Variations on Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices

Y. Saad
IMAG
BP 53
38041 Grenoble, France

Submitted by Å. Björck

ABSTRACT

It is shown that the method of Arnoldi can be successfully used for solving large unsymmetric eigenproblems. Like the symmetric Lanczos method, Arnoldi's algorithm realizes a projection process onto the Krylov subspace K_m spanned by $v_1, Av_1, \ldots, A^{m-1}v_1$, where v_1 is the initial vector. We therefore study the convergence of the approximate eigenelements obtained by such a process. In particular, when the eigenvalues of A are real, we obtain bounds for the rates of convergence similar to those for the symmetric Lanczos algorithm. Some practical methods are presented in addition to that of Arnoldi, and several numerical experiments are described.

1. INTRODUCTION

Efficient numerical methods for solving large unsymmetric eigenvalue problems are rare. One might mention in particular the simultaneous iteration method, which has been extended in recent years to unsymmetric matrices [5, 19]. Also of interest is singular vector iteration, which reduces the problem to a series of symmetric eigenproblems for which powerful algorithms already exist [9]. Unfortunately, however, we do not have at our disposal such a powerful tool as the symmetric Lanczos method. The biorthogonalization Lanczos algorithm for unsymmetric matrices is unstable and above all does not possess that convergence property which, for the large symmetric problems makes the Lanczos algorithm behave as a rapidly converging iterative method.

This remarkable property of rapid convergence in the symmetric case can be explained by considering the method as a Rayleigh-Ritz projection method on the Krylov subspace K_m spanned by $v_1, Av_1, \ldots, A^{m-1}v_1$, where

52 Vanderbilt Ave., New York, NY 10017 0024-3795/80/060269+27$01.75

v_1 is the starting vector and A the considered matrix [13, 16]. This leads us to ask first whether this convergence property holds for unsymmetric matrices. In other words, if by any process one computes the Ritz elements of A on the Krylov subspace K_m, can we expect some of the Ritz elements to become, as m increases, good approximations to some eigenelements of A? The answer is yes in general, as will be seen.

Any Rayleigh-Ritz-Galerkin projection process applied to an unsymmetric matrix A on the Krylov subspaces K_m will be said to belong to the class of generalized Lanczos methods. The second problem which arises then is to find algorithms based upon the generalized Lanczos methods that can be effectively used for large matrices. The present paper is more particularly devoted to this practical second question than to the first, which will be treated in detail in a forthcoming paper.

The simplest algorithm which achieves the generalized Lanczos method is the one proposed by Arnoldi in 1951 [1]. It reduces the given matrix A sequentially to Hessenberg form. The approximate eigenvalues are obtained by computing the eigenvalues of the Hessenberg matrix H_m of order m, produced at the mth step of Arnoldi's process. However, because of storage considerations, the dimension m of the Hessenberg matrix H_m cannot be chosen as large as necessary to ensure the desired accuracy. This difficulty may be overcome by using the process iteratively like the iterative Lanczos method [15]. Another possibility, which will be discussed, is to perform an incomplete orthogonalization. Then A will be represented by a band Hessenberg matrix, but the basis will no longer be orthogonal.

In Sec. 2 a theoretical introduction to the generalized Lanczos methods is given and some results on the convergence are established. In Sec. 3 we describe the basic Arnoldi algorithm, the iterative Arnoldi algorithm, and the incomplete orthogonalization algorithm. Finally, various numerical experiments are reported in Sec. 4, including comparisons with the simultaneous iteration method.

