SIAM J. NUMER. ANAL.
Vol. 19, No. 6, December 1982

0036-1429/82/1906-0010 $01.00/0

A TRACE MINIMIZATION ALGORITHM FOR THE GENERALIZED EIGENVALUE PROBLEM*

AHMED H. SAMEH† AND JOHN A. WISNIEWSKI‡

Abstract. An algorithm for computing a few of the smallest (or largest) eigenvalues and associated eigenvectors of the large sparse generalized eigenvalue problem $Ax = \lambda Bx$ is presented. The matrices A and B are assumed to be symmetric, and haphazardly sparse, with B being positive definite. The problem is treated as one of constrained optimization and an inverse iteration is developed which requires the solution of linear algebraic systems only to the accuracy demanded by a given subspace. The rate of convergence of the method is established, and a technique for improving it is discussed. Numerical experiments and comparisons with other methods are presented.

1. Introduction. In this paper, we consider the problem of computing a few of the smallest (or largest) eigenvalues and eigenvectors of the large, sparse, generalized eigenvalue problem

$$Ax = \lambda Bx, \tag{1.1}$$

where x is an n-vector, λ is a scalar and A, B are $n \times n$ symmetric matrices, with B being positive definite. We are interested in the case where the matrices A and B are very large, very sparse and have no general pattern of nonzeros. In this case, factorization of either matrix would be impractical. Problems of this type arise in certain structural mechanics [AnIZ68], [CrBa68], [BaWi72], [BaWi73] and plasma physics applications [GrGJ76], [ChGG78], [Grub78].

Although considerable work has been done on this problem [Stew76], no efficient method for simultaneously obtaining several eigenvalues and eigenvectors is available. For example, generalizations of the block Lanczos method of Golub and Underwood [GoUn77] and the method of simultaneous iteration [Baue57], [Ruti69] and [Ruti70] suffer from the disadvantage that several systems of linear algebraic equations of the form $Bx = f$ must be solved accurately at each iteration. While our method will also require the solution of systems of linear algebraic equations, we need only to solve these systems to the accuracy demanded by certain subspaces.

Other methods that can be used for solving this problem employ iterative schemes for minimizing the Rayleigh quotient (x^TAx/x^TBx) so as to obtain the smallest eigenvalue. Hence they must rely on deflation in order to obtain several of the smallest eigenvalues and the corresponding eigenvectors. Such deflation processes often break down numerically, especially when the number of required eigenvectors increases. This is particularly true in the case of clusters (poorly separated eigenvalues). One of the most effective methods of this kind is that of Geradin [Gera71], which uses the conjugate gradient algorithm for minimizing the Rayleigh quotient. Unfortunately, the overall work required by this method is quite high, since as many as $5n$ iterations per eigenvalue are needed to obtain the corresponding eigenvector to full accuracy.

Another approach, that of Jensen [Jens72], uses the idea of sectioning. Again, it is not suitable for our problem since it requires repeated accurate solution of systems of n linear algebraic equations involving the matrix B. The method of coordinate

* Received by the editors March 17, 1981, and in final revised form January 15, 1982. This work was supported in part by the National Science Foundation under grant US NSF MCS 79-18394 and by the U.S. Department of Energy.

† Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.

‡ Division 2113, Sandia National Laboratories, Albuquerque, New Mexico 87185.

relaxation [FaFa63], [Kaha66], [Ruhe74] and [Schw74] computes only a single eigenvalue; deflation must be used if more than one eigenvalue is sought. The method has the serious disadvantage of having severe convergence restrictions similar to those of SOR methods for solving systems of linear algebraic equations [Schw74]. Furthermore, it uses a relaxation parameter which can be difficult to estimate for general problems.

Recently, Longsine and McCormick [LoMc80] introduced two methods (SIRQIT,SIRQIT-CG) which avoid many of the above drawbacks by simultaneously minimizing several Rayleigh quotients $x_i^TAx_i/x_i^TBx_i$, $i=1, 2, \cdots, p$. Their methods, however, suffer from slow asymptotic rates of convergence, and global convergence is proven only for x_1, when $B=I$. Since their methods are closely related to ours, we discuss them in detail in § 4.

Throughout this paper we use the notation of Householder [Hous64]. We assume that upper case letters, both Roman and Greek, represent matrices. We take lower case Roman letters to represent vectors and lower case Greek letters to represent scalars. The two-norm will be used throughout unless otherwise stated. We also use e_j to denote the jth column of the identity matrix.

We now present some theorems which are fundamental in establishing the algorithm of § 2.

THEOREM 1.1 [Fran68, p. 106]. *Let A and B be symmetric $n \times n$ matrices. If B is positive definite then there is an $n \times n$ matrix Z for which*

$$Z^TBZ = I_n \quad \textit{and} \quad Z^TAZ = \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n), \tag{1.2}$$

where $\lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_n$ are the eigenvalues of problem (1.1) *and the columns of Z are their associated eigenvectors. Furthermore, if A is positive definite, then all of the eigenvalues λ_i are positive.*

THEOREM 1.2. *Let A and B be as given in Theorem* 1.1 *and Y^* be the set of all $n \times p$ matrices Y for which $Y^TBY = I_p$. Then*

$$\min_{Y \in Y^*} \operatorname{tr}(Y^TAY) = \sum_{i=1}^{p} \lambda_i. \tag{1.3}$$

In other words,

$$\min_{Y \in Y^*} \operatorname{tr}(Y^TAY) = \operatorname{tr}(X^TAX)$$

with

$$X^TBX = I_p \quad \textit{and} \quad X^TAX = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_p),$$

where X corresponds to the first p columns of the matrix Z of Theorem 1.1.

Proof. The proof follows directly from Theorem 1.1 and the Courant–Fischer theorem [Wilk65, pp. 99–101]. □

The following useful theorem due to Ostrowski [Ostr60] will allow us to obtain bounds on the eigenvalues of a symmetric matrix $M = Y^TAY$ in terms of the eigenvalues of A and those of the matrix Y^TY.

THEOREM 1.3. *Let A be a symmetric matrix of order n with eigenvalues $\lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_n$, of which π_A are positive and ν_A are negative. Let $M = Y^TAY$ for some $n \times p$ matrix Y, $p < n$. Let M have eigenvalues $\mu_1 \leqq \mu_2 \leqq \cdots \leqq \mu_p$, with π_M positive and ν_M negative eigenvalues. Then $\pi_M < \pi_A$ and $\mu_M < \mu_A$. Furthermore, if we denote the positive eigenvalues of A and M respectively by*

$$0 < \lambda_1^+ \leqq \lambda_2^+ \leqq \cdots \leqq \lambda_{\pi_A}^+ \quad \textit{and} \quad 0 < \mu_1^+ \leqq \mu_2^+ \leqq \cdots \leqq \mu_{\pi_M}^+$$

and the negative eigenvalues by

$$\lambda_1^- \leqq \lambda_2^- \leqq \cdots \leqq \lambda_{\nu_A}^- < 0 \quad \textit{and} \quad \mu_1^- \leqq \mu_2^- \leqq \cdots \leqq \mu_{\nu_M}^- < 0,$$

then the following inequalities hold:

$$\frac{\mu^+_{\pi_M - j}}{\lambda^+_{\pi_A - j}} \leqq \rho, \qquad j = 0, 1, \cdots, \pi_M - 1,$$

$$\frac{\mu^-_{\nu_M - j}}{\lambda^-_{\nu_A - j}} \leqq \rho, \qquad j = 0, 1, \cdots, \nu_M - 1,$$

where ρ is the largest eigenvalue of the symmetric positive semidefinite matrix Y^TY.

