JOURNAL OF COMPLEXITY 2, 131–162 (1986)

On the Optimal Solution of Large Eigenpair Problems*

JACEK KUCZYŃSKI†

Department of Computer Science, Columbia University, New York, New York 10027

Received August 15, 1985

The problem of approximation of an eigenpair of a large $n \times n$ matrix A is considered. We study algorithms which approximate an eigenpair of A using the partial information on A given by $b, Ab, \ldots, A^j b, j \ll n$, i.e., by Krylov subspaces. A new algorithm called the generalized minimal residual (gmr) algorithm is analyzed. Its optimality for some classes of matrices is proved. We compare the gmr algorithm with the widely used Lanczos algorithm for symmetric matrices. The gmr and Lanczos algorithms cost essentially the same per step and they have the same stability characteristics. Since the gmr algorithm never requires more steps than the Lanczos algorithm, and sometimes uses substantially fewer steps, the gmr algorithm seems preferable. We indicate how to modify the gmr algorithm in order to approximate p eigenpairs of A. We also show some other problems which can be nearly optimally solved by gmr-type algorithms. The gmr algorithm for symmetric matrices was implemented and some numerical results are described. The detailed implementation, more numerical results, and the Fortran subroutine can be found in Kuczyński ("Implementation of the gmr Algorithm for Large Symmetric Eigenproblems," Report, Columbia University, 1985). The Fortran subroutine is also available via anonymous FTP as "pub/gmrval" on COLUMBIA·EDU [128.59.16.1] on the Arpanet.

1. INTRODUCTION

Suppose we wish to find an approximation to an eigenpair of a very large matrix A. That is, we wish to compute (x, ρ), where x is an $n \times 1$ normalized vector, $\|x\| = 1$, ρ is a complex number such that

$$\|Ax - \rho x\| < \epsilon \tag{1.1}$$

*This research was supported in part by the National Science Foundation under Grant DCR-82-14322.

†Permanent address: Institute of Computer Science, Polish Academy of Sciences, P. O. Box 22, 00-901 Warsaw, PKiN, X floor, Poland.

0885-064X/86 $3.00

for a given positive ϵ. Here $\|\cdot\|$ denotes the 2-norm, $\|x\| = \|x\|_2$. Note that if (x, ρ) satisfies (1.1) and $\rho = (Ax, x) = x^H Ax$ then there exists a matrix E, $\|E\| < \epsilon$, such that $(A - E)x = \rho x$; i.e., the pair (x, ρ) is the exact eigenpair of $A - E$. For instance, we may take $E = xr^H + rx^H$, $\|E\| = \|r\|$, where $r = Ax - \rho x$. A pair (x, ρ) satisfying (1.1) is called a *residual ϵ-approximation*.

The usual procedure for large sparse matrices is to approximate eigenpairs of A from the behavior of A in a given subspace of small dimension. The most commonly used method for approximating eigenpairs of symmetric matrices is the *Lanczos* algorithm, which is the Rayleigh–Ritz algorithm using Krylov subspace. It may be described as follows. At the jth step of this algorithm we know *information*

$$N_j(A, b) = [b, Ab, \ldots, A^j b]$$

for a real nonzero vector b. For sparse matrices A the cost of computing $N_j(A, b)$ is proportional to nj. This information $N_j(A, b)$ is equivalent to the knowledge of the jth Krylov subspace $A_j = \operatorname{span}(b, Ab, \ldots, A^{j-1}b)$ and the vector $A^j b$. If vectors $b, \ldots, A^j b$ are linearly independent then we construct an $n \times (j + 1)$ matrix $Q_{j+1} = (q_1, q_2, \ldots, q_{j+1})$, where $q_1, q_2, \ldots, q_{j+1}$ is an orthonormal basis of the subspace A_{j+1}, the so-called Lanczos basis, such that the $(j + 1) \times j$ matrix D_j,

$$D_j \overset{\text{df}}{=} Q_{j+1}^T A Q_j = \begin{pmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \alpha_2 & & \\ & & \ddots & \beta_{j-1} \\ & & \beta_{j-1} & \alpha_j \\ 0 & & & \beta_j \end{pmatrix} = \begin{pmatrix} H_j \\ \beta_j e_j^T \end{pmatrix}; \qquad e_j = (0, \ldots, 0, 1)^T, \tag{1.2}$$

is tridiagonal. In other words the matrix Q_j partially reduces the matrix A to the tridiagonal form. The Lanczos algorithm disregards the last codiagonal coefficient β_j and deals with the resulting $j \times j$ tridiagonal matrix H_j. In fact, β_j is used, but only to judge the accuracy of the approximations. Pairs $(Q_j g_i, \theta_i)$, $i = 1, 2, \ldots, j$, where (g_i, θ_i) are all eigenpairs of the matrix H_j, serve as approximations of eigenpairs of A. If the codiagonal coefficient β_j is equal to zero then the pairs $(Q_j g_i, \theta_i)$, $i = 1, 2, \ldots, j$, are the exact eigenpairs of A. In general, i.e., for any β_j, we have the following formula on the (smallest) *residual r_j^L of the Lanczos algorithm* (see Parlett, 1980, p. 260),

$$r_j^L = \min_{1 \le i \le j} \|AQ_j g_i - \theta_i Q_j g_i\| = |\beta_j| \min_{1 \le i \le j} |g_i^j| \le |\beta_j|, \tag{1.3}$$

where g_i^j is the jth (the last) component of the vector g_i. This estimate explains why for small β_j the Lanczos algorithm produces small residual error. However, it is not obvious whether the Lanczos algorithm produces the best possible result, especially for "large" β_j.

The main problem addressed in our paper is to find an optimal algorithm which produces a residual ϵ-approximation, i.e., (x, ρ) satisfying (1.1), regardless of the magnitude of β_j. By an optimal algorithm we mean the algorithm which computes (x, ρ) using the minimal j, i.e., the minimal number of matrix–vector multiplications.

We define a new algorithm in Section 2. It is called the *generalized minimal residual algorithm* (the gmr algorithm). In Section 3 we prove that the gmr algorithm almost optimally uses information $N_j(A, b)$. The gmr algorithm is defined for any complex matrix. In the jth step this algorithm constructs the pair (x_j^*, ρ_j^*) such that $\|x_j^*\| = 1$ and the residual $r_j^G = \|Ax_j^* - \rho_j^* x_j^*\|$ is minimal in the jth Krylov subspace, i.e.,

$$r_j^G = \min\{\|Ax - \rho x\| : x \in A_j, \|x\| = 1, \rho \in \mathbb{C}\}.$$

The gmr algorithm has attractive optimality properties. It uses information in an almost optimal way in the following sense. As we mentioned before, the matrix A belongs to a given class F. We assume that class F is *unitarily invariant*; i.e., $A \in F$ implies that $Q^H A Q \in F$ for any unitary matrix Q. Examples of such classes F include the class of all Hermitian matrices or the class of Hermitian matrices with fixed eigenvalues (for some other examples see Traub and Woźniakowski (1984)). We show that if the matrix A belongs to the unitarily invariant class F then the gmr algorithm is *almost strongly optimal* in F. Roughly speaking, this means that the gmr algorithm minimizes the number of matrix–vector multiplications (up to the additive constant not greater than 2) in order to find a residual ϵ-approximation over all possible algorithms that use $N_j(A, b)$. This holds for any matrix A from F, for any nonzero vector b, and for any positive ϵ. The precise meaning of optimality can be found in Section 2. We also prove that for the class of symmetric matrices $F = \{A : A = A^H\}$ the gmr algorithm is *strongly optimal* in F; i.e., it minimizes the number of steps for any matrix A from F, any vector b, and any positive ϵ.

In Section 4 we compare the gmr algorithm for symmetric matrices with the Lanczos algorithm. We prove that the *residual* r_j^G *obtained in the jth step of the gmr algorithm* is given by

$$r_j^G = \min\{\sqrt{\|Ax\|^2 - (Ax, x)^2} : x \in A_j, \|x\| = 1\},$$

while the residual r_j^L obtained in the jth step of the Lanczos algorithm is given by

$$r_j^{\mathrm{L}} = \min\{\sqrt{\|Ax\|^2 - (Ax, x)^2} : x \in A_j, \|x\| = 1,$$
$$(A - (Ax, x)I)x \perp A_j\}.$$

We see that r_j^{G} and r_j^{L} are defined by similar formulas. The difference is only in the set over which the minimum is taken. The set which appears for the Lanczos algorithm is, in general, a proper subset of the set which appears for the gmr algorithm. This may look like a small difference between these two algorithms. We show that this small difference causes completely different results. It is easy to see that $r_j^{\mathrm{G}} \leq r_j^{\mathrm{L}}$. Moreover $r_1^{\mathrm{G}} = r_1^{\mathrm{L}}$ and $r_n^{\mathrm{G}} = r_n^{\mathrm{L}} = 0$. What can happen to the residuals r_j^{G} and r_j^{L} for $j \in [2, n-1]$? We construct an example of the $n \times n$ matrix A, the $n \times 1$ vector b, and $\epsilon > 0$ such that the gmr algorithm computes a residual ϵ-approximation in the second step, while the Lanczos algorithm needs exactly n steps to solve the problem. We also construct a matrix A such that the residual error r_j^{L} of the Lanczos algorithm not only increases but the ratio $r_{j+1}^{\mathrm{L}}/r_j^{\mathrm{L}}$ can be arbitrary large; i.e., for any positive constants $M_1, M_2, \ldots, M_{n-2}$ there exists an $n \times n$ matrix A such that

$$r_{j+1}^{\mathrm{L}}/r_j^{\mathrm{L}} = M_j, \qquad j = 1, 2, \ldots, n-2.$$

This is a serious drawback of the Lanczos algorithm. The gmr algorithm does not have this defect since the sequence of residuals r_j^{G} of the gmr algorithm is nonincreasing for any matrix A.