2. THE GENERALIZED LANCZOS METHOD — THEORETICAL ASPECTS

2.1. *Notations and General Theory*

Given a real matrix A of dimension $N \times N$, and a system $V_m \equiv [v_1, v_2, \ldots, v_m]$ of m linearly independent vectors, the projection method (or Galerkin method) on the subspace span $[V_m]$ aims to approximate an eigenpair λ, ϕ of A by a pair $\lambda^{(m)}, \phi^{(m)}$ satisfying

$$\begin{aligned} &\phi^{(m)} \in \text{span}[V_m], \\ &(A - \lambda^{(m)} I)\phi^{(m)} \perp v_j, \qquad j = 1, 2, \ldots, m. \end{aligned} \tag{2.1}$$

Writing $\phi^{(m)} = V_m y^{(m)}$, it is easily seen from (2.1) that $\lambda^{(m)}, y^{(m)}$ are eigenelements of the generalized eigenvalue problem

$$\left(C_m - \lambda^{(m)} B_m\right) = 0, \tag{2.2}$$

where the $m \times m$ matrices C_m and B_m are defined by $C_m = V_m^T A V_m$, $B_m = V_m^T V_m$. Obviously, $\lambda^{(m)}$ and $\phi^{(m)}$ can be obtained by computing the solutions on the classical eigenvalue problem

$$\left(B_m^{-1} C_m - \lambda^{(m)} I\right) y^{(m)} = 0. \tag{2.3}$$

All the solutions $\lambda^{(m)}$ of the problem (2.1) are usually called Ritz values on the subspace span $[V_m]$. To each Ritz value $\lambda^{(m)}$ is associated a Ritz vector $\phi^{(m)}$ (see [13]).

Throughout the paper the Ritz vectors $\phi^{(m)}$, as well as the eigenvectors ϕ of A, are supposed normalized so that $\|\phi^{(m)}\| = 1$, $\|\phi\| = 1$, where $\|\cdot\|$ denotes the (complex) Hermitian norm associated with the scalar product $(\cdot, \cdot)$.

Many applications of the projection method involve an orthonormal system V_m, so that B_m in (2.2) and (2.3) reduces to the identity matrix. The best illustration of such a process is the symmetric Lanczos method. It uses as V_m the orthonormal system obtained by orthogonalizing the Krylov vectors $v_1, Av_1, \ldots, A^{m-1}v_1$, where v_1 is a starting vector. In this case, the matrix C_m produced is tridiagonal, and Lanczos has made the remarkable and very useful observation that the sequence v_m, $m = 1, 2, \ldots$, as well as the tridiagonal matrices C_m can be obtained from a simple three term recurrence formula [8].

We shall denote by K_m the Krylov subspace of $\mathbb{C}^N$ spanned by the vectors $v_1, Av_1, \ldots, A^{m-1}v_1$. *Any projection method on the subspace K_m applied to unsymmetric matrices will be referred to as a generalized Lanczos method.*

It is important to introduce the orthogonal projector π_m on the subspace K_m (see [18]). Another way of stating the problem (2.1) is then

$$\pi_m (A - \lambda^{(m)} I) \phi^{(m)} = 0,$$

so that the Ritz vectors are nothing but the eigenvectors of the operator $A_m = \pi_m A \pi_m$ which belong to K_m, and the Ritz values are the corresponding eigenvalues. Note that each vector belonging to $K_m^{\perp}$ is an eigenvector of A_m associated with the eigenvalue 0. These will be called trivial eigenvectors.

2.2. The Convergence

In this section a brief analysis of the convergence will be given. Given an eigenvalue λ of A, with associated eigenvector ϕ, we may naturally ask whether there is a (finite) sequence of approximate eigenelements $\lambda^{(m)}, \phi^{(m)}$ which converge rapidly to λ, ϕ. Indeed, it is usually observed that the Ritz elements $\lambda^{(m)}, \phi^{(m)}$ converge so rapidly to the exact eigenelements that satisfactory accuracy is obtained for values of m much smaller than the dimension N of the matrix A.

In order to analyze this property of fast convergence, it is important to establish certain *a priori* error bounds. This will be accomplished in two stages. We first give error bounds in terms of $\|(I-\pi_m)\phi\|$, the distance between ϕ and the Krylov subspace K_m (Sec. 3.2.1). Then the behavior of $\|(I-\pi_m)\phi\|$ is studied in Sec. 3.2.2.