2. Computing eigenvalues by trace minimization.

We present an algorithm for obtaining a few of the largest or smallest eigenvalues and the corresponding eigenvectors of the generalized eigenvalue problem (1.1). For the moment we require the additional assumption that the matrix A be positive definite. If A is indefinite, problem (1.1) is replaced by

$$(A - \nu B)x = (\lambda - \nu)Bx,$$

where $\nu < \lambda_1 < 0$, thus assuring that $(A - \nu B)$ is positive definite.

The algorithm developed in this section is motivated by Theorem 1.2.

DEFINITION. An $n \times p$ matrix Y forms a *section* of the eigenvalue problem (1.1) if

$$Y^TAY = \Sigma, \qquad Y^TBY = I_p, \tag{2.1}$$

with $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \cdots, \sigma_p)$.

Our approach is to find a sequence of iterates $Y_{k+1} = F(Y_k)$, where both Y_k and Y_{k+1} form a section of (1.1), with $\operatorname{tr}(Y_{k+1}^TAY_{k+1}) < \operatorname{tr}(Y_k^TAY_k)$. From Theorem 1.2, the matrix Y in (2.1) which minimizes $\operatorname{tr}(Y^TAY)$ is that matrix consisting of the eigenvectors associated with the p smallest eigenvalues of problem (1.1). We will choose $F(Y)$ in such a way that the global convergence of the process is assured. In order to achieve a descent step at every iteration, we treat problem (1.1) as the quadratic minimization problem

$$\text{minimize } \operatorname{tr}(Y^TAY)$$

subject to the constraints

$$Y^TBY = I_p. \tag{2.2}$$

THEOREM 2.1. *Let $K = S^THS$ be a $p \times p$ matrix, where S is $n \times p$ and of rank p, and H is an $n \times n$ positive definite matrix. Then* $\operatorname{tr}(K) \leqq \rho \operatorname{tr}(H)$, *where ρ is the largest eigenvalue of S^TS.*

Proof. Since K is positive definite, we write the eigenvalues of H and K as

$$0 < \lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_n, \qquad 0 < \tau_1 \leqq \tau_2 \leqq \cdots \leqq \tau_p$$

respectively. From Ostrowski's theorem (Theorem 1.3), we see that

$$\tau_{p-j} \leqq \rho\lambda_{n-j}, \qquad j = 0, 1, \cdots, p-1,$$

where ρ is the largest eigenvalue of S^TS, and the result readily follows. □

2.1. The basic algorithm. Let Y_k be an $n \times p$ matrix approximating the p eigenvectors corresponding to the smallest p eigenvalues of (1.1), i.e.,

$$Y_k^T A Y_k = \Sigma_k = \operatorname{diag}(\sigma_1^{(k)}, \sigma_2^{(k)}, \cdots, \sigma_p^{(k)})$$

and

$$Y_k^T B Y_k = I_p. \tag{2.3}$$

Construct the matrix

$$Y_{k+1} = (Y_k + \Delta_k) S_k, \tag{2.4}$$

where S_k and Δ_k are chosen such that

$$Y_{k+1}^T A Y_{k+1} = \Sigma_{k+1} = \operatorname{diag}(\sigma_1^{(k+1)}, \sigma_2^{(k+1)}, \cdots, \sigma_p^{(k+1)}),$$

$$Y_{k+1}^T B Y_{k+1} = I_p \tag{2.5}$$

and

$$\operatorname{tr}(Y_{k+1}^T A Y_{k+1}) < \operatorname{tr}(Y_k^T A Y_k). \tag{2.6}$$

2.1.1. Choice of S_k and Δ_k. Naturally, we consider only corrections Δ_k, in (2.4), which are in the tangent subspace of the set of constraints $Y_k^T B Y_k = I_p$; see [Luen73, Chaps. 10 and 11]. This implies that

$$\Delta_k^T B Y_k = 0.$$

This condition is satisfied if we choose

$$\Delta_k = (I - P_k) Z_k, \tag{2.7}$$

where Z_k is an $n \times p$ matrix yet to be specified and

$$P_k = B Y_k (Y_k^T B^2 Y_k)^{-1} Y_k^T B \tag{2.8}$$

is the projector onto the subspace spanned by the columns of BY_k, i.e. $PBY_k = BY_k$.

If we let $\hat{Y} = (Y_k + \Delta_k)$, then S_k is chosen simply as the matrix of eigenvectors for the $p \times p$ generalized eigenvalue problem $\hat{Y}^T A \hat{Y} = \lambda \hat{Y}^T B \hat{Y}$.

To see how S_k is determined, from the two $p \times p$ matrices

$$\hat{Y}^T A \hat{Y} = \Sigma_k + Y_k^T A \Delta_k + \Delta_k^T A Y_k + \Delta_k^T A \Delta_k \tag{2.9}$$

and

$$\hat{Y}^T B \hat{Y} = I_p + \Delta_k^T B \Delta_k. \tag{2.10}$$

Observing that (2.10) is a positive definite matrix, it has the spectral decomposition

$$\hat{Y}^T B \hat{Y} = U D^2 U^T,$$

where U is orthogonal and $D^2 = \operatorname{diag}(\delta_1^2, \cdots, \delta_p^2)$. Note that $\delta_i^2 \geqq 1$; see [Wilk65, pp. 101–102].

Hence, $D^{-1} U^T (\hat{Y}^T B \hat{Y}) U D^{-1} = I_p$. Now, we can construct an orthogonal matrix V such that

$$V^T D^{-1} U^T (\hat{Y}^T A \hat{Y}) U D^{-1} V = \Sigma_{k+1}.$$

In other words,

$$S_k = U D^{-1} V. \tag{2.11}$$

Therefore,

$$\begin{aligned}\operatorname{tr}(Y_{k+1}^T A Y_{k+1}) &= \operatorname{tr}[V^T(D^{-1}U^T\hat{Y}^T A\hat{Y}UD^{-1})V]\\ &= \operatorname{tr}[D^{-1}(U^T\hat{Y}^T A\hat{Y}U)D^{-1}] \leqq \operatorname{tr}(\hat{Y}^T A\hat{Y}),\end{aligned}$$

where we have used Theorem 2.1 and the fact that $\delta_i^2 \geqq 1$.

Consequently, we see that the scaling matrix S_k is chosen so that Y_{k+1} forms a new section (2.1) of problem (1.1), while guaranteeing that $\operatorname{tr}(Y_{k+1}^T A Y_{k+1})$ does not increase. We will see, however, that while the matrix S_k is chosen to maintain the B-orthonormality and A-orthogonality of the iterates, the matrix Δ_k plays a crucial role. It is chosen to yield improved approximations to the eigenvectors of problem (1.1) by reducing $\operatorname{tr}(\hat{Y}^T A\hat{Y})$. We now concentrate on the choice of the matrix Δ_k.