We next discuss the properties of the gmr algorithm for symmetric matrices. We prove that the gmr algorithm reduces the residuals at least in every second step. More precisely, we show that for any symmetric matrix A if $r_j^{\mathrm{G}} > 0$, then

$$r_{j+2}^{\mathrm{G}} < r_j^{\mathrm{G}}.$$

In Section 5 we analyze the speed of convergence of two algorithms. For the gmr algorithm we prove that for every real symmetric matrix A we have

$$r_j^{\mathrm{G}} \leq \|A\|/j.$$

This estimate is sharp since for every $j < n$ there exists a real symmetric matrix A such that

$$r_j^{\mathrm{G}} \geq \|A\|/2j.$$

For the Lanczos algorithm we easily conclude that for every real symmetric matrix A we have

$$r_j^{\mathrm{L}} \leq \|A\|/\sqrt{j}.$$

This estimate is sharp since for every $j < n$ there exists a real symmetric matrix A such that

$$r_j^{\mathrm{L}} \geq \frac{\|A\|}{\sqrt{j+1}}.$$

In Section 6 we prove that information $N_j(A, b)$ is too weak for finding a residual ϵ-approximation for nonsymmetric matrices. More precisely, we construct a nonsymmetric real matrix A for which the gmr algorithm produces residuals r_j^{G} equal to the first one, i.e.,

$$r_{n-1}^{\mathrm{G}} = r_{n-2}^{\mathrm{G}} = \cdots = r_1^{\mathrm{G}} > 0.$$

Thus there exists a matrix for which in order to find any approximation of an eigenpair with the residual error smaller than r_1^{G} the gmr algorithm must perform exactly n steps. Since the gmr algorithm is strongly optimal in the subclass of all nonsymmetric matrices we conclude that information $N_j(A, b)$ is too weak for the nonsymmetric case.

Section 7 contains the generalization of the gmr algorithm to the problem of finding approximations to p eigenpairs of a symmetric matrix, $p \geq 1$. We prove that also in this case the gmr algorithm (called in this case the p-gmr algorithm) is almost strongly optimal for unitarily invariant classes of matrices. More precisely, the p-gmr algorithm minimizes the number of steps up to a constant not greater than $p + 1$ over all possible algorithms. Here by a residual ϵ-approximation we mean p pairs $(x_1, \rho_1), \ldots, (x_p, \rho_p)$ satisfying

$$\sum_{i=1}^{p} \|Ax_i - \rho_i x_i\|^2 < \epsilon^2 \qquad \text{or} \qquad \max_{1 \leq i \leq p} \|Ax_i - \rho_i x_i\| < \epsilon.$$

We also study some related problems in infinite-dimensional Hilbert spaces which may be almost optimally solved by the gmr-type algorithms.

This paper deals mainly with theoretical properties of the gmr algorithm.

In the last section we report a few numerical tests of the gmr algorithm for the symmetric eigenproblem. A sketch of the implementation of the gmr algorithm for this problem is described in Kuczyński (1983). The Fortran subroutine of the gmr algorithm and extensive numerical tests may be found in Kuczyński (1985).

We end this introduction with a comment on Krylov information. In a recent paper, Chou (1985) studies more general information

$$\overline{N}_j(A, b) = [b, Az_1, \ldots, Az_j]$$

where $b \neq 0$ and z_i depends on previously computed information, i.e., $z_i = z_i(b, Az_1, \ldots, Az_{i-1})$ for $i = 1, 2, \ldots, j$. Chou asks how to find z_i

in order to minimize matrix–vector multiplications which are required for an ϵ-approximation. On the basis of the result of Nemirovsky and Yudin (1983, p. 262), he proves that there exists no choice of z_i for which one can find an ϵ-approximation performing less than half of the steps needed by the gmr algorithm using the Krylov information. Thus the Krylov information is almost optimal.

2. Basic Definitions

Let F be a class of $n \times n$ matrices. For a given $\epsilon > 0$ and any matrix A from F we want to find a vector $x \in \mathbb{C}^n$ (or $\mathbb{R}^n$), $\|x\| = 1$, and a number $\rho \in \mathbb{C}$ (or $\mathbb{R}$) satisfying (1.1), i.e.,

$$\|Ax - \rho x\| < \epsilon.$$

Here

$$\|x\| = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2} \qquad \text{for } x = (x_1, x_2, \ldots, x_n)^T.$$

Adapting terminology and notation from Traub and Woźniakowski (1980, 1984) we formalize the concept of partial information as follows:

Let S_n be a unit sphere in $\mathbb{C}^n$ (or $\mathbb{R}^n$) and let a vector b belong to S_n. Then we define information $N_j(A, b)$ as

$$N_j(A, b) = [b, Ab, \ldots, A^j b], \qquad \forall A \in F, \forall b \in S_n, j = 0, 1, 2, \ldots.$$

By an algorithm we mean a sequence $\Phi = \{\Phi_j\}_{j=0}^{\infty}$ of any mappings

$$\Phi_j: N_j(F, S_n) \to S_n \times \mathbb{C} \text{ (or } S_n \times \mathbb{R}), \qquad (x_j, \rho_j) = \Phi_j(N_j(A, b)).$$

Let $V(N_j(A, b))$ be the set of all matrices which have the same information as the matrix A, i.e.,

$$V(N_j(A, b)) = \{\tilde{A} : \tilde{A} \in F, N_j(\tilde{A}, b) = N_j(A, b)\}.$$

Define the index of the algorithm Φ as

$$k(\Phi, A, b) = \min\{j : \|\tilde{A}x_j - \rho_j x_j\| < \epsilon, \forall \tilde{A} \in V(N_j(A, b)),$$
$$(x_j, \rho_j) = \Phi_j(N_j(A, b))\}.$$

If this set is empty then $k(\Phi, A, b) = +\infty$. Of course, $k(\Phi, A, b)$ depends also on ϵ, n, and the subclass F. Since ϵ, n, and F are fixed, this is not listed

as the arguments of the index. The index $k(\Phi, A, b)$ shows how many steps one has to perform to find an ϵ-approximation of an eigenpair by the algorithm $\Phi = \{\Phi_j\}$ for all matrices which share the same information.

DEFINITION 2.1. The algorithm Φ^* is strongly optimal in F iff

$$k(\Phi^*, A, b) = \min_{\Phi} k(\Phi, A, b), \qquad \forall (A, b) \in F \times S_n,$$

and is almost strongly optimal in F iff there exists a constant c of order unity such that

$$k(\Phi^*, A, b) \leq \min_{\Phi} k(\Phi, A, b) + c, \qquad \forall (A, b) \in F \times S_n.$$

In other words a strongly optimal algorithm performs the smallest number of steps to calculate an ϵ-approximation of the eigenpair for each matrix from the class F. An almost strongly optimal algorithm will perform only a few steps more than a strongly optimal one.

We now define the generalized minimal residual algorithm (gmr). The optimality of this algorithm will be proved in Section 3.

DEFINITION 2.2. For $j = 0$ we know $N_0(A, b) = b$. Set

$$\Phi_0^{\text{gmr}}(N_0(A, b)) = (x_0^*, \rho_0^*) = (b, 0).$$

For $j > 1$ we know $N_j(A, b) = [b, Ab, \ldots, A^j b]$. Let

$$E_j = \{(x, \rho) : x \in A_j, \|x\| = 1, \rho \in \mathbb{C} \text{ (or } \mathbb{R})\},$$

where $A_j = \text{span}(b, Ab, \ldots, A^{j-1}b)$. Define $j + 1$ numbers $c_0^*, c_1^*, \ldots, c_{j-1}^* \in \mathbb{C}$ (or $\mathbb{R}$) and $\rho^* \in \mathbb{C}$ (or $\mathbb{R}$) by

$$\|(A - \rho^* I)(c_0^* b + c_1^* Ab + \cdots + c_{j-1}^* A^{j-1} b)\| = \min_{(x, \rho) \in E_j} \|(A - \rho I)x\|,$$

where I is the $n \times n$ identity matrix. Note that c_i^* and ρ^* depend only on $b, Ab, \ldots, A^j b$, i.e., on $N_j(A, b)$. The jth step of the generalized minimal residual algorithm is given by

$$\Phi_j^{\text{gmr}}(N_j(A, b)) = (x_j^*, \rho_j^*) = (c_0^* b + c_1^* Ab + \cdots + c_{j-1}^* A^{j-1} b, \rho^*).$$

Obviously, the index $k(\Phi^{\text{gmr}}, A, b) \leq n$ for any A, b.

The definition of the gmr algorithm is a simple generalization of a well-known minimal residual (mr) algorithm for solving linear systems $Ax = b$. Knowing information $N_j(A, b)$ the mr algorithm finds an x_j, $x_j \in A_j$, such that $\|Ax_j - b\| = \min_{x \in A_j} \|Ax - b\|$; see, e.g., Traub and Woźniakowski (1984).

3. Optimality of the Generalized Minimal Residual Algorithm

We now prove almost strong optimality of the gmr algorithm for classes of matrices which are unitarily invariant. We recall this concept from Traub and Woźniakowski (1984).

Definition 3.1. The class F is said to be unitarily (orthogonally) invariant iff

$$A \in F \Rightarrow Q^{\mathrm{H}}AQ \in F, \qquad \forall Q \text{ unitary (orthogonal)}.$$

We are ready to formulate the main theorem of this paper.

Theorem 3.1. *If F is unitarily (orthogonally) invariant then the gmr algorithm is almost strongly optimal in F, i.e.,*

$$k(\Phi^{\mathrm{gmr}}, A, b) = \min_{\Phi} k(\Phi, A, b) + a, \qquad \forall (A, b) \in F \times S_n,$$

where $a \in \{0, 1, 2\}$.

Proof. For simplicity we present the proof only for the complex case. The proof for the real case can be found in Kuczyński (1983).

Let $\Phi = \{\Phi_j\}$ be an arbitrary algorithm with a finite index $k = k(\Phi, A, b) < +\infty$. This means that $\|\tilde{A}x_k - \rho_k x_k\| < \epsilon$, $\forall \tilde{A} \in V(N_k(A, b))$, where $(x_j, \rho_j) = \Phi_j(N_j(A, b))$. Recall that $A_j = \operatorname{span}(b, Ab, \ldots, A^{j-1}b)$ and $E_j = \{(x, \rho) : x \in A_j, \|x\| = 1, \rho \in \mathbb{C}\}$. Obviously $A_{j+1} \subseteq A_{j+2}$, $\forall j$. Let us consider two cases.

Case I. $A_{k+1} = A_{k+2}$, i.e., $E_{k+1} = E_{k+2}$.

In this case A_{k+1} is an invariant subspace of the matrix A and since the field $\mathbb{C}$ is algebraically closed A_{k+1} contains at least one eigenvector of A. Hence the gmr algorithm using information $N_{k+1}(A, b)$ produces an exact eigenpair $(x_{k+1}^*, \rho_{k+1}^*)$ of the matrix A,

$$0 = \|Ax_{k+1}^* - \rho_{k+1}^* x_{k+1}^*\| \leq \|Ax_k - \rho_k x_k\|.$$

Case II. $A_{k+2} \neq A_{k+2}$, i.e., $E_{k+1} \neq E_{k+2}$.