Let us mention that the theory developed in [6], [11], and [16], for the symmetric Lanczos method, uses mainly the Courant characterization of the eigenvalues, which is no longer valid here, since the operator A is not self-adjoint. Consequently, we suggest another approach using the residual vectors. If one wants to assert that the nontrivial eigenpair $\lambda^{(m)}, \phi^{(m)}$ of A_m is close to an eigenpair λ, ϕ of A, one should show that either of the residuals $(A-\lambda^{(m)}I)\phi^{(m)}$ or $(A_m-\lambda I)\phi$ is small [20].

The second of these possibilities is considered in the following theorem.

THEOREM 2.1. *Let* $\gamma_m = \|\pi_m A(I-\pi_m)\|$. *Then*

$$\|(A_m-\lambda I)\phi\| \leq \sqrt{|\lambda|^2+\gamma_m^2}\,\|(I-\pi_m)\phi\|. \tag{2.4}$$

Proof. We have

$$\begin{aligned}(A_m-\lambda I)\phi &= \pi_m(A-\lambda I)\pi_m\phi-\lambda(I-\pi_m)\phi\\ &= \pi_m(A-\lambda I)(\pi_m\phi-\phi)-\lambda(I-\pi_m)\\ &= -\pi_m(A-\lambda I)(I-\pi_m)\phi-\lambda(I-\pi_m)\phi.\end{aligned}$$

Since $I-\pi_m$ is a projector, the factor $(I-\pi_m)\phi$ can be replaced by $(I-\pi_m)^2\phi$ to yield $(A_m-\lambda I)\phi = -\pi_m(A-\lambda I)(I-\pi_m)(I-\pi_m)\phi-\lambda(I-\pi_m)\phi$. The two vectors on the right hand side are orthogonal; thus

$$\begin{aligned}\|(A_m-\lambda I)\phi\|^2 &= \|\pi_m(A-\lambda I)(I-\pi_m)(I-\pi_m)\phi\|^2\\ &\quad +|\lambda|^2\|(I-\pi_m)\phi\|^2.\end{aligned} \tag{2.5}$$

Observing that the first term on the right hand side satisfies

$$\|\pi_m(A-\lambda I)(I-\pi_m)(I-\pi_m)\phi\| \leqslant \|\pi_m(A-\lambda I)(I-\pi_m)\phi\|\,\|(I-\pi_m)\phi\|$$
$$= \gamma_m\|(I-\pi_m)\phi\|,$$

then (2.4) follows directly from (2.5). ■

2.3. *Inequalities for* $\|(I-\pi_m)\phi\|$

From now on we shall assume for simplicity that the eigenvalues of A are all simple. The starting vector v_1 of the Krylov subspace K_m can then be written as

$$v_1 = \sum_{j=1}^{N} \alpha_j \phi_j, \tag{2.6}$$

where $\{\phi_j\}_{j=1,2,\ldots,N}$ is a basis of $\mathbb{C}^N$ formed by eigenvectors of A, of norm 1. We shall denote by P_k the space of polynomials of degree not exceeding k. The next proposition is easy to prove.

PROPOSITION 2.1. *Let us assume that* $\alpha_i \neq 0$ *and let* $\xi_i = \Sigma_{j=1, j\neq i}^{N} |\alpha_j|/|\alpha_i|$. *Then*

$$\|(I-\pi_m)\phi_i\| \leqslant \xi_i \min_{\substack{p\in\mathsf{P}_{m-1}\\ p(\lambda_i)=1}} \max_{\substack{j=1,2,\ldots,N\\ j\neq i}} |p(\lambda_j)|. \tag{2.7}$$