2.2. Choice of Δ_k as a steepest descent step. A reasonable choice for Z_k, (see (2.7)) for which inequality (2.6) holds, is $Z_k = AY_kW_k$, i.e.,

$$\Delta_k = (I - P_k)AY_kW_k, \tag{2.12}$$

where $W_k = \operatorname{diag}(\omega_1^{(k)}, \omega_2^{(k)}, \cdots, \omega_p^{(k)})$ is yet to be determined. It is interesting to note that the matrix Δ_k can also be written as

$$\Delta_k = (I - P_k)R_kW_k,$$

where $R_k = AY_k - BY_k\Sigma_k$ is the residual matrix of problem (1.1) corresponding to Y_k. This follows from the fact that $(I - P_k)BY_k = 0$. As a result, the matrix Δ_k can either be treated as a scaled projection of the kth residual, or as a scaled projection of the gradient of $\operatorname{tr}(Y_k^T A Y_k)$.

Now, from (2.12) and (2.9) we have

$$\operatorname{tr}(\hat{Y}^T A\hat{Y}) = \operatorname{tr}(\Sigma_k) + 2\operatorname{tr}(C_kW_k) + \operatorname{tr}(E_kW_k^2),$$

where

$$C_k = Y_k^T A(I - P_k)AY_k \quad \text{and} \quad E_k = Y_k^T A(I - P_k)A(I - P_k)AY_k.$$

Consequently, we wish to determine W_k so that

$$\phi_k = \operatorname{tr}(2C_kW_k + E_kW_k^2)$$

is minimized. This is achieved simply by choosing

$$\omega_i^{(k)} = -\gamma_{ii}^{(k)}/\varepsilon_{ii}^{(k)}, \qquad i = 1, 2, \cdots, p, \tag{2.13}$$

where $\gamma_{ii}^{(k)}$ and $\varepsilon_{ii}^{(k)}$ are the diagonal elements of the matrices C_k and E_k, respectively. Note that $\varepsilon_{ii}^{(k)} \geqq 0$. If $\varepsilon_{jj}^{(k)} = 0$ for any j, i.e., when Y_ke_j is an eigenvector of (1.1), we simply choose $\omega_j^{(k)}$ to be zero also. Hence,

$$\min_{W_k} \phi_k = -\sum_{\substack{i=1\\ \varepsilon_{ii}^{(k)} \neq 0}}^{p} [\gamma_{ii}^{(k)2}/\varepsilon_{ii}^{(k)}]$$

and

$$\operatorname{tr}(Y_{k+1}^T A Y_{k+1}) \leqq \operatorname{tr}(\hat{Y}^T A\hat{Y}) < \operatorname{tr}(\Sigma_k) = \operatorname{tr}(Y_k^T A Y_k). \tag{2.14}$$

2.3. The choice of Δ_k as an optimal subspace iterate. In this section we treat the formulation (2.2) of problem (1.1) by again considering corrections Δ in the subspace orthogonal to the constraints

$$Y^T B Y = I_p.$$

Thus, we consider an iteration of the form

$$\bar{Y} = Y - \Delta, \tag{2.15}$$

where Δ is chosen so as to

$$\begin{aligned} &\text{minimize tr } (Y-\Delta)^T A (Y-\Delta), \\ &\text{subject to } Y^T B \Delta = 0. \end{aligned} \tag{2.16}$$

Since the matrix A is positive definite (by assumption), we consider instead the equivalent problem:

$$\begin{aligned} &\text{minimize } (y_j - d_j)^T A (y_j - d_j), \\ &\text{subject to } Y^T B d_j = 0, j = 1, 2, \cdots, p, \end{aligned} \tag{2.17}$$

where $d_j = \Delta e_j$.

The problem in (2.17) can also be written as

$$\begin{aligned} &\text{minimize } \|b_j - A^{1/2} y_j\| \\ &\text{subject to } Y^T B A^{-1/2} b_j = 0, j = 1, 2, \cdots, p. \end{aligned} \tag{2.18}$$

where $b_j = A^{1/2} d_j$. The solution to this linear least squares problem is obtained by setting $A^{1/2} y_j - b_j$ equal to the orthogonal projection of $A^{1/2} y_j$ onto the space spanned by $A^{-1/2} B Y$ (see, for example, [LaHa74, Chaps. 20–22]). Consequently,

$$A^{1/2} y_j - b_j = A^{-1/2} B Y (Y^T B A^{-1} B Y)^{-1} Y^T B y_j$$

and

$$y_j - d_j = A^{-1} B Y (Y^T B A^{-1} B Y)^{-1} Y^T B y_j.$$

Hence, since $Y^T B Y = I$ we have

$$\bar{Y} = Y - \Delta = A^{-1} B Y (Y^T B A^{-1} B Y)^{-1}. \tag{2.19}$$

From (2.19), it is easy to see that the optimal subspace iterate is equal to the subspace obtained by one step of simultaneous iteration [Ruti69]. Our iteration differs from that of Rutishauser's only by the presence of the $p \times p$ scaling matrix $(Y^T B A^{-1} B Y)^{-1}$. In other words, iteration (2.19) generates exactly the same sequence of subspaces as simultaneous iteration for computing the largest eigenvalues of the problem $A^{-1} B u = \mu u$.

In fact, both the global convergence and rate of convergence of (2.19) follow directly from those of simultaneous iteration, [Ruti69]. Since each column of $\bar{Y}$ and Y can be expressed as a linear combination of the eigenvectors of (1.1), i.e.,

$$\bar{Y} = Z\bar{G} \quad \text{and} \quad Y = ZG,$$

where Z is the eigenvector matrix, and since as $Y \to X$, $G^T \to (I_p, 0)$, we write

$$G = \begin{bmatrix} I_p \\ 0 \end{bmatrix} + F = \begin{bmatrix} I_P + F_1 \\ F_2 \end{bmatrix},$$

and

$$\bar{G} = \begin{bmatrix} I_P \\ 0 \end{bmatrix} + \bar{F} = \begin{bmatrix} I_P + \bar{F}_1 \\ \bar{F}_2 \end{bmatrix}.$$

Note that the columns of F and $\bar{F}$ are vectors representing the errors in the associated eigenvector approximations Y and $\bar{Y}$, respectively. We now state the main result.

THEOREM 2.2. *Let A and B be symmetric positive definite matrices and assume that the eigenvalues of problem* (1.1) *satisfy* $0<\lambda_1\leqq\lambda_2\leqq\cdots\leqq\lambda_p<\lambda_{p+1}\leqq\cdots\leqq\lambda_n$. *Also, let the initial iterate Y of the algorithm be chosen such that it has linearly independent columns and that the corresponding matrix F is not deficient in any eigencomponent. Then, column j of Y, y_j, globally converges to the eigenvector x_j corresponding to λ_j for $j=1,2,\cdots,p$, with an asymptotic rate of convergence less than or equal to $|\lambda_j/\lambda_{p+1}|$. That is, we have*

$$\|\bar{F}e_j\|\leqq|\lambda_j/\lambda_{p+1}|\,\|Fe_j\|+O(\|F\|^2).$$

The initial choice of Y as a set of vectors whose elements are randomly chosen, in general, will satisfy the assumption that Y is of rank p.