Let $x_k = z_1 + z_2$, where $z_1 \in A_{k+1}$ and $z_2 \in A_{k+1}^{\perp}$. There exists a vector ξ such that $\xi \in A_{k+2}$ and $\xi \in A_{k+1}^{\perp}$ and $\|\xi\| = 1$. We now prove that there exist a complex number c, $|c| = 1$, and a unitary matrix Q such that

$$Qv = v, \ \forall v \in A_{k+1}, \qquad \text{and} \qquad Qz_2 = c\|z_2\|\xi.$$

Indeed, if vectors z_2 and ξ are linearly dependent then $z_2 = c_1\|z_2\|\xi$, where $|c_1| = 1$ and it is easy to verify that the number $c = -c_1$ and the matrix $Q = I - 2\xi\xi^{\mathrm{H}}$ satisfy the conditions as claimed.

Let z_2 and ξ be linearly independent. Let

$$c = \begin{cases} \dfrac{(z_2, \xi)}{|(z_2, \xi)|} & \text{if } (z_2, \xi) \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

Since $\|z_2\| > |c(z_2, \xi)|$ then the number

$$|\alpha|^2 = \frac{1}{2[(z_2, z_2) - \bar{c}\|z_2\|(z_2, \xi)]}$$

is well defined. It is easy to check that the matrix

$$Q = I - 2ww^{\mathrm{H}},$$

where $w = \alpha(z_2 - c\|z_2\|\xi)$, $\|w\| = 1$, is unitary and

$$Qv = v, \forall v \in A_{k+1}, \qquad \text{and} \qquad Qz_2 = c\|z_2\|\xi$$

as claimed.

Let $\tilde{A} = Q^{\mathrm{H}}AQ$. Since $Qv = v = Q^{\mathrm{H}}v$, $\forall v \in A_{k+1}$, then $\tilde{A}^i b = A^i b$ for $i = 0, 1, \ldots, k$. Due to unitary invariance of the class F we conclude that $\tilde{A} \in F$ and $\tilde{A} \in V(N_k(A, b))$. We have

$$\begin{aligned} \|\tilde{A}x_k - \rho_k x_k\| &= \|Q^{\mathrm{H}}AQx_k - \rho_k x_k\| = \|AQ(z_1 + z_2) - \rho_k Q(z_1 + z_2)\| \\ &= \|A(z_1 + Qz_2) - \rho_k(z_1 + Qz_2)\| = \|Az - \rho_k z\|, \end{aligned}$$

where $z = z_1 + Qz_2$. Note that $z \in A_{k+2}$ and $\|z\| = 1$. Since the gmr algorithm at the $(k + 2)$nd step produces $(x^*_{k+2}, \rho^*_{k+2})$, which is the best approximation in the set E_{k+2}, we have

$$\|\tilde{A}x_k - \rho_k x_k\| = \|Az - \rho_k z\| \geq \|Ax^*_{k+2} - \rho^*_{k+2}x^*_{k+2}\|.$$

This proves that the residual of the algorithm Φ at the kth step is no smaller than the residual of the gmr algorithm at the $(k + 2)$nd step. Since this holds for any algorithm Φ we conclude

$$k(\Phi, A, b) + 2 \geq k(\Phi^{\mathrm{gmr}}, A, b), \qquad \forall \Phi, \forall (A, b) \in F \times S_n.$$

On the other hand

$$\min_{\Phi} k(\Phi, A, b) \leq k(\Phi^{\mathrm{gmr}}, A, b), \qquad \forall (A, b) \in F \times S_n.$$

Hence

$$k(\Phi^{\mathrm{gmr}}, A, b) = \min_{\Phi} k(\Phi, A, b) + a,$$

where $a = 0, 1, 2$. ■

Theorem 3.1 states a very strong optimality property of the gmr algorithm. Neglecting the term a, we see that the gmr algorithm minimizes the number of steps for every matrix A from the class F. Note that this result holds for any unitarily invariant class F, for any ϵ, for any size n of the matrix A, and for any normalized vector b.

Usually $k(\Phi^{\mathrm{gmr}}, A, b)$ is large, especially for large n and small ϵ. Therefore the presence of the term a in Theorem 3.1 is not a limitation in practice. Nevertheless it may be shown that for some unitarily invariant classes the term a may be equal to zero or to one. Thus, Theorem 3.1 is best possible for $a = 0, 1$. We do not know whether $a = 2$ is necessary in Theorem 3.1. We begin with a class $F = F_1$ for which $a = 0$. Let

$$F_1 = \{A : A = A^{\mathrm{H}}\}.$$

Using the notation of the proof of Theorem 3.1 we formulate and prove the following lemma.

LEMMA 3.1. *Let j be any positive integer less than n and let M be any real number. Let Φ be any algorithm, $(x_j, \rho_j) = \Phi_j(N_j(A, b))$. If there exists a matrix A from F_1 and a vector b from S_n such that $x_j \not\in A_j$ then there exists a matrix $\tilde{A} \in V(N_j(A, b))$ for which $\|\tilde{A}x_j - \rho_j x_j\| > M$.*

Proof. For every real η we construct a matrix A_η such that

$$\text{(i) } A_\eta \in V(N_j(A, b)),$$
$$\text{(ii) } \|(A_\eta - \rho_j I)x_j\| \xrightarrow[\eta\to\infty]{} \infty.$$

Let $x_j = y_1 + y_2$, where $y_1 \in A_j$ and $0 \neq y_2 \in A_j^{\perp}$. Let us define

$$A_\eta = A + \eta y_2 y_2^{\mathrm{H}}.$$

Since $A_\eta = A_\eta^{\mathrm{H}}$ then $A_\eta \in F_1$. It is easy to show inductively that $A_\eta^i b = A^i b$, $i = 0, 1, \ldots, j$, for every η. Thus $A_\eta \in V(N_j(A, b))$. We have

$$\|(A_\eta - \rho_j I)x_j\|^2 = \|(A + \eta y_2 y_2^{\mathrm{H}} - \rho_j I)x_j\|^2 = \|(A - \rho_j I)x_j + \eta y_2 y_2^{\mathrm{H}} x_j\|^2$$
$$= \|(A - \rho_j I)x_j\|^2 + 2\eta\|y_2\|^2((A - \rho_j I)x_j, y_2) + \eta^2\|y_2\|^6 \xrightarrow[\eta\to\infty]{} \infty.$$

Taking sufficiently large η we obtain

$$\|(A_\eta - \rho_j I)x_j\| > M.$$

Hence Lemma 3.1 is proved. ■

Lemma 3.1 states that if an algorithm Φ at the jth step produces a vector x_j which does not belong to the subspace A_j then the residuals $\|\tilde{A}x_j - \rho_j x_j\|$ for $\tilde{A} \in V(N_j(A, b))$ are unbounded.

We now prove the following theorem.

THEOREM 3.2. *The gmr algorithm is strongly optimal in the class F_1, i.e.,*

$$k(\Phi^{\text{gmr}}, A, b) = \min_{\Phi} k(\Phi, A, b), \qquad \forall (A, b) \in F_1 \times S_n.$$

Proof. Let $\Phi = \{\Phi_j\}$ be any algorithm. If $x_j \in A_j$ then from the definition of the gmr algorithm it follows that

$$\|(A - \rho_j I)x_j\| = \|(\tilde{A} - \rho_j I)x_j\| \geq \|(\tilde{A} - \rho_j^* I)x_j^*\| = \|(A - \rho_j^* I)x_j^*\|, \quad \forall \tilde{A} \in V(N_j(A, b)),$$

and Theorem 3.2 follows immediately. If $x_j \notin A_j$ then Theorem 3.2 follows from Lemma 3.1. ■

Thus in the class F_1, $a = 0$ in Theorem 3.1.

We note that Theorem 3.2 remains true when we replace the class F_1 by the class $\{A : A = A^{\text{H}} > 0\}$ or by $\{A : A = A^{\text{H}} < 0\}$. Then in the proof of Lemma 3.1 we choose positive η or negative η, respectively.

An example of the class F for which $a = 1$ in Theorem 3.1 is presented in Kuczyński (1983).

An example which shows that the assumption of unitary invariance of the class F in Theorem 3.1 is essential can also be found in Kuczyński (1983).

We illustrate Theorem 3.1 by rather a surprising example.

Let F_2 be the class defined as

$$F_2 = \left\{ A : A = Q^{\text{H}} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} Q,\ Q \text{ unitary} \right\}$$

where λ_i, $i = 1, \ldots, n$, are arbitrary but fixed numbers from $\mathbb{C}$. Observe that all matrices from F_2 share the same eigenvalues. Since F_2 is unitarily invariant, Theorem 3.1 states that the gmr algorithm is almost strongly optimal in F_2. Thus even if we know all eigenvalues of the matrix the gmr algorithm is still almost best possible. This shows that knowledge of all eigenvalues makes the problem of approximating an eigenpair no easier.

4. Comparison with the Lanczos Algorithm

In this section we compare the gmr algorithm with the Lanczos algorithm (L algorithm). We show that the L algorithm can increase the residual error arbitrarily for $(n-1)$ steps, while the residuals obtained from the gmr algorithm are always nonincreasing. We restrict ourselves in this section to the real symmetric case.

Let us briefly describe the jth step of the L algorithm. For a detailed analysis of the L algorithm see Parlett (1980). Knowing information $N_j(A, b)$, where A is a real symmetric matrix and b is a real vector with $\|b\| = 1$, perform the following steps.

1. Find an orthonormal basis $q_1, q_2, \ldots, q_j$ of the subspace A_j; let $Q_j = (q_1, q_2, \ldots, q_j)$ be the $n \times j$ matrix.
2. Form the $j \times j$ matrix $H_j = Q_j^T A Q_j$; compute eigenpairs of H_j;

$$H_j g_j = \theta_i g_i, \qquad (g_i, g_m) = \delta_{i,m}, \; i, m = 1, 2, \ldots, j.$$

3. Compute the Ritz vectors $z_i = Q_j g_i$ and the residual

$$r_j^L = \min_{1 \le i \le j} \|Az_i - \theta_i z_i\|.$$

4. Define

$$Z_j = \{(z_i, \theta_i), i = 1, 2, \ldots, j : \|Az_i - \theta_i z_i\| = r_j^L\};$$

the jth step of the L algorithm is defined by

$$\Phi_j^L(N_j(A, b)) = (x_j, \rho_j),$$

where (x_j, ρ_j) is an arbitrary element from the set Z_j.