We shall set throughout

$$\varepsilon_i^{(m)} = \min_{\substack{p\in\mathsf{P}_{m-1}\\ p(\lambda_i)=1}} \max_{\substack{j=1,\ldots,N\\ j\neq i}} |p(\lambda_j)|, \tag{2.8}$$

so that Proposition 2.1 now reads

$$\|(I-\pi_m)\phi_i\| \leqslant \xi_i \varepsilon_i^{(m)}. \tag{2.9}$$

The polynomial $\bar{p}(z)$ which achieves the minimum in (2.8) is the best uniform approximation of the null function on the (discrete) set $\{\lambda_j\}_{j\neq i}$ by polynomials of degree $m-1$ satisfying the constraint $p(\lambda_i)=1$. $\varepsilon_i^{(m)}$ is therefore the so-called degree of approximation of the null function by these

polynomials [11], and the inequality (2.9) shows that the problem of studying the convergence is reduced to that of estimating this degree of approximation.

The problem of estimating the degree of approximation is a difficult one. Except for some particular shapes of spectra, such as purely real spectra or almost purely real spectra, it will not be easy to establish bounds for $\varepsilon_i^{(m)}$ which are at the same time sharp and simple.

Consideration is however given below to the cases of purely real spectra. The general case will be studied in a forthcoming paper which will be devoted to a detailed analysis of the convergence. It is also briefly discussed in [17].

THEOREM 2.2. *Assume that all the eigenvalues of A are real and simple, and number them in descending order:*

$$\lambda_1 > \lambda_2 > \cdots > \lambda_N.$$

Set

$$\gamma_i = 1 + 2\frac{\lambda_i - \lambda_{i+1}}{\lambda_{i+1} - \lambda_N}$$

and

$$\kappa_i = \prod_{j=1}^{i-1} \frac{\lambda_j - \lambda_N}{\lambda_j - \lambda_i} \qquad \text{if} \quad i \neq 1,$$

$$\kappa_1 = 1.$$

Then

$$\varepsilon_i^{(m)} \leqslant \frac{\kappa_i}{T_{m-i}(\gamma_i)}, \tag{2.10}$$

where $T_k(x)$ is the k th degree Tchebycheff polynomial of the first kind.

Proof. Let Q_i be the space of all polynomials of the form $q(x) = l_i(x) r(x)$, where

$$l_i(x) = \frac{(x-\lambda_1)(x-\lambda_2)\cdots(x-\lambda_{i-1})}{(\lambda_i-\lambda_1)(\lambda_i-\lambda_2)\cdots(\lambda_i-\lambda_{i-1})}$$

[when $i=1$ take $l_1(x)=1$], and where r is a polynomial of degree not exceeding $m-i$, satisfying $r(\lambda_i)=1$. Clearly $q\in\mathsf{P}_{m-1}$ and $q(\lambda_i)=1$, so that from (2.8) we get

$$\varepsilon_i^{(m)} \leqslant \min_{q\in Q_i} \max_{\substack{j=1,\dots,N \\ j\neq i}} |q(\lambda_j)|. \tag{2.11}$$

Since $q(\lambda_j)=0$, $j=1, i-1$, the maximum in (2.11) is attained for a certain λ_j with $j>i$. Hence

$$\begin{aligned}\varepsilon_i^{(m)} &\leqslant \min_{\substack{r\in\mathsf{P}_{m-i} \\ r(\lambda_i)=1}} \max_{i<j\leqslant N} \left|\frac{(\lambda_j-\lambda_1)\cdots(\lambda_j-\lambda_{i-1})}{(\lambda_i-\lambda_1)\cdots(\lambda_i-\lambda_{i-1})}\right| r(\lambda_j) \\ &\leqslant \kappa_i \min_{\substack{r\in\mathsf{P}_{m-i} \\ r(\lambda_i)=1}} \max_{i<j\leqslant N} |r(\lambda_j)| \\ &\leqslant \kappa_i \min_{\substack{r\in\mathsf{P}_{m-i} \\ r(\lambda_i)=1}} \max_{\lambda\leqslant x\leqslant\lambda_{i+1}} |r(x)|. \end{aligned} \tag{2.12}$$

Now it is well known that the minimax term in (2.12) is equal to $[T_{m-i}(\gamma_i)]^{-1}$ (see [10]), which completes the proof. ■

A comparison with the results in [16] shows that when the eigenvalues are all real, the rates of convergence of the generalized Lanczos methods are bounded by quantities similar to those of the symmetric Lanczos process.