Since the most time-consuming computation in the algorithm is that of solving problem (2.17), an efficient method of obtaining d_j, $j=1,2,\cdots,p$, is essential. It is on this matter that we now focus our attention.

Using Lagrange multipliers and dropping the subscripts from (2.17), we see that any of the systems (2.17) is equivalent to the system of linear equations

$$\begin{bmatrix} A & BY \\ Y^TB & 0 \end{bmatrix}\begin{bmatrix} d \\ l \end{bmatrix}=\begin{bmatrix} Ay \\ 0 \end{bmatrix}, \tag{2.20}$$

where $2l$ is a vector of order p representing the Lagrange multipliers. Writing BY in terms of its orthogonal factorization (note that rank $(BY)=p$),

$$BY=QR=[Q_1,Q_2]R,$$

where $R^T=[R'^T,0]$, in which R' is upper triangular of order p, and Q is orthogonal with $Q_1\in R^{n\times p}$, we obtain the equivalent system

$$\begin{bmatrix} Q^TAQ & R \\ R^T & 0 \end{bmatrix}\begin{bmatrix} g \\ l \end{bmatrix}=\begin{bmatrix} f \\ 0 \end{bmatrix} \tag{2.21}$$

with $g=Q^Td$ and $f=Q^TAy$. Writing $g^T=[g'^T,\tilde{g}^T]$, where g' is a vector of order p, the equation $R^Tg=0$, and the nonsingularity of R' implies that $g'=0$. Consequently, (2.22) reduces to

$$Q^TAQ\begin{bmatrix} 0 \\ \tilde{g} \end{bmatrix}+\begin{bmatrix} R'l \\ 0 \end{bmatrix}=f.$$

Hence, we obtain

$$Q_1^TAQ_2\tilde{g}+R'l=Q_1^TAy \tag{2.22}$$

and

$$Q_2^TAQ_2\tilde{g}=Q_2^TAy. \tag{2.23}$$

Since we are only interested in solving (2.20) for d, we need only solve the positive definite system (2.23) for $\tilde{g}$. Since

$$d=Qg=Q_2\tilde{g},$$

the conjugate gradient method (version 2 of [Reid71]) for solving the system (2.23) is given by:

$$\tilde{r}_0=Q_2^TAy-Q_2^TAQ_2\tilde{g}_0,$$

$$\tilde{p}_0=\tilde{r}_0,$$

for $k=0,1,2,\cdots$,

$$\tilde{g}_{k+1}=\tilde{g}_k+\tilde{\alpha}_k\tilde{p}_k,$$
$$\tilde{\alpha}_k=\tilde{r}_k^T\tilde{r}_k/\tilde{p}_k^TQ_2^TAQ_2\tilde{p}_k,$$
$$\tilde{r}_{k+1}=\tilde{r}_k-\tilde{\alpha}_kQ_2^TAQ_2\tilde{p}_k,$$
$$\tilde{p}_{k+1}=\tilde{r}_{k+1}+\tilde{\beta}_k\tilde{p}_k,$$
$$\tilde{\beta}_k=\tilde{r}_{k+1}^T\tilde{r}_{k+1}/\tilde{r}_k^T\tilde{r}_k.$$

Observing that $(I-P)\equiv(I-Q_1Q_1^T)=Q_2Q_2^T$, the CG algorithm is transformed into the original coordinates by the relations

$$r_k=Q_2\tilde{r}_k,\quad d_k=Q_2\tilde{g}_k,\quad p_k=Q_2\tilde{p}_k.$$

Making the convenient choice $\tilde{g}_0=0$ and forming the iterates $y-d_k$ directly instead of only d_k, results in the following:

$$r_0=(I-P)Ay_0,\qquad y_0=y,$$
$$p_0=r_0,$$

for $k=0,1,2,\cdots$,

$$y_{k+1}=y_k-\alpha_kp_k,$$
$$\alpha_k=r_k^Tr_k/p_k^TAp_k,$$
$$r_{k+1}=r_k-\alpha_k(I-P)Ap_k,$$
$$p_{k+1}=r_{k+1}+\beta_kp_k,$$
$$\beta_k=r_{k+1}^Tr_{k+1}/r_k^Tr_k.$$

The relationship between the choice of Δ_k as a steepest descent step and Δ_k as an optimal subspace iterate is now clear. The choice of Δ_k as a steepest descent results in executing the above CG algorithm only for the case $k=0$, whereas the computation of Δ_k as an optimal subspace iterate would require execution of the CG algorithm until the solution to the system (2.23) is obtained. The two choices represent opposite extremes in deciding how much the trace should be reduced in the current subspace.

At this point we should mention that (2.19) suggests an alternative algorithm, namely, the use of Rutishauser's simultaneous iteration to obtain a few of the largest eigenvalues of $A^{-1}B$ employing the method of conjugate gradients when performing the matrix-vector products $(A^{-1}B)z=y$. Numerical experiments have shown, however, that such an algorithm yields results comparable with ours only if the linear systems $Ay=Bz$ are solved quite accurately, which can be very time consuming. In § 2.4 we present a criterion for terminating the CG process which makes the trace minimization approach even more economical.

2.4. Terminating the CG process. In this section, we take advantage of the error analysis of the CG algorithm (see, for example [Luen73]) and the error analysis of iteration (2.19), Theorem 2.2, in deriving a useful stopping criterion for the CG process.

DEFINITION. Let the error function $\tilde{\varepsilon}(\tilde{g}_m)$ of the mth step of the CG algorithm applied to the system (2.23) be given by

$$\tilde{\varepsilon}(\tilde{g}_m)=(\tilde{g}_m-\tilde{g}^*)^TQ_2^TAQ_2(\tilde{g}_m-\tilde{g}^*),$$

where $\tilde{g}^*$ is the unique solution to (2.23). This gives the equivalent error function (in the original coordinates):

$$\varepsilon(y_m)=(y_m-y^*)^TA(y_m-y^*),$$

where $y^*=y-d^*$, $d^*=Q_2\tilde{g}^*$.

We now quote a relevant theorem concerning the errors in the CG algorithm.

THEOREM 2.3 [Luen73, p. 187]. *For the* CG *algorithm*

$$\varepsilon(y_m)\leqq 4\left[\frac{1-\kappa^{-1/2}}{1+\kappa^{-1/2}}\right]^{2m}\varepsilon(y_0),$$

where κ is the condition number of $Q_2^TAQ_2$.

From Theorem 2.2, we see that

$$\bar{f}_j^TA\bar{f}_j\leqq(\lambda_j/\lambda_{p+1})^2f_j^TAf_j+O(\|F\|^3),$$

where f_j and $\bar{f}_j$ represent the errors in the eigenvector approximations (column j of Y) for two successive iterates. The above limitation in accuracy is of course due to the fact that we are restricting our next eigenvector approximations to the space orthogonal to BY. Consequently, even if we were to compute the right-hand sid*r* of (2.19) exactly, we could expect no more accuracy in our iterates than that given by Theorem 2.2. In light of this, Theorem 2.3 and the fact that our iterates for the CG process are restricted to the space orthogonal to BY, any contraction of the error in a given iterate of the process smaller than $(\lambda_j/\lambda_{p+1})^2$, will gain us little (if any) additional accuracy in the eigenvector approximations for the given subspace. As a result, we choose our step number m for the CG method as that smallest integer for which

$$\varepsilon(y_j^{(m)})\leqq(\lambda_j/\lambda_{p+1})^2\varepsilon(y_j^{(0)}). \tag{2.24}$$

The quantity $\varepsilon(y_j^{(k)})$ may be estimated by

$$\hat{\varepsilon}(y_j^{(k)})=(y_j^{(k)}-y_j^{(k+1)})^TA(y_j^{(k)}-y_j^{(k+1)}),$$

which can be easily obtained from the CG algorithm in § 2.3 via one multiplication $(\alpha_k\|r_k\|^2)$. This underestimation of the error is compensated for by the fact that we use the ratio (σ_j/σ_{p+1}) as an estimate for $(\lambda_j/\lambda_{p+1})$, where in general $\sigma_j/\sigma_{p+1}<\lambda_j/\lambda_{p+1}$, especially where σ_j is sufficiently close to λ_j. It should be noted that an extra column vector $(p+1)$ is needed for the computation.

3. Computational aspects. In this section, we consider several techniques which are necessary for a successful computer implementation of our algorithm. We begin by treating the details associated with the incorporation of Ritz shifts into the algorithm.

The acceleration method that we have chosen to incorporate into our algorithm is analogous to a Rayleigh quotient iteration scheme (e.g., see [Parl80, pp. 70–80]). The method consists of solving the sequence of eigenvalue problems

$$(A-\sigma_j^{(k)}B)x_j=\bar{\lambda}_j^{(k)}Bx_j,\qquad j=1,2,\cdots,p. \tag{3.1}$$

These problems have the same eigenvectors as those of problem (1.1), and their eigenvalues are simply given by

$$\bar{\lambda}_j^{(k)}=\lambda_j-\sigma_j^{(k)},$$

where λ_j is the jth eigenvalue of problem (1.1). The shift parameters $\sigma_j^{(k)}$ are given by (2.3), the Ritz approximations for the kth iteration of our algorithm. Wilkinson

[Wilk72] has shown that convergence of the Rayleigh quotient iteration to an eigenvalue λ is ensured if the Ritz approximation is sufficiently close to λ. Furthermore, the rate of convergence is cubic.

Clearly, the global convergence of our algorithm (discussed in § 2.3) is not preserved. So, it is at this point that we also relax our assumption regarding the positive definiteness of A. From the fact that $\sigma_j^{(k+1)} < \sigma_j^{(k)}$ for all k, i.e., the Ritz approximations for the algorithm approach the eigenvalues of problem (1.1) from above and from the well-known theorem:

THEOREM 3.1 [Parl80, p. 318]. *For an arbitrary nonzero vector u and scalar σ, there is an eigenvalue λ of* (1.1) *such that*

$$|\lambda-\sigma| \leqq \|(A-\sigma B)u\|_{B^{-1}}/\|Bu\|_{B^{-1}}.$$

We have developed a heuristic strategy that maintained global convergence successfully for a variety of test problems [Wisn81]. This strategy may be outlined as follows:

(i) $\sigma_j^{(k)} = \sigma_1^{(k)}$, for $1 \leqq j \leqq p$, whenever $\sigma_1^{(k)} < 0$,

(ii) the CG process is terminated whenever we encounter a nondescent step and

(iii) σ_l is used as a shift for column j, $l < j$, only if $\sigma_l < \lambda_j$.

This is assured when

$$\sigma_l - \sigma_j < \lambda_j - \sigma_j < -\|r_j\|_{B^{-1}}, \tag{3.2}$$

where $r_j = Ay_j - \sigma_j B y_j$.

The algorithm is best accelerated when we can use σ_j as the shift for column j. That the resulting method has a cubic rate of convergence is a consequence of the following theorem.

THEOREM 3.2. *For $j \leqq p$, $\lambda_j - \sigma_j^{(k)} = -e_j^T F^T \Phi_j F e_j$, where $\Phi_j = \Lambda - \lambda_j I$.*

Proof. Let Y_k be given by (2.3), writing $Y_k = ZG$; then from (1.2) we get $G^T G = I_p$ with

$$G = \begin{bmatrix} I_p \\ 0 \end{bmatrix} + F.$$

Consequently,

$$e_j^T G^T G e_j = 1,$$

or

$$e_j^T e_j + 2e_j^T F e_j + e_j^T F^T F e_j = 1,$$

hence $e_j^T F^T F e_j = -2e_j^T F e_j$.

Furthermore,

$$\begin{aligned} y_j^{(k)T} A y_j^{(k)} = e_j^T G^T \Lambda G e_j &= e_j^T \Lambda e_j + 2\lambda_j e_j^T F e_j + e_j^T F^T \Lambda F e_j \\ &= \lambda_j - \lambda_j e_j^T F^T F e_j + e_j^T F^T \Lambda F e_j \\ &= \lambda_j + e_j^T F^T \Phi_j F e_j, \end{aligned}$$

from which the theorem follows. □

The efficiency of the algorithm crucially depends on the shifting strategy employed. If we shift column j by σ_j too late, the algorithm becomes inefficient in the sense that we take several steps of the algorithm at a slower (linear) rate of convergence when a much better (cubic) rate is possible. On the other hand, if we shift column j by σ_j too soon, the objective function (2.17), with shifted A, increases rather than decreases and global convergence is lost. Step (ii) in our heuristic strategy attempts to prevent such loss of global convergence, and step (iii) avoids needless delay in shifting with the proper Ritz approximation.

In the presence of shifting, it is important to estimate the error reduction factor used in terminating the CG process of § 2.4. The error test (2.24) for the CG process now becomes

$$\varepsilon(y_j^{(m)}) \leqq (\bar{\lambda}_j^{(k)}/\bar{\lambda}_{p+1}^{(k)})^2 \varepsilon(y_j^{(0)}), \tag{3.3}$$

where $\bar{\lambda}_j^{(k)} = \lambda_j - \sigma_j^{(k)}$. From Theorem 3.2,

$$\sigma_j^{(k+1)} - \sigma_j^{(k)} = e_j^T \bar{F}^T \Phi_j \bar{F} e_j - e_j^T F^T \Phi_j F e_j \cong -e_j^T F^T \Phi_j F e_j,$$

where we have used the fact that with shifting, $\|\bar{F}\| = O(\|F\|^3)$. Hence,

$$\bar{\lambda}_j^{(k)} \cong \sigma_j^{(k+1)} - \sigma_j^{(k)}. \tag{3.4}$$

Assuming that $\lambda_{p+1} \lesssim \sigma_p^{(0)}$, we approximate $\bar{\lambda}_{p+1}^{(k)}$ by

$$\bar{\lambda}_{p+1}^{(k)} \cong \sigma_p^{(0)} - \sigma_j^{(k)}. \tag{3.5}$$

Such an assumption is reasonable especially in the case $\lambda_p \leqq \lambda_{p+1} \ll \lambda_n$. In any event, approximation (3.5) is not as critical as (3.4). Now, the termination criterion (3.3) can be written as

$$\hat{\varepsilon}(y_j^{(m)}) \lesssim \left[\frac{\sigma_j^{(k+1)} - \sigma_j^{(k)}}{\sigma_p^{(0)} - \sigma_j^{(k)}}\right]^2 \hat{\varepsilon}(y_j^{(0)}), \tag{3.3$'$}$$

where $\hat{\varepsilon}(y_j^{(k)}) = (y_j^{(k)} - y_j^{(k+1)})^T (A - \sigma_j^{(k)} B)(y_j^{(k)} - y_j^{(k+1)})$. Our numerical experiments have shown that the approximate criterion (3.3)′ works well in practice. In addition to the termination criterion (3.3)′, the CG iterations are also terminated if

$$\hat{\varepsilon}(y_j^{(m)}) \leqq 100\|A + \sigma_j^{(k)} B\| \eta^2,$$

where η is the roundoff error level of the machine. This is done because any increments which are added to $y_j^{(k)}$, $k \geqq m$, will be at or below the roundoff level of the machine.

The analysis so far has dealt only with a single column in the block. Certainly, each column of Y has a different asymptotic rate of convergence. Furthermore, the error reduction for any column depends on the overall accuracy of the iterates in the subspace. We have observed that the overall error reduction is bounded by the maximum reduction given by Theorem 2.2 and the criterion (3.3)′ used to terminate the CG process. Our strategy is then to use an error reduction factor which is the geometric mean of the error reduction factors of the individual columns in the block. This strategy is well balanced in that it avoids performing excessive CG iterations and avoids computing excessively many sections. It has worked well on our test problems.

Additional computational aspects concerning the efficient implementation of our algorithm are found in [Wisn81]. Treated there is the efficient computation of a section, (2.2), and the formation of the projection matrix $(I - P)$. Our method freezes a column of Y once it has converged and requires the user to choose the block size of the method $s > p$, where p is the number of desired eigenvectors.

Our convergence test follows from the observation made by Moler and Stewart [MoSt73] concerning the QZ algorithm. "If an eigenvalue and eigenvector are not too "ill disposed," then they produce a small relative residual." Hence, when all the eigenvector–eigenvalue pairs have small relative residuals, we terminate the iteration process. Furthermore, a column of Y is accepted as a valid eigenvector approximation whenever the CG process is terminated due to corrections d_j that are at the roundoff level of the machine.

Finally, we mention in passing that the above algorithm is ideally suited for array or parallel computers. For example, the CG process (the most time-consuming part of the algorithm) can be implemented on s independent processors, one processor per column of Y. These processors, however, need an interconnection network that would allow the efficient implementation of the following two tasks:

(a) The B-orthonormalization of the columns of Y via the modified Gram–Schmidt process [Rice66], which is needed in computing the projector P of (2.8), and

(b) the evaluation of all the eigenvalues and eigenvectors of a dense $s \times s$ matrix via the "parallel" Jacobi method in [Same71] needed for computing a section (2.2).

Because of the nature of the above two tasks, it turns out that a simple network such as that of the ILLIAC IV computer [Bouk72] is sufficient.

4. Numerical experiments. We have tested our algorithm, which we call TRACMN, on a test set consisting of eight problems. In the first five problems we chose the matrices A and B such that $\kappa(B)$, the spectral condition number of B is 10 and the eigenvalues of the problem (1.1) have the distribution shown in Table 1.

TABLE 1
Eigenvalue distributions and parameters for TRACMN.

Problem	Distribution	N	P	S
1	$(0, 10, 20, 30, 31, \cdots, 36)$	10	3	4
2	$(1, 1.001, \cdots, 1.004, 49.981, 49.982, \cdots, 50)$	25	3	4
3	$(1, 1.001, \cdots, 1.004, 49.981, 49.982, \cdots, 50)$	25	6	7
4	$(1, 1.5, 2, 2.5, 5, 5.5, 6, 6.5, 9, 9.5, \cdots, 24.5)$	40	8	9
5	$(-3, -1, 1, 3, 5, \cdots, 35)$	20	4	5

For problems 6 and 7 we consider the determination of few of the natural frequencies of a simply supported and clamped isotropic plate, respectively. We use the Hermite bicubic finite element discretization of the biharmonic operator on the unit square for problem 6 and on the rectangle $0 \leqq x \leqq \pi$ and $0 \leqq y \leqq 2^{1/4}\pi$ for problem 7. For problem 6 we take $n = 64$, $p = 5$ and $s = 6$ and for problem 7 $n = 64$, $p = 10$ and $s = 15$. Finally, for problem 8 we take in (1.1), $B = I$ and A the five-point difference Poisson operator of order $n = 992$; we also take $p = 2$ and $s = 3$.

Problem 1 is one of the test set used by Longsine and McCormick [LoMc80] to test their algorithms. Problems 2, 3 and 4 are chosen to indicate the performance of our algorithm, that of Longsine and McCormick and the Lanczos algorithm on problems with mild to moderate eigenvalue clustering. Problem 5 is chosen to illustrate the performance of our method when A is indefinite. The condition number of B, $\kappa(B)$, is chosen to be small so as not to favor our method over that of Lanczos. For all problems, we request a relative residual tolerance of 10^{-12}.

In Table 2 we present the numerical results of our algorithm on the test set. The algorithm terminated normally for all problems, indicating that the tolerance criteria were met. Table 2 shows the number of outer iterations (sections) required by the algorithm to obtain the solution, the number of multiplications of the matrices A and B by a vector and the CPU time in seconds. All computations are performed in single precision on the Cyber 175 which has approximately 14 decimal digits of accuracy.

TABLE 2
Numerical results of TRACMN (*with shifting*).

Problem	Iterations	A mults	B mults	Time
1	13	252	207	1.13
2	6	368	333	1.84
3	14	426	373	2.57
4	8	655	573	4.50
5	8	409	379	2.16
6	5	731	731	9.20
7	5	1217	1217	9.77
8	6	726	619	12.59

As stated in § 1, the methods of Longsine and McCormick [LoMc80], SIRQIT and SIRQIT-CG most closely resemble ours in their basic development. It is clear, however, that the end products are different, a point well illustrated by Fig. 1. Their algorithms solve the p problems

$$\text{minimize } (y_i^T A y_i / y_i^T B y_i), \qquad i = 1, 2, \cdots, p.$$

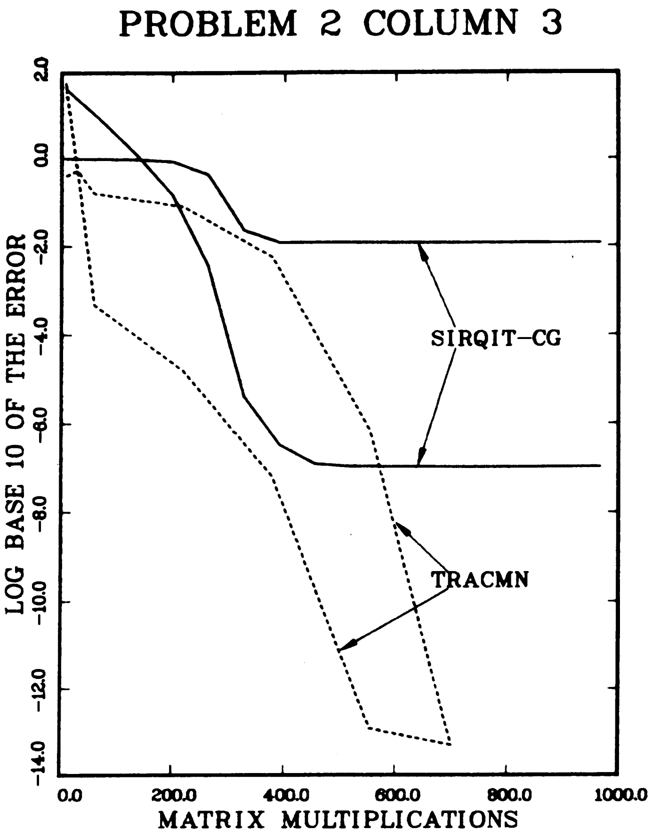

FIG. 1. *Trace minimization* (*dashed line*) *vs.* SIRQIT-CG (*solid line*). *The lower curves represent the errors in the eigenvalue, and the upper curves the error in the corresponding eigenvector.*

At the kth step of the minimization procedure in [LoMc80]

$$y_i^{(k+1)} = y_i^{(k)} + \alpha_i^{(k)} d_i^{(k)}, \qquad i = 1, 2, \cdots, p.$$

In SIRQIT, $d_i^{(k)}$ and $\alpha_i^{(k)}$ are determined by the steepest descent procedure of Hestenes and Karush [HeKa52], and in SIRQIT-CG, these quantities are determined by the CG procedure of Bradbury and Fletcher [BrF166]. It should be noted that the search directions $d_i^{(k)}$ only ensure that

$$y_i^{(k+1)T} B y_i^{(k+1)} = 1, \qquad i = 1, 2, \cdots, p. \tag{4.1}$$

This limited treatment of the constraints, resulting in lack of coupling between the iterates $y_i^{(k)}$, is the main difference between our methods. As a result, they have only been able to prove the global convergence of the first vector in their block to the eigenvector corresponding to $\lambda_{\min}$ of the standard eigenvalue problem $Ax = \lambda x$. It is of interest to note that for a single column and $B = I$, SIRQIT reduces to the method of geodesic descent, see [Luen73, pp. 254–261], with the same rate of convergence. From the implementation point of view, while the number of CG steps required in the inner iteration of TRACMN is determined dynamically, the CG process in SIRQIT-CG is incorporated into the outer iteration and no heuristic is given (other than trial and error) for determining the frequency of recycling the CG process. Our experience indicates that the efficiency of SIRQIT-CG is sensitive to the recycling frequency.

Table 3 illustrates the performance of SIRQIT-CG vs. TRACMN on two of the test problems. The best recycle frequency of SIRQIT-CG is determined experimentally using crude tolerances (a recycle frequency of 1 indicates that SIRQIT, rather than SIRQIT-CG, is executed). The convergence characteristics of TRACMN and SIRQIT-CG are illustrated in Fig. 1. The dashed curves represent the errors associated with TRACMN and the solid lines SIRQIT-CG. Two curves are plotted for each method, the lower curve represents the error in the third smallest eigenvalue and the upper curve the 2-norm of the error in the corresponding eigenvector.

TABLE 3
Performance of SIRQIT-CG [LoMc80] *vs.* TRACMN.

	Problem	Recycle frequency	Time in seconds	A mults	B mults
	1	1	1.65	258	258
SIRQIT-CG	1	2	1.98	399	402
	2	4	69.52	13,094	13,712
TRACMN	1	—	1.13	252	207
	2	—	1.84	368	333

Now we compare the performance of our method with that of the generalized Lanczos process where the conjugate gradient algorithm is used to solve the resulting systems of linear equations of the form $Bz = f$. We consider a typical implementation of the Lanczos process where the tridiagonal matrix is computed without reorthogonalization, and once the eigenvalues of the tridiagonal matrix are computed, we evaluate the corresponding eigenvectors using inverse iteration via subroutine SYMMLQ of Paige and Saunders [PaSa75]. In all problems of our test set we have observed that the linear systems involving B, that are encountered in every step of the Lanczos process, must be solved to high accuracy (a relative residual of about $\eta^{0.75}$, where η is the machine precision) so as to avoid premature introduction of spurious eigenvalues. The Lanczos algorithm can also be used to solve the system $Bz = f$ more efficiently [Parl80a], yielding a drastic reduction in execution time. The method, however, uses secondary storage (disk), thus, we decided not to use it for comparative purposes. Table 4 gives the result of such comparisons for obtaining the required eigenvectors (and eigenvalues) of problems 3 and 4 to the prescribed tolerance. While the execution time is comparable for both methods on problem 4, TRACMN requires less time than the Lanczos algorithm for problem 3. The reason for this is the time consumed by the classification process within the Lanczos algorithm. This is the process of determining when the algorithm has computed the p smallest eigenvalues. In problem 3, the Lanczos algorithm introduces 7 spurious eigenvalues, making it necessary to compute 13 eigenvector approximations to guarantee that the desired 6 eigenvectors are obtained. Recently, Parlett and Reid [PaRe80] have made some progress in handling this classification process for the standard eigenvalue problem.

TABLE 4
Comparison of the Lanczos process without reorthogonalization with TRACMN.

	Problem	Iterations	A mults	B mults	Time in seconds
Lanczos	3	29	796	1,579	4.42
	4	35	448	1,707	4.13
TRACMN	3	14	426	373	2.57
	4	8	655	573	4.50

It is also of interest to note that our algorithm, TRACMN, requires roughly half the number of matrix–vector multiplications required by the Lanczos process. This indicates that for very large problems TRACMN is preferable to the Lanczos method without reorthogonalization.

Alternative implementations of the Lanczos algorithm make use of selective orthogonalization [PaSc79] or periodic reorthogonalization [Grca81]. While these methods practically avoid the introduction of spurious eigenvalues, they must keep the entire set, or a subset, of the Lanczos basis vectors in secondary storage (disk). For very large problems this can be rather impractical. In addition, it is vitally important to solve the linear systems $Bz = f$ to full accuracy since such methods attempt to compute, via reorthogonalization, the exact Lanczos basis for the eigenproblem. The disk I/O must be carefully interleaved with the arithmetic so as to prevent the algorithm from becoming I/O bound. The difficult classification problem is not entirely eliminated. Moreover, unlike our algorithm, TRACMN, eigenvalue multiplicities are not detected unless a block version of the Lanczos method [GoUn77] is used. Since our method uses only high speed memory, we have opted not to compare it with the above, secondary storage intensive, Lanczos methods.