We compare r_j^L with the residual r_j^G produced by the jth step of the gmr algorithm.

Let

$$r_j^G = \min_{(x,\rho) \in E_j} \|Ax - \rho x\| \tag{4.1}$$

and

$$\tilde{r}_j^G = \min_{(x,\rho) \in \tilde{E}_j} \|Ax - \rho x\|,$$

where $\tilde{E}_j = \{(x, \rho) : x \in \tilde{A}_j, \; \|x\| = 1, \; \rho \in \mathbb{R}\}$, and $\tilde{A}_j = \text{span}\,(b, Ab, \ldots, A^{j-1}b)$ is a subspace over the real field. The following lemma holds.

LEMMA 4.1. *If A is a real symmetric matrix and b is a real normalized vector then the minimum in (4.1) is attained for a real vector x for $\rho = (Ax, x)$ and*

$$r_j^G = \tilde{r}_j^G = \min\{[\|Ax\|^2 - (Ax, x)^2]^{1/2} : x \in \tilde{A}_j, \|x\| = 1\}.$$

For the proof see Kuczyński (1983).

A similar relation holds for the L algorithm. Since the residual vector $Ax_j - \rho_j x_j$ of the L algorithm is orthogonal to the subspace $\tilde{A}_j$, we can easily find that

$$r_j^L = \min\{[\|Ax\|^2 - (Ax, x)^2]^{1/2} : x \in \tilde{A}_j, \|x\| = 1, \\ (A - (Ax, x)I)x \perp \tilde{A}_j\}.$$

Of course $r_j^G \leq r_{j-1}^G$ and $r_j^G \leq r_j^L$. It is known (see Parlett, 1980) that $r_1^G = r_1^L$. We see that r_j^G and r_j^L are defined by similar formulas. The difference is only in the set over which the minimum is taken. The set which appears for the L algorithm is, in general, a proper subset of the set which appears for the gmr algorithm. This may look like a small difference between these two algorithms. The following theorem shows that this small difference causes completely different results.

THEOREM 4.1. *For any sequence $M_1, M_2, \ldots, M_{n-2}$ of positive numbers there exist a real symmetric matrix A and a real vector b such that*

$$r_{j+1}^L / r_j^L = M_j, \qquad j = 1, 2, \ldots, n-2.$$

Proof. Let

$$A = \begin{pmatrix} 0 & \beta_1 & & & \\ \beta_1 & 0 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & 0 & \beta_{n-1} \\ & & & \beta_{n-1} & 0 \end{pmatrix}, \tag{4.2}$$

$\beta_i > 0$ and $b = (1, 0, \ldots, 0)^T$. Take an aribtrary positive β_1. Then

$$r_1^L = \beta_1.$$

Consider now the second step of the L algorithm. Since $\tilde{A}_2 = \text{span}\,(b, Ab)$ then it is not difficult to calculate that this algorithm yields the two eigenpairs

$$(z_1, \theta_1) = \left(\frac{1}{\sqrt{2}}(1, 1, 0, \ldots, 0)^T, \beta_1\right)$$

and

$$(z_2, \theta_2) = \left(\frac{1}{\sqrt{2}}(-1, 1, 0, \ldots, 0)^T, -\beta_1\right).$$

For both these cases $\|Az_i - \theta_i z_i\| = \beta_2/\sqrt{2}$. If we choose β_2 in such a way that $\beta_2 = \sqrt{2}\, M_1 \beta_1$ we obtain

$$r_2^L = M_1 r_1^L.$$

Assume inductively that $\beta_1, \beta_2, \ldots, \beta_{i-1}, \beta_j \neq 0$, are already defined such that

$$r_j^L = M_{j-1} r_{j-1}^L, \qquad \text{for } j = 2, 3, \ldots, i-1.$$

Since $\tilde{A}_i = \text{span}\,(e_1, e_2, \ldots, e_i)$ then

$$H_i = Q_i^T A Q_i = \begin{pmatrix} 0 & \beta_1 & & & \\ \beta_1 & 0 & \beta_2 & & \\ & & & & \beta_{i-1} \\ & & & \beta_{i-1} & 0 \end{pmatrix}.$$

Let $(g_j, \theta_j) = ((g_j^1, \ldots, g_j^i)^T, \theta_j)$, $j = 1, 2, \ldots, i$, be all eigenpairs of H_i, $g_j \in \mathbb{R}^i$, $\|g_j\| = 1$. Since the last component of any eigenvector of H_i is nonzero ($\beta_j \neq 0$), then

$$\kappa_i \stackrel{df}{=} \min_{1 \leq j \leq i} |g_j^i| > 0.$$

Since $r_i^L = \kappa_i \beta_i$ then we choose a number β_i such that $\beta_i = M_{i-1} r_{i-1}^L / \kappa_i$, which gives

$$r_i^L = M_{i-1} r_{i-1}^L.$$

By induction we obtain that

$$r_i^L = M_{i-1} r_{i-1}^L, \qquad \text{for } i = 2, 3, \ldots, n-1. \quad \blacksquare$$

Theorem 4.1 states that it may occur that the residual error r_i^L not only increases but the ratio r_{i+1}^L / r_i^L can be arbitrary large for $i = 1, 2, \ldots, n-2$. On the contrary, the gmr algorithm does not increase the residuals,

$r_i^G \le r_{i-1}^G$, $i = 2, 3, \ldots n$. This is a serious drawback of the Lanczos algorithm. It should be noted that the norm of A from the proof of Theorem 4.1 is large for large M_j. We have performed a number of tests for matrices of norm bounded by one. For such matrices, numerical tests confirmed that in many cases the ratio r_{i+1}^L/r_i^L was larger than one for some i. Sometimes the ratio r_{i+1}^L/r_i^L was very large, up to 150 for random matrices and up to 24,000 for the tridiagonal matrix with zero diagonal and $\beta_i = 1/20$ for $i = 1, 11, 21, \ldots, 91$ and $\beta_i = \frac{1}{2}$ for the remaining i from 2 to 100.

To explain the poor behavior of the L algorithm observe that although the coefficients $\alpha_1, \ldots, \alpha_i, \beta_1, \ldots, \beta_{i-1}, \beta_i$ of (1.2) are known, the L algorithm does not use the number β_i at the ith step. It is worth mentioning that even in the $(i + 1)$st step, when the L algorithm uses all numbers $\beta_1, \ldots, \beta_i$ and $\alpha_1, \ldots, \alpha_i, \alpha_{i+1}$ it may happen that $r_{i+1}^L \gg r_i^L$. This proves that the L algorithm does not exploit information in an optimal way.

From Theorem 4.1 one can easily construct an example of A such that the gmr algorithm finds an ϵ-approximation at the second step and the L algorithm needs exactly n steps. Indeed, choose a number β_2 such that $\sqrt{2}\,\beta_1 < \beta_2 < 2\beta_1$ in (4.2). Then from the proof of Theorem 4.1 we see that

$$r_1^L < r_2^L.$$

Define $\beta_3, \ldots, \beta_{n-1}$ in (4.2) in such a way that

$$r_2^L < r_3^L < \cdots < r_{n-1}^L.$$

Consider now the gmr algorithm for this matrix. Then $r_1^G = \beta_1$. It is easy to calculate that in the second step of the gmr algorithm we get

$$r_2^G = \min_{c_1^2+c_2^2=1} [(-2\beta_1 c_1^2 c_2 + \beta_1 c_2)^2 + (\beta_1 c_1 - 2\beta_1 c_1 c_2^2)^2 + \beta_2^2 c_2^2]^{1/2}$$

$$= \min_{c_1^2+c_2^2=1} [\beta_1^2 + c_2^2(\beta_2^2 - 4c_1^2\beta_1^2)]^{1/2}.$$

Since $\beta_2 < 2\beta_1$ then $r_2^G < \beta_1 = r_1^G$. Taking ϵ from (r_2^G, r_1^G) we get the desired result.

Observe that if we take $\beta_2 > 2\beta_1$ in (4.2) then $r_1^G = r_2^G$. This shows that the residuals produced by the gmr algorithm do not necessarily decrease at every step. The following theorem is shown in Kuczyński (1983).

THEOREM 4.2. *Let A be a real symmetric matrix and b a real unit vector. If $r_i^G \neq 0$ then $r_{i+2}^G < r_i^G$, for $i = 1, 2, \ldots, n - 1$.*

This means that the residuals of the gmr algorithm decrease at least in every second step. We do not know if there exists a matrix for which the residuals are of the form

$$r_1^G = r_2^G > r_3^G = r_4^G > r_5^G = r_6^G > r_7^G = \cdots .$$

Some numerical tests suggest that such a matrix exists; see Section 8.

5. Convergence of the gmr and Lanczos Algorithms for Real Symmetric Matrices

In this section we want to establish how fast the sequences of residuals of the gmr and Lanczos algorithms decrease for the real symmetric case. In the rest of this section we denote $r_i^G = r_i^G(A, b)$ and $r_i^L = r_i^L(A, b)$ to stress that the residuals come from the matrix A and vector b. For the gmr algorithm we have

THEOREM 5.1. *For every real symmetric matrix A and every real unit vector b,*

$$r_j^G(A, b) \leq \frac{1}{j}\|A\|, \qquad \forall j \geq 1.$$

For every j, $j < n$, there exist a real symmetric matrix A and a real unit vector b such that

$$r_j^G(A, b) \geq \frac{1}{2j}\|A\|.$$

Proof. The proof is rather long and complicated. It consists of a series of lemmas. First of all, observe that

$$r_j^G(A, b) = |c| r_j^G(A/c, b), \qquad \forall c \in \mathbb{R}, c \neq 0.$$

Taking $c = \|A\|$ we can assume without loss of generality that $\|A\| = 1$. Let

$$F = \{A : A = A^T, A \text{ real}, \|A\| = 1\}.$$

We prove that

$$f(j) = \sup_{\substack{A \in F \\ b \in S_n}} (r_j^G(A, b))^2 \leq j^{-2}.$$

Since A is symmetric there exist (v_i, λ_i), $i = 1, \ldots, n$, such that

$$Av_i = \lambda_i v_i, \qquad (v_i, v_m) = \delta_{i,m}.$$

Let

$$b = \sum_{i=1}^{n} c_i v_i \qquad \text{and} \qquad \sum_{i=1}^{n} c_i^2 = 1.$$

Then it is easy to see that

$$x \in \tilde{A}_j \qquad \text{iff } x = w(A)b = \sum_{i=1}^{n} c_i w(\lambda_i) v_i,$$

where $w \in W_{j-1}$ and W_{j-1} denotes the class of all real polynomials of degree not greater than $j - 1$. Let

$$\Omega_n = \left\{ \alpha = (\alpha_1, \ldots, \alpha_n) : \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i = 1 \right\}.$$

Define a set $W_{j-1}(\alpha, \lambda)$, where $\lambda = (\lambda_1, \ldots, \lambda_n)$, as follows:

$$W_{j-1}(\alpha, \lambda) = \left\{ w \in W_{j-1} : \sum_{i=1}^{n} \alpha_i w^2(\lambda_i) = 1 \right\}.$$

Hence

$$(r_j^G(A, b))^2 = \min_{(x,\rho)\in\tilde{E}_j} \|Ax - \rho x\|^2 = \min_{\substack{\rho\in\mathbb{R} \\ w\in W_{j-1}(c^2,\lambda)}} \sum_{i=1}^{n} c_i^2(\lambda_i - \rho)^2 w^2(\lambda_i),$$

where $c^2 = (c_1^2, \ldots, c_n^2)$ and

$$f(j) = \sup_{\substack{\lambda_i\in[-1,1] \\ c^2\in\Omega_n}} \min_{\substack{\rho\in\mathbb{R} \\ w\in W_{j-1}(c^2,\lambda)}} \sum_{i=1}^{n} c_i^2(\lambda_i - \rho)^2 w^2(\lambda_i).$$

Without loss of generality we may assume that all c_i are not equal to zero and all λ_i are distinct. The eigenvalues λ_i and coefficients c_i generate an inner product in the subspace W_{n-1}:

$$\langle h(t), g(t) \rangle = \sum_{i=1}^{n} c_i^2 h(\lambda_i) g(\lambda_i).$$

Consider two linear functionals

$$I(h) = \sum_{i=1}^{n} c_i^2 h(\lambda_i) \qquad \text{and} \qquad \hat{I}(h) = \sum_{i=1}^{j+1} \alpha_i h(y_i),$$

where y_i, $i = 1, 2, \ldots, j+1$, are all zeros of the orthogonal polynomial P_{j+1} of degree $j+1$ in the inner product $\langle \cdot, \cdot \rangle$ and α_i are the corresponding Christoffel numbers

$$\alpha_i = \sum_{m=1}^{n} c_m^2 \left(\frac{P_{j+1}(\lambda_m)}{P'_{j+1}(y_i)(\lambda_m - y_i)} \right)^2 \quad \left(\text{or } \alpha_i^{-1} = \sum_{m=0}^{j} P_m^2(y_i)\right),$$
$$i = 1, 2, \ldots, j+1.$$

It may be proved (see Szegö, 1939, p. 47) that

$$I(h) = \hat{I}(h), \qquad \forall h \in W_{2j+1}.$$

Let $\alpha = (\alpha_1, \ldots, \alpha_{j+1})$ and $y = (y_1, \ldots, y_{j+1})$. Then we have

$$\begin{aligned} f(j) &= \sup_{\substack{\lambda_i \in [-1,1] \\ c^2 \in \Omega_n}} \min_{\substack{\rho \in \mathbb{R} \\ w \in W_{j-1}(c^2, \lambda)}} \sum_{i=1}^{n} c_i^2 (\lambda_i - \rho)^2 w^2(\lambda_i) \\ &\leq \sup_{\substack{y_i \in [-1,1] \\ \alpha \in \Omega_{j+1}}} \min_{\substack{\rho \in \mathbb{R} \\ w \in W_{j-1}(\alpha, y)}} \sum_{i=1}^{j+1} \alpha_i (y_i - \rho)^2 w^2(y_i), \end{aligned}$$

where $y_i = y_i(\lambda_1, \ldots, \lambda_n, c_1, \ldots, c_n)$, $\alpha_i = \alpha_i(\lambda_1, \ldots, \lambda_n, c_1, \ldots, c_n) > 0$, for $i = 1, 2, \ldots, j+1$. On the other hand, having α_i and y_i for $i = 1, 2, \ldots, j+1$, since $n \geq j+1$, we may define $c_i = \sqrt{\alpha_i}$, $\lambda_i = y_i$ for $i = 1, 2, \ldots, j+1$ and $c_i = \lambda_i = 0$ for $i = j+2, \ldots, n$. Thus we get the opposite inequality. Thus

$$f(j) = \sup_{\substack{y_i \in [-1,1] \\ \alpha \in \Omega_{j+1}}} \min_{\substack{\rho \in \mathbb{R} \\ w \in W_{j-1}(\alpha, y)}} \sum_{i=1}^{j+1} \alpha_i (y_i - \rho)^2 w^2(y_i).$$

Define the function $g(j, \rho, \alpha, y)$

$$g(j, \rho, \alpha, y) = \min_{w \in W_{j-1}(\alpha, y)} \sum_{i=1}^{j+1} \alpha_i (y_i - \rho)^2 w^2(y_i).$$

It is easy to see that

$$f(j) = \sup_{\substack{y_i \in [-1,1] \\ \alpha \in \Omega_{j+1}}} \min_{\rho \in \mathbb{R}} g(j, \rho, \alpha, y)$$

and

$$(r_j^{\mathrm{G}}(A, b))^2 = \min_{\rho \in \mathbb{R}} g(j, \rho, \alpha, y).$$

We now prove the following lemma concerning the function $g(j, \rho, \alpha, y)$.

LEMMA 5.1. *Let*

$$\delta_j = \min_{\substack{i \neq m \\ 1 \leq i, m \leq j+1}} |y_i - y_m|.$$

Then $\min_{\rho \in \mathbb{R}} g(j, \rho, \alpha, y) \leq \frac{1}{4}\delta_j^2$.

Proof. Let $\delta_j = y_p - y_q$. Define a nonzero polynomial w of degree not greater than $j - 1$ in the following way:

$$w(y_i) = 0, \qquad i = 1, 2, \ldots, j + 1, i \neq p, i \neq q$$

and

$$\alpha_p w^2(y_p) + \alpha_q w^2(y_q) = 1.$$

Since α_p, α_q, $w^2(y_p)$, and $w^2(y_q)$ are positive then such a polynomial exists. For this polynomial we have

$$\begin{aligned} \min_{\rho \in \mathbb{R}} g(j, \rho, \alpha, y) &\leq \min_{\rho \in \mathbb{R}} [\alpha_p (y_p - \rho)^2 w^2(y_p) + \alpha_q (y_q - \rho)^2 w^2(y_2)] \\ &= \alpha_p \alpha_q w^2(y_p) w^2(y_q) \delta_j^2 \leq \tfrac{1}{4}\delta_j^2. \end{aligned}$$

The last inequality follows immediately by taking $\rho = \frac{1}{2}(y_p + y_q)$. ∎

From Lemma 5.1 it follows that

$$f(j) \leq \frac{1}{4} \sup_{y_i \in [-1, 1]} \min_{\substack{i \neq m \\ 1 \leq i, m \leq j+1}} |y_i - y_m|^2 \leq \frac{1}{4}\frac{4}{j^2} = \frac{1}{j^2}.$$

Thus the first part of the proof of Theorem 5.1 is complete. In order to prove the second part we need some other lemmas concerning the function $g(j, \rho, \alpha, y)$.

LEMMA 5.2. *For arbitrary* $\alpha_m > 0$ *and* y_m, $m = 1, 2, \ldots, j + 1$,

$$\tfrac{1}{2} \min_{1 \leq m \leq j+1} g(j, y_m, \alpha, y) \leq \min_{\rho \in \mathbb{R}} g(j, \rho, \alpha, y) \leq \min_{1 \leq m \leq j+1} g(j, y_m, \alpha, y).$$

Proof. It is easy to see that the function $g(j, \rho, \alpha, y)$ is a polynomial of the second degree with respect to ρ. So it reaches its minimum when

$$\rho_{\min} = \sum_{i=1}^{j+1} \alpha_i y_i w^2(y_i).$$

Thus we have

$$\min_{\rho\in\mathbb{R}} g(j,\rho,\alpha,y) = g(j,\rho_{\min},\alpha,y)$$
$$= \min_{w\in W_{j-1}(\alpha,y)}\left[\sum_{i=1}^{j+1}\alpha_i y_i^2 w^2(y_i) - \left(\sum_{i=1}^{j+1}\alpha_i y_i w^2(y_i)\right)^2\right]$$
$$= \min_{w\in W_{j-1}(\alpha,y)}\left[\sum_{i=1}^{j+1}\alpha_i y_i^2 w^2(y_i)\sum_{i=1}^{j+1}\alpha_i w^2(y_i) - \left(\sum_{i=1}^{j+1}\alpha_i y_i w^2(y_i)\right)^2\right].$$

Applying the Lagrange identity

$$\left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right) - \left(\sum_{i=1}^{n} a_i b_i\right)^2 = \sum_{\substack{i,m=1\\ i<m}}^{n} (a_i b_m - a_m b_i)^2$$

to the last expression with $a_i = \sqrt{\alpha_i}\, y_i w(y_i)$ and $b_i = \sqrt{\alpha_i}\, w(y_i)$ we obtain

$$\min_{\rho\in\mathbb{R}} g(j,\rho,\alpha,y) = \frac{1}{2}\min_{w\in W_{j-1}(\alpha,y)}\sum_{i,m=1}^{j+1}\alpha_i\alpha_m(y_i-y_m)^2 w^2(y_i)w^2(y_m)$$
$$= \frac{1}{2}\min_{w\in W_{j-1}(\alpha,y)}\sum_{m=1}^{j+1}\alpha_m w^2(y_m)\left(\sum_{i=1}^{j+1}\alpha_i(y_i-y_m)^2 w^2(y_i)\right)$$
$$\geq \frac{1}{2}\min_{\substack{1\leq m\leq j+1\\ w\in W_{j-1}(\alpha,y)}}\sum_{i=1}^{j+1}\alpha_i(y_i-y_m)^2 w^2(y_i)$$
$$= \frac{1}{2}\min_{1\leq m\leq j+1} g(j,y_m,\alpha,y).$$

The second inequality of Lemma 5.2 is obvious and thus the proof of the lemma is complete. ■

We now need the following lemmas.

LEMMA 5.3 (Paszkowski, 1982). *For arbitrary $\alpha_i > 0$ and any distinct y_i and for any m, $1 \leq m \leq j+1$ the function $g(j, y_m, \alpha, y)$ is equal to the smallest positive zero of the function*

$$B_m(\mu) = \sum_{i=1}^{j+1}\left[\alpha_i[(y_i-y_m)^2-\mu]\prod_{\substack{p=1\\ p\neq i}}^{j+1}(y_p-y_i)^2\right]^{-1}.$$

Proof. Recall that

$$g(j,y_m,\alpha,y) = \min_{w\in W_{j-1}(\alpha,y)}\sum_{i=1}^{j+1}\alpha_i(y_i-y_m)^2 w^2(y_i).$$

Since $w \in W_{j-1}$ its jth divided difference vanishes, $w[y_1, y_2, \ldots, y_{j+1}] = 0$. Thus

$$\sum_{i=1}^{j+1} w(y_i) \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i) = 0 \tag{5.1}$$

$$\sum_{i=1}^{j+1} \alpha_i w^2(y_i) = 1. \tag{5.2}$$

We seek a minimum $\Sigma_{i=1}^{j+1} \alpha_i (y_i - y_m)^2 w^2(y_i)$ under constraints (5.1) and (5.2). Let $w(y_i) = w_i$. The Lagrange function of this problem is

$$L(w_1, w_2, \ldots, w_{j+1}, \lambda, \mu) = \sum_{i=1}^{j+1} \alpha_i (y_i - y_m)^2 w_i^2 + 2\lambda \sum_{i=1}^{j+1} w_i \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i) - \mu \left(\prod_{i=1}^{j+1} \alpha_i w_i^2 - 1 \right).$$

$$\frac{\partial L}{\partial w_i} = 2\alpha_i (y_i - y_m)^2 w_i - 2\lambda \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i) - 2\mu \alpha_i w_i = 0.$$

Hence

$$w_i(\alpha_i (y_i - y_m)^2 - \mu \alpha_i) = \lambda \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i),$$

and

$$w_i = \lambda \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i) \alpha_i [(y_i - y_m)^2 - \mu].$$

From (5.1) it follows that

$$\lambda \sum_{i=1}^{j+1} 1 \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2 \alpha_i [(y_i - y_m)^2 - \mu] = 0.$$

From (5.2) it follows that

$$\lambda^2 \sum_{i=1}^{j+1} 1 \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2 \alpha_i [(y_i - y_m)^2 - \mu]^2 = 1.$$

Hence our minimum is equal to

$$\sum_{i=1}^{j+1} \alpha_i (y_i - y_m)^2 w_i^2$$

$$= \sum_{i=1}^{j+1} \alpha_i (y_i - y_m)^2 \lambda^2 \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2 \alpha_i^2 [(y_i - y_m)^2 - \mu]^2$$

$$= \lambda^2 \sum_{i=1}^{j+1} [(y_i - y_m)^2 - \mu + \mu] \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2 \alpha_i [(y_i - y_m)^2 - \mu]^2$$

$$= \lambda^2 \prod_{i=1}^{j+1} 1 \Big/ \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2 \alpha_i [(y_i - y_m)^2 - \mu]$$

$$= \lambda^2 \sum_{i=1}^{j+1} \mu \Big/ \sum_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2 \alpha_i [(y_i - y_m)^2 - \mu]^2 = \mu.$$

Thus our minimum is equal to the smallest μ which satisfies the equation

$$\sum_{i+1}^{j+1} [\alpha_i [(y_i - y_m)^2 - \mu] \prod_{\substack{p=1 \\ p \neq i}}^{j+1} (y_p - y_i)^2]^{-1} = 0. \quad \blacksquare$$

LEMMA 5.4. *Let μ_m be the smallest positive zero of the function B_m and let $\mu^* = \min_{1 \leq m \leq j+1} \mu_m$; then*

$$\tfrac{1}{2}\mu^* \leq \min_{\rho \in \mathbb{R}} g(j, p, \rho, y) \leq \mu^*.$$

The proof follows from Lemmas 5.2 and 5.4. Since $(r_j^G(A, b))^2 = \min_{\rho \in \mathbb{R}} g(j, \rho, \alpha, y)$ then Lemma 5.4 gives an upper and a lower bound on $r_j^G(A, b)$.

We are now ready to complete the proof of Theorem 5.1. This part of the proof has been suggested by J. Domsta (1983).

Let y_i, $i = 1, 2, \ldots, j + 1$, be equidistant points distributed uniformly in the interval $[-1, 1]$, i.e.,

$$y_i = -1 + (i - 1)2/j, \qquad i = 1, 2, \ldots, j + 1.$$

Let α_i, $i = 1, 2, \ldots, j + 1$, be defined as

$$\alpha_i = \frac{c}{\prod_{\substack{m=1 \\ m \neq i}}^{j+1} (y_m - y_i)^2}, \qquad i = 1, 2, \ldots, j + 1,$$

where the constant c is chosen in such a way that $\sum_{i=1}^{j+1} \alpha_i = 1$. Let A and b

be the matrix and vector which generate y_i and α_i at the jth step, $i = 1, \ldots, j + 1$. Then from Lemma 5.4 we have

$$\min_{1 \le m \le j+1} \tfrac{1}{2}\mu_m \le (r_j^G(A, b))^2 \le \min_{1 \le m \le j+1} \mu_m,$$

where μ_m is the smallest positive zero of the function

$$\sum_{i=1}^{j+1} \frac{1}{(4/j^2)(i - m)^2 - \mu}.$$

Let m be any integer $1 \le m \le j + 1$. Then μ_m is the smallest positive zero of the function φ_m where

$$\varphi_m(\mu) = -\frac{1}{j^2\mu} + \sum_{i=1}^{m-1} \frac{1}{4(i - m)^2 - j^2\mu} + \sum_{i=m+1}^{j+1} \frac{1}{4(i - m)^2 - j^2\mu}.$$

It is easy to verify that $\varphi_m(1/2j^2) < 0$ for any j, $1 \le m \le j + 1$, since both sums are partial sums of the convergent series $\Sigma_{i=1}^{\infty} (1/(4i^2 - \frac{1}{2}))$, which is bounded by 1. Since the functions φ_m are increasing in the interval $(0, 1/2j^2]$ for all m, $1 \le m \le j + 1$, then we conclude that $\mu_m > 1/2j^2$ for m, $1 \le m \le j + 1$. This means that

$$(r_j(A, b))^2 \ge \frac{1}{2} \min_{1 \le m \le j+1} \mu_m > \frac{1}{4j^2},$$

which completes the proof of Theorem 5.1. ■

We believe that the second part of Theorem 5.1 can be generalized. Numerical experiments suggest

Conjecture 5.1. There exist a real symmetric matrix A, a real unit vector b, and a constant c of order unity such that

$$r_j^G(A, b) \ge \frac{c}{j}\|A\|, \qquad \forall j < n.$$

We now analyze the speed of convergence of the Lanczos algorithm.

THEOREM 5.2. *For every real symmetric matrix A and every real unit vector b,*

$$r_j^L(A, b) \le \frac{1}{\sqrt{j}}\|A\|, \qquad j \ge 1.$$

For every j, $j < n$, there exist a real symmetric matrix A and a real unit vector b such that

$$r_j^L(A, b) \geq \frac{1}{\sqrt{j}+1}\|A\|.$$

Proof. The first part easily follows from (1.3) since $|\beta_j| \leq \|A\|$ and the minimal element of the last components of normalized eigenvectors of a symmetric matrix of size j is at most $1/\sqrt{j}$.

To prove the second part we ocnstruct a matrix A and a unit vector b. Let Y be a $j \times j$ symmetric tridiagonal unreduced matrix such that $\|Y\| = 1$ and each eigenvector of Y has the last component equal to $1/\sqrt{j}$. Define

$$A = \left(\begin{array}{c|c|c} Y & \begin{matrix}0\\ \vdots\\ 0\\ 1\end{matrix} & \begin{matrix}0\\ \vdots\\ 0\end{matrix} \\ \hline 0\ldots01 & 0 & 0\ldots0 \\ \hline \begin{matrix}0\\ \vdots\\ 0\end{matrix} & \begin{matrix}0\\ \vdots\\ 0\end{matrix} & \begin{matrix}0\\ \vdots\\ 0\end{matrix} \end{array}\right), \qquad b = (1, 0, \ldots, 0)^T.$$

Since the matrix Y is unreduced then at the jth step of the Lanczos algorithm we obtain j first columns of the matrix A. From (1.3) we have

$$r_j^L(A, b) = \frac{1}{\sqrt{j}}.$$

From Parlett (1980, p. 69) we have

$$\|A\| \leq 1 + r_j^L(A, b) = 1 + \frac{1}{\sqrt{j}}.$$

Thus

$$r_j^L(A, b) = \frac{1}{\sqrt{j}} \geq \frac{1}{\sqrt{j}}, \qquad \frac{\|A\|}{1 + 1/\sqrt{j}} = \frac{1}{\sqrt{j}+1}\|A\|,$$

which completes the proof the the theorem. ■

Let us stress that since the residuals of the Lanczos algorithm do not

necessarily decrease, it might happen that $r_j^{\mathrm{L}}(A, b) \geq \|A\|/(\sqrt{j} + 1)$, but $r_{j-i}^{\mathrm{L}} \ll \|A\|/(\sqrt{j - i} + 1)$, for some $i, i < j$. However, as numerical experiments suggest, the following conjecture holds:

Conjecture 5.2. There exist a real symmetric matrix A, a real unit vector b, and a constant c of order unity such that

$$r_j^{\mathrm{L}}(A, b) \geq \frac{c}{\sqrt{j}}\|A\|, \qquad \forall j < n.$$

6. Convergence of the gmr algorithm for Real Nonsymmetric Matrices

We prove that the decrease of the residual of the gmr algorithm cannot be guaranteed for the nonsymmetric case. We assume that the matrix A is real and, in general, nonsymmetric. We deal with information $N_j(A, b)$ with a real vector b. Here we consider complex algorithms.

THEOREM 6.1. *There exist a real nonsymmetric matrix A and a unit starting vector b for which*

$$r_1^{\mathrm{G}} = r_2^{\mathrm{G}} = \cdots = r_{n-1}^{\mathrm{G}} = 1.$$

The proof is rather long and may be found in Kuczyński (1983).

Observe that in the class of all real nonsymmetric matrices the gmr algorithm is also strongly optimal. The proof is, in fact, the same as the proof of Theorem 3.2; we need only replace y_2 by Re y_2 or Im y_2 in the proof of Lemma 3.1. Since for some matrices this algorithm does not decrease the residual error until the nth step, we conclude that for the nonsymmetric case the information $N_j(A, b)$ may be too weak. For nonsymmetric matrices we suggest using information not only about the matrix A, but also about A^{T}. We intend to study this problem in the future.

7. Generalizations of the gmr Algorithm

In this section we deal with some other problems which can be solved by gmr-type algorithms.

(i) First we consider the problem of finding p, $p \geq 1$, eigenpairs of a normal matrix. That is, let F be any class of $n \times n$ normal matrices. For a given $\epsilon > 0$ and any matrix A from F find p numbers $\rho_1, \ldots, \rho_p$ and p orthonormal vectors $x_1, \ldots, x_p$ such that

$$\sum_{i=1}^{p} \|Ax_i - \rho_i x_i\|^2 < \epsilon^2. \tag{7.1}$$

Let $N_j(A, b) = [b, Ab, \ldots, A^j b]$, for $b \in S_n$. For simplicity assume that b is chosen such that $\dim A_p = p$, where $A_j = \operatorname{span}(b, \ldots, A^{j-1}b)$. We define the p-gmr algorithm as follows. At the jth step the p-gmr algorithm finds p pairs $(\tilde{x}_1^j, \tilde{\rho}_1^j), \ldots, (\tilde{x}_p^j, \tilde{\rho}_p^j)$ for which $(\tilde{x}_i^j, \tilde{x}_m^j) = \delta_{i,m}$ and

$$\sum_{i=1}^{p} \|A\tilde{x}_1^j - \tilde{\rho}_i^j \tilde{x}_i^j\| \le \sum_{i=1}^{p} \|Ax_i - \rho_i x_i\| \qquad \text{for all } x_1, \ldots, x_p \in A_j,$$

x_i orthonormal, and for all $\rho_1, \ldots, \rho_p \in \mathbb{C}$. As in Section 2 let $k(\Phi, A, b)$ denote the minimal number of steps to solve (7.1) using the algorithm Φ. Then we have

THEOREM 7.1. *If F is unitarily invariant then the p-gmr algorithm is almost strongly optimal in F, i.e.,*

$$k(\Phi^{p\text{-gmr}}, A, b) \le \min_{\Phi} k(\Phi, A, b) + p + 1,$$

for $(A, b) \in F \times S_n$.

The proof is quite similar to the proof of Theorem 3.1. The unitary matrix Q from the proof of Theorem 3.1 is defined here as a product $Q_p Q_{p-1} \ldots Q_1$ of suitable chosen unitary matrices Q_1.

For $p = 1$ Theorem 7.1 coincides with Theorem 3.1.

One can generalize the error criterion (7.1) for approximating p eigenpairs. That is, one wants to find $(x_1, \rho_1), \ldots, (x_p, \rho_p)$ for which $(x_i, x_m) = \delta_{i,m}$ and $f(\|Ax_1 - \rho_1 x_1\|, \ldots, \|Ax_p - \rho_p x_p\|) < \epsilon$, where f is an arbitrary but fixed function $f: \mathbb{R}^p \to \mathbb{R}$. The jth step of the p-gmr algorithm is then modified to find p orthonormal vectors $\tilde{x}_1^j, \ldots, \tilde{x}_p^j$ and p numbers $\tilde{\rho}_1^j, \ldots, \tilde{\rho}_p^j$ such that

$$f(\|A\tilde{x}_1^j - \tilde{\rho}_1^j \tilde{x}_1^j\|, \ldots, \|A\tilde{x}_p^j - \tilde{\rho}_p^j \tilde{x}_p^j\|) \le f(\|Ax_1 - \rho_1 x_1\|, \ldots, \|Ax_p - \rho_p x_p\|)$$

for all $x_1, \ldots, x_p \in A_j$, x_i orthonormal, and for all $\rho_1, \ldots, \rho_p \in \mathbb{C}$. (For simplicity we assume that such vectors and such numbers exist.) Theorem 7.1 then remains true for any function f. For $f(z_1, \ldots, z_p) = (\sum_{i=1}^p z_i^2)^{1/2}$ we get (7.1). For $f(z_1, \ldots, z_p) = \max_{1 \le i \le p} z_i$ the orthonormal x_i and ρ_i, $i = 1, \ldots, p$, satisfy $\max_{1 \le i \le p} \|Ax_i - \rho_i x_i\| < \epsilon$.

(ii) Let H be a Hilbert space and let f be any operator, not necessarily linear, from H to H. Let F be any subclass of operators $f: H \to H$. For a given $\epsilon > 0$ and any operator $f \in F$ we want to find an element $\bar{x}$ such that

$$\|f(\bar{x})\| - \inf_{x \in H} \|f(x)\| < \epsilon, \qquad \text{for } f \in F. \tag{7.2}$$

Let $\{A_j\}_{j=1}^{\infty}$ be the sequence of subspaces of H such that for every positive integer j

$$\begin{aligned} \dim A_j &= j, \\ A_j &\subset A_{j+1}, \\ f(A_j) &\subset A_{j+s}, \qquad \text{where } s \geq 0. \end{aligned}$$

The information operator N_j is given by $N_j(f) = f_j: A_j \to H$, $f_j(x) = f(x)$, $\forall x \in A_j$. That is, we know the restriction of the operator f to the subspace A_j.

By an algorithm we now mean a sequence $\Phi = \{\Phi_i\}$ of arbitrary mappings $\Phi_i: N_i(F) \to H$, $x_i = \Phi_i(N_i(f))$.

Let $V(N_j(f))$ be the set of all operators f from F which share the same information as f, i.e.,

$$V(N_j(f)) = \{\tilde{f} : \tilde{f} \in F, N_j(f) = N_j(\tilde{f})\}.$$

By the index of the algorithm we mean

$$k(\Phi, f) = \min\{j : \|\tilde{f}(x_j)\| - \inf_{x \in H} \|\tilde{f}(x)\| < \epsilon, \forall \tilde{f} \in V(N_j(f))\}.$$

If this set is empty then $k(\Phi, f) = +\infty$.

For simplicity assume that

$$B_j \stackrel{df}{=} \{x : \|f(x)\| = \inf_{z \in A_j} \|f(z)\|\} \neq \emptyset.$$

The gmr* algorithm for the problem (7.2) is defined by

$$x^* = \Phi_j^{\text{gmr*}}(N_j(f)), \qquad \text{where } x_j^* \in B_j.$$

We have

THEOREM 7.2. *If F is unitarily invariant, i.e., $f \in F \Rightarrow Q^{\mathrm{H}} f Q \in F$, for unitary Q, then the gmr* algorithm is almost strongly optimal in F, i.e.,*

$$k(\Phi^{\text{gmr*}}, f) \leq \min_{\Phi} k(\Phi, f) + s + 1, \qquad \forall f \in F.$$

The proof is similar to the proof of Theorem 3.1.

We illustrate Theorem 7.2 by two examples showing that the problems studied in Traub and Woźniakowski (1984) and here are special cases of Theorem 7.2.

EXAMPLE 7.1. *Solution of linear systems.* Let $H = \mathbb{C}^n$ and let $f(x) = Ax - b$, $\forall x \in \mathbb{C}^n$, and A belongs to a given class F of $n \times n$ non-

singular matrices, $b \in \mathbb{C}^n$. Then $\inf_{x\in\mathbb{C}^n} \|f(x)\| = 0$. Thus our problem is to find $x \in \mathbb{C}^n$ such that

$$\|Ax - b\| < \epsilon, \qquad \text{for } A \in F.$$

If $A_j = \operatorname{span}(b, Ab, \ldots, A^{j-1}b)$ then the gmr* algorithm coincides with the minimal residual (mr) algorithm for solving linear systems. Theorem 7.2 yields its almost strong optimality for unitarily invariant classes of matrices, i.e.,

$$k(\Phi^{\mathrm{mr}}, A, b) \le \min_{\Phi} k(\Phi, A, b) + a, \qquad a \le 2.$$

It was proved in Traub and Woźniakowski (1984) that $a \le 1$.

EXAMPLE 7.2. *Finding an eigenpair of a matrix.* Let $H = \mathbb{C}^{n+1}$. Define the operator f for $x \neq 0$ by $f(x, \rho) = ((A - \rho I)(x/\|x\|), 0)$, where A belongs to a subclass F of $n \times n$ matrices, $x \in \mathbb{C}^n$ and $\rho \in \mathbb{C}$. Then

$$\inf_{\substack{x\in\mathbb{C}^n \\ \rho\in\mathbb{C}}} \|f(x, \rho)\| = 0.$$

Thus our problem is to find a nonzero x and a number ρ such that

$$\left\| (A - \rho I)\frac{x}{\|x\|} \right\| < \epsilon.$$

Then the gmr* algorithm for $A_j = \operatorname{span}(b, Ab, \ldots, A^{j-1}b)$ coincides with the gmr algorithm defined in Section 2. Theorem 7.2 reduces now to Theorem 3.1.

8. NUMERICAL RESULTS

In this section we report a few numerical tests of the gmr algorithm for the real symmetric eigenproblem. A sketch of the implementation of this algorithm can be found in Kuczyński (1983) and we do not repeat it here. The Fortran subroutine and extensive numerical tests may be found in Kuczyński (1985). The Fortran subroutine is also available via anonymous FTP as "pub/gmrval" on COLUMBIA·EDU [128.59.16.1] on the Arpanet. Calculations were performed on a DEC-20 computer at Columbia University. This machine has 8-decimal-digit precision ($2^{-28} \approx 0.745 \times 10^{-8}$). We tested symmetric tridiagonal matrices of sizes up to 501 with various coefficients with the vector $b = (1, 0, \ldots, 0)^{\mathrm{T}}$. We were primarily interested in comparing the gmr and Lanczos algorithms. Since the cost of one step of these two algorithms is essentially the same we compare the number of steps which

are required to find an ϵ-approximation. Numerical experiments confirmed the theoretical results of Sections 4 and 5. For all matrices tested and for all ϵ, $\epsilon \geq 10^{-8}$, the number of steps needed to find an ϵ-approximation for the gmr algorithm was no greater than the number of steps required by the Lanczos algorithm. In other words, for every matrix and for every step, the residual of the gmr algorithm was no greater than the corresponding residual of the Lanczos algorithm. For many cases, especially for matrices whose coefficients were randomly selected from the interval $[-\frac{1}{3}, \frac{1}{3}]$ with the uniform distribution, the differences between the residuals of these two algorithms were, in general, small. For random matrices, both algorithms reduced the residual to the level 10^{-8} after about 20 steps. For two examples the number of steps was about 50 and for one example the number was 77 (see Table 8.1). High efficiency of both algorithms for random matrices can be easily explained. We chose the coefficients randomly from the interval $[-\frac{1}{3}, \frac{1}{3}]$. With high probability, the codiagonal contains small elements β_j, which make the eigenproblem easy to solve since $r_j^G \leq r_j^L \leq \beta_j$.