REMARKS.

(1) As indicated in [16], the right side of (2.10) is of the same order as $2\kappa_i\tau_i^{-m+i}$, where $\tau_i=\gamma_i+\sqrt{\gamma_i^2-1}$. So one can take τ_i as a lower bound for the *rate* of convergence of the process. For example, when $-1=\lambda_N<\lambda_{N-1}<\cdots<\lambda_2=1<\lambda_1$ we have

$$\tau_1=\lambda_1+\sqrt{\lambda_1^2-1}\ .$$

(2) There is no difficulty in proving a result similar to (2.10) for the case of purely imaginary spectra or for the cases where the spectrum lies *on a straight line* of the complex plane.

3. PRACTICAL METHODS

In this section some algorithms based upon the generalized Lanczos method will be presented. We first shall describe the method of Arnoldi in Sec. 3.1. Then in order to remedy some practical difficulties encountered with Arnoldi's method, an alternative algorithm, which will be called the incomplete orthogonalization algorithm, will be proposed in Sec. 3.2.

3.1. *The Method of Arnoldi*

Let v_1 be a starting vector of norm one, and let m be chosen not exceeding the dimension N of the matrix A. The method of Arnoldi computes a sequence of vectors $v_1, v_2, \ldots, v_m$ by the recurrence

$$h_{j+1,j}v_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij}v_i, \tag{3.1}$$

where the h_{ij}, $i=1, j+1$, are chosen in such a way that $v_{j+1} \perp v_i$, $i=1,\ldots,j$, and $\|v_{j+1}\| = 1$. This can be done by the following algorithm.

ALGORITHM 3.1.

For $j := 1$ until m do

1. $w := Av_j$.
2. For $i := 1$ until j do

$$w := w - v_i \times (h_{ij} := (Av_j, v_i)). \tag{3.2}$$

3.

$$v_{j+1} := w/(h_{j+1,j} := \|w\|). \tag{3.3}$$

One notes that the algorithm stops for $j \leqslant m$ if $h_{j+1,j}$ vanishes. For simplicity it will be assumed throughout that at each step j we have $h_{j+1,j} \neq 0$, $j=2,3,\ldots,m$. This is not actually a strong restriction on the vector v_1. Indeed, it can be shown that it is equivalent to the following assumption.

$$\textit{For any polynomial } p, \quad p(A)v_1 = 0 \rightarrow \textit{degree}(p) \geqslant m. \tag{3.4}$$

In other words, *the annihilating polynomial of* v_1 *has degree not less than* m. Note that it is also equivalent to the fact that $\dim(K_m)=m$ (see [15] for the proofs of analogous results in the symmetric case). The above algorithm builds up v_{j+1} from $v_1, v_2, \ldots, v_j$ by first computing the vector Av_j, which is then orthonormalized against the vectors $v_1, v_2, \ldots, v_j$. The following theorem is easy to prove:

THEOREM 3.1.

(1) *The system* $\{v_1, v_2, \ldots, v_m\}$ *computed from the algorithm* (3.1) *is an orthonormal basis of the subspace* K_m *spanned by* $\{v_1, Av_1, \ldots, A^{m-1}v_1\}$.

(2) *Let* V_m *be the* $N \times m$ *matrix formed by the column vectors* $v_1, v_2, \ldots, v_m$. *Then the matrix* $H_m = V_m^T A V_m$ *is an upper* $m \times m$ *Hessenberg matrix with elements* h_{ij} *given by* (3.2) *and* (3.3).

REMARKS.

(1) This result is true in exact arithmetic. In finite precision the computation of w by (3.2) undergoes a severe cancellation, so that the resulting system $\{v_1, v_2, \ldots, v_m\}$ can be far from orthonormal. