ initially, RQI converges in 1961 total iterations and 30.2 seconds. So even with an extremely accurate initial estimate of the eigenvalue, RQI is slower than the PL Algorithm.

5. Conclusion. Like the Davidson and GD methods, the PL Algorithm uses preconditioning to accelerate the convergence. However, the expense per iteration is reduced by applying the Lanczos iteration. A double iteration scheme is used. The inner iteration is terminated early for efficiency, but convergence is still asymptotically quadratic with respect to the outer loop.

The PL Algorithm can significantly reduce the expense of computing eigenvalues and eigenvectors of some matrices. It is most useful for sparse matrices, for difficult problems, and for computing only a very small number of extreme eigenpairs.

We conclude with some possibilities for further research. If an eigenvalue estimate σ is known, $A - \rho_k I$ can be replaced with $A - \sigma I$ in step 2 of the PL Algorithm, for the first few outer loops. Also, the method might be more robust if the x_k vectors are saved and combined using the Rayleigh-Ritz procedure with A. This would not be too expensive if only the last few x_k vectors are used. Finally, an interior implementation [14] of the Lanczos loop might allow computation of more than one eigenvalue without using deflation. This also might allow computation of several eigenvalues at a time if estimates are known of the desired eigenvalues.

Acknowledgments. The authors would like to thank the referees for many suggestions that considerably improved the paper.

REFERENCES

[1] O. B. Axelsson, *On the efficiency of a class of A-stable methods*, BIT, 14 (1974), pp. 279-287.

[2] A. Cline, *Several observations on the use of conjugate gradient methods*, ICASE Report 76-22, NASA Langley Research Center, Hampton, VA, 1976.

[3] P. Concus, G. H. Golub, and G. Meurant, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 220-252.

[4] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, J. Comput. Phys., 17 (1975), pp. 87-94.

[5] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1-14.

[6] T. Ericsson and A. Ruhe, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Math. Comp., 35 (1980), pp. 1251-1268.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[8] I. Gustafsson, *A class of first order factorization methods*, BIT, 18 (1978), pp. 142-156.

[9] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, New York, 1981.

[10] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409-436.

[11] C. Lanczos, *An iterative method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255-282.

[12] J. G. Lewis, *Algorithms for sparse matrix eigenvalue problems*, Ph.D. thesis, Stanford University, Stanford, CA, 1977.

[13] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148-162.

[14] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 154-156 (1991), pp. 289-309.

[15] R. B. MORGAN, *Davidson's method and preconditioning for generalized eigenvalue problems*, J. Comput. Phys., 89 (1990), pp. 241-245.

[16] ———, *Preconditioning eigenvalue problems*, Ph.D. thesis, University of Texas, Austin, TX, 1986.

[17] R. B. MORGAN AND D. S. SCOTT, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 817-825.

[18] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617-629.

[19] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[20] D. S. SCOTT, *The advantages of inverted operators in Rayleigh-Ritz approximations*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 68-75.

[21] ———, *Solving sparse symmetric generalized eigenvalue problems without factorization*, SIAM J. Numer. Anal., 18 (1981), pp. 102-110.

[22] D. B. SZYLD, *Criteria for combining inverse and Rayleigh quotient iteration*, SIAM J. Numer. Anal., 25 (1988), pp. 1369-1375.