Numerical tests confirmed that the sequence of residuals of the gmr algorithm is always nonincreasing, while the residuals of the Lanczos algorithm do not have this property. Among the 20 matrices reported in Table 8.1, the Lanczos algorithm generated nonincreasing residuals for only 1 matrix. For one of the tested random matrices we obtained $r_{65}^L = 0.145 \times 10^{-6}$ while $r_{66}^L = 0.216 \times 10^{-4}$. Thus the residual of the 66th step of the Lanczos algorithm was more than 150 times larger than the residual of the previous step.

TABLE 8.1

	$\epsilon = 10^{-1}$		$\epsilon = 10^{-2}$		$\epsilon = 10^{-3}$		$\epsilon = 10^{-4}$		$\epsilon = 10^{-5}$		$\epsilon = 10^{-6}$		$\epsilon = 10^{-7}$	
	L	G	L	G	L	G	L	G	L	G	L	G	L	G
*	4	4	7	7	8	8	9	9	10	10	10	10	11	11
	2	2	4	4	10	10	11	11	13	13	14	14	15	15
	2	2	5	5	6	6	10	10	11	11	13	13	15	15
	4	3	9	8	16	14	25	25	31	31	39	36	49	43
	2	2	9	6	14	14	20	20	24	23	29	29	33	33
*	2	2	10	7	29	25	35	35	39	39	44	44	46	46
	3	3	7	5	7	7	7	7	8	8	10	10	13	13
	1	1	9	7	14	14	19	16	24	21	30	30	39	35
	1	1	5	5	7	7	11	11	13	13	15	15	19	19
*	5	4	14	10	30	24	36	34	49	42	58	58	71	64
*	1	1	1	1	7	7	9	9	10	10	12	12	13	13
*	2	2	7	7	9	9	11	11	11	11	13	13	17	17
	2	2	7	7	9	9	11	11	13	13	15	15	19	19
	1	1	7	7	13	12	13	13	16	16	18	18	22	21
	1	1	5	4	7	7	9	9	13	13	17	16	19	18
	4	4	7	7	11	11	13	13	15	15	17	16	19	19
	1	1	3	3	3	3	6	5	9	9	11	11	14	12
	1	1	3	3	4	4	7	6	10	9	14	14	14	14
	2	2	4	4	6	6	10	10	10	10	12	12	14	14
	3	3	11	11	12	12	13	13	14	14	14	14	15	15
Total	44	42	134	118	222	209	285	278	343	331	405	400	477	456
Average	2.2	2.1	6.7	5.9	11.1	10.45	14.25	13.90	17.15	16.55	20.25	20.0	23.85	22.8

The residuals of the Lanczos algorithm increased very often; however, the ratios r_{i+1}^{L}/r_i^{L} were usually slightly larger than one.

Table 8.1 exhibits the number of steps used by the Lanczos algorithm (L) and by the gmr algorithm (G) to reduce the residual to the level less than ϵ for $\epsilon = 10^{-i}$, $i = 1, 2, \ldots, 7$. It was done for 20 tridiagonal matrices with coefficients chosen randomly from the interval $[-\frac{1}{3}, \frac{1}{3}]$ with the uniform distribution. The asterisk in the first column indicates a matrix with constant (fixed) main diagonal.

Finally we discuss two nonrandom examples.

EXAMPLE 8.1. Let A be a tridiagonal matrix of dimension $n = 201$ with diagonal elements $\alpha_i = 0$, $i = 1, 2, \ldots, n$, and codiagonal elements $\beta_i = \frac{1}{2}\sqrt{i/(n-1)}$, $i = 1, 2, \ldots, n-1$ ($\|A\| \leq 1$). For this matrix all residuals of the Lanczos algorithms (up to 200) were constant and

$$r_i^{L} \simeq 0.035, \qquad i = 1, 2, \ldots, n-1.$$

Thus in order to find an ϵ-approximation with any ϵ less than 0.035 using the Lanczos algorithm we have to perform 201 steps, i.e., to solve the full-dimensional 201×201 eigenproblem. The gmr algorithm started with the same residual $r_1^{G} = 0.035$ at the first step and slowly decreased the residuals at every step. We obtained

$r_{10}^{G} \simeq 0.0164$,	$r_{20}^{G} \simeq 0.0120$,	$r_{30}^{G} \simeq 0.0099$,	$r_{40}^{G} \simeq 0.0086$,
$r_{50}^{G} \simeq 0.0077$,	$r_{60}^{G} \simeq 0.0071$,	$r_{70}^{G} \simeq 0.0066$,	$r_{80}^{G} \simeq 0.0062$,
$r_{90}^{G} \simeq 0.0058$,	$r_{100}^{G} \simeq 0.0055$,	$r_{110}^{G} \simeq 0.0053$,	$r_{120}^{G} \simeq 0.0050$,
$r_{130}^{G} \simeq 0.0048$,	$r_{140}^{G} \simeq 0.0047$,	$r_{150}^{G} \simeq 0.0045$,	$r_{160}^{G} \simeq 0.0044$,
$r_{170}^{G} \simeq 0.0042$,	$r_{180}^{G} \simeq 0.0041$,	$r_{190}^{G} \simeq 0.0040$,	$r_{200}^{G} \simeq 0.0039$.

We calculated also the sequences $\{p_j^{(L)}\}$ and $\{p_j^{(G)}\}$ which measure how fast the residuals of the Lanczos algorithm and of the gmr decrease in comparison to the sequence $\{1/j\}$. They are defined by

$$r_j^{L} = (j^{p_j^{(L)}})^{-1}, \qquad r_j^{G} = (j^{p_j^{(G)}})^{-1}$$

for $j = 2, 3, \ldots, n-1$. From Theorems 5.1 and 5.2 we know that $p_j^{(G)} \geq 1$ and $p_j^{(L)} \geq \frac{1}{2}$. (See Table 8.2.)

We believe that for larger dimension n, sequences $\{p_j^{(L)}\}$ and $\{p_j^{(G)}\}$ approach 0.5 and 1, respectively, as j approaches n. The matrix of Example 8.1 suggests how the matrices satisfying Conjectures 5.1 and 5.2 might be constructed.

EXAMPLE 8.2. Consider the tridiagonal matrix A of dimension $n = 501$, with diagonal elements $\alpha_i = 0$, $i = 1, 2, \ldots, n$, and codiagonal elements

TABLE 8.2

j	2	20	40	60	80	100	120	140	160	180	200
$p_j^{(L)}$	4.28	1.12	0.91	0.82	0.76	0.73	0.70	0.68	0.66	0.64	0.63
$p_j^{(G)}$	5.03	1.48	1.29	1.21	1.16	1.13	1.11	1.09	1.07	1.06	1.05

$\beta_i = \frac{1}{2}(i/(n-1))$, $i = 1, 2, \ldots, n-1$. For this matrix numerically computed residuals of the gmr algorithm decreased at exactly every second step, i.e.,

$$r_1^G = r_2^G > r_3^G = r_4^G > r_5^G = r_6^G > \cdots > r_{499}^G = r_{500}^G > r_{501}^G = 0.$$

The residuals of the Lanczos algorithm were increasing at every second step. More precisely, they satisfied the relations

$$r_{2i-2}^{(L)} < r_{2i}^{(L)},\ r_{2i-1}^{(L)} < r_{2i+1}^{(L)}, \qquad i = 2, 3, \ldots, 248$$

and

$$r_{2i+3}^{(L)} > r_{2i}^{(L)}, \qquad i = 1, 2, \ldots, 248.$$

Both algorithms started with $r_1^{(L)} = r_1^{(G)} = 0.001$ and at the final steps they reached $r_{499}^{(L)} \approx r_{500}^{(L)} \approx 0.011$ and $r_{499}^{(G)} \approx r_{500}^{(G)} \approx 0.00036$. Thus, $r_{500}^{(L)}/r_{500}^{(G)} \approx 31$.

From all the tests we have performed we conclude that the gmr algorithm is essentially superior to the Lanczos algorithm for matrices with constant or increasing codiagonal elements. For random matrices or matrices with decreasing codiagonal elements, both algorithms produce nearly the same residuals.

ACKNOWLEDGMENTS

I would like to thank Professor Stefan Paszkowski and Dr. Joachim Domsta for their remarks concerning the proof of the second part of Theorem 5.1. Many thanks are due to Professors Åke Björck and Andrzej Kiełbasinski, who carefully read earlier versions of this paper and suggested many interesting improvements.

REFERENCES

CHOU, A. (1985), On the optimality of Krylov information, in progress.

DOMSTA, J. (1983), private communication.

KUCZYŃSKI, J. (1983), "Optimality and Sketch of Implementation of the Generalized Minimal

Residual Algorithm for Finding an Eigenpair of a Large Matrix", Report LiTH-MAT-R-83-06 of Linköping University.

KUCZYŃSKI, J. (1985), "Implementation of the gmr Algorithm for Large Symmetric Eigenproblems," Report, Columbia University.

NEMIROVSKI, A. S., AND YUDIN, D. B. (1983), "Problem Complexity and Method Efficiency in Optimization," Wiley–Interscience, New York.

PARLETT, B. N. (1980), "The Symmetric Eigenvalue Problem," Prentice–Hall, Englewood Cliffs, N. J.

PASZKOWSKI, S. (1982), private communication.

SZEGÖ, G. (1939), "Orthogonal Polynomials," Amer. Math. Soc. Colloq. Publ. No. 23, New York.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1980), "A General Theory of Optimal Algorithms," ACM Monograph Series, Academic Press, New York.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1984), On the optimal solution of large linear systems, *J. Assoc. Comput. Mach.* **31**, 545–559.