The efficiency of the Lanczos method, in a parallel computing environment, can only be improved by the vectorization of the matrix–vector products. Consequently, the maximum speedup which can be attained over the sequential algorithm is of order n, where n is the problem dimension. The I/O penalties that are incurred, in a parallel computing environment, are much more costly when periodic reorthogonalization is used, since the method clearly will become I/O bound. Our method, on the other hand, utilizes only high speed memory, and in a parallel computing environment, can be vectorized with the minimum of interprocessor communication. Thus, TRACMN can achieve a maximum speedup of order ns, where s is the block size, which is superior to that of the basic Lanczos method. While the block Lanczos method should also achieve a maximum speedup of ns over a sequential version of the same method, as far as the arithmetic is concerned, the interprocessor communication degrades this speedup significantly, see for example [SaSa80], especially when the ns processors are "loosely" interconnected.

Acknowledgments. The first author would like to acknowledge the stimulating discussions with Professors Gene Golub and Alston Householder regarding the generalized eigenvalue problem during a visit to the celebrated Serra House of the Department of Computer Science at Stanford University. We would also like to thank the referee for many valuable suggestions.

REFERENCES

[AnIZ68] R. G. Anderson, B. M. Irons and O. C. Zienkiewicz, *Vibration and stability of plates using finite elements*, Int. J. of Solids and Structures, 4 (1968), pp. 1031–1055.

[Baue57] F. L. Bauer, *Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme*, ZAMP 8, pp. 214–235.

[BaWi72] K. J. Bathe and E. L. Wilson, *Large eigenvalue problems in dynamic analysis*, ASCE J. Engng. Mech. Div., 98 (1972), pp. 1471–1485.

[BaWi73] ———, *Solution methods for eigenvalue problems in structural mechanics*, Int. J. Num. Meth. Engng., 6, pp. 213–226.

[Bouk72] W. J. Bouknight, *et al.*, *The* Illiac IV *system*, Proc. IEEE, 60 (1972), pp. 369–388.

[ChGG78] M. S. Chance, J. M. Greene, R. C. Grimm, et al., *Comparative numerical studies of ideal magnetohydrodynamic instabilities*, J. Comp. Physics, 28 (1978), pp. 1–13.

[CrBa68] M. M. Craig and M. C. C. Bampton, *Coupling of substructures for dynamic analysis*, Amer. Inst. Aeronautics and Astronautics, 6 (1968), pp. 1313–1319.

[FaFa63] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*, W. H. Freeman, San Francisco, 1963.

[Fran68] J. N. Franklin, *Matrix Theory*, Prentice Hall, Englewood Cliffs, NJ, 1968.

[Gera71] M. Geradin, *The computational efficiency of a new minimization algorithm for eigenvalue analysis*, J. Sound and Vibration, 19 (1971), pp. 319–331.

[GoUn77] G. H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, pp. 361–377.

[Grca81] J. Grcar, *Analysis of the Lanczos algorithm and of the approximation problem in Richardson's method*, Ph.D. Thesis, University of Illinois, Urbana-Champaign, 1981.

[GrGJ76] R. C. Grimm, J. M. Greene and J. L. Johnson, *Computation of the magnetohydrodynamic spectrum in axisymmetric toroidal confinement systems*, in Methods of Computational Physics, 16, Academic Press, New York, 1978.

[Grub78] R. Gruber, *Finite hybrid elements to compute the ideal magnetohydrodynamic spectrum of an axisymmetric plasma*, J. Comp. Physics, 26 (1978), pp. 379–389.

[HeKa52] M. R. Hestenes and W. Karush, *Solutions of $Ax = \lambda Bx$*, J. Res. Nat. Bur. Stds., 49 (1952), pp. 409–436.

[Hous64] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Dover, New York, 1975.

[Jens72] P. S. Jensen, *The solution of large symmetric eigenproblems by sectioning*, this Journal, 9 (1972), pp. 534–545.

[Kaha66] W. Kahan, *Relaxation methods for an eigenproblem*, Dept. Comp. Sci. Tech. Rept. CS-44, Stanford Univ., Stanford, CA, 1966.

[LaHa74] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[LoMc80] D. E. Longsine and S. F. McCormick, *Simultaneous Rayleigh quotient minimization methods for $Ax = \lambda Bx$*, Lin. Alg. Appl., 34 (1980), pp. 195–234.

[Luen73] D. G. LUENBERGER, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

[MoSt73] C. B. MOLER AND G. W. STEWART, *An algorithm for generalized matrix eigenvalue problems*, this Journal, 10 (1973), pp. 241–256.

[Ostr60] A. M. OSTROWSKI, *A quantitative formulation of Sylvester's law of inertia* (*I and II*), Proc. National Academy of Sciences, 45 (1959), pp. 740–744, and 46 (1960), pp. 859–862.

[PaRe80] B. N. PARLETT AND J. K. REID, *Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems*, this Journal, 18 (1981), pp. 135–155.

[Parl80] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[Parl80a] ———, *A new look at the Lanczos algorithm for solving symmetric systems of linear equations*, Lin. Alg. Appl., 29 (1980), pp. 323–346.

[PaSa75] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, this Journal, 12 (1975), pp. 617–629.

[PaSc79] B. N. PARLETT AND D. S. SCOTT, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 311–328.

[Reid71] J. K. REID, *On the method of conjugate gradients for the solution of large sparse systems of linear equations*, Large Sparse Sets of Linear Equations, J. K. Reid, ed., Academic Press, London, 1971, pp. 231–254.

[Rice66] J. R. RICE, *Experiments on Gram–Schmidt orthogonalization*, Math. Comp., 20 (1966), pp. 325–328.

[Ruhe74] A. RUHE, *SOR methods for the eigenvalue problem with large sparse matrices*, Math. Comp., 28 (1974), pp. 695–710.

[Ruti69] H. RUTISHAUSER, *Computational aspects of F. L. Bauer's simultaneous iteration method*, Numer. Math., 13 (1969), pp. 4–13.

[Ruti70] ———, *Simultaneous iteration method for symmetric matrices*, Numer. Math., 16 (1970), pp. 205–223.

[SaSa80] Y. SAAD AND A. SAMEH, *Block conjugate gradient methods and parallel computation*, in Elliptic Problem Solvers, Academic Press, New York, 1980, pp. 405–411.

[Same71] A. H. SAMEH, *On Jacobi and Jacobi-like algorithms for a parallel computer*, Math. Comp., 25 (1971), pp. 579–590.

[Schw74] H. R. SCHWARZ, *The method of coordinate overrelaxation for* $(A-\lambda B)x=0$, Numer. Math., 23 (1974), pp. 135–151.

[Stew76] G. W. STEWART, *A bibliographical tour of the large, sparse generalized eigenvalue problem*, in Sparse Matrix Computations, Academic Press, New York, 1976, pp. 113–130.

[Wilk65] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[Wilk72] ———, *Inverse iteration in theory and in practice*, in Symposia Mathematica X, Academic Press, London, 1972, pp. 361–379.

[Wisn81] J. A. WISNIEWSKI, *On solving the large sparse generalized eigenvalue problem*, Ph.D. thesis; also, R-81-1056, Dept. Computer Science, Univ. of Illinois, Urbana-Champaign, 1981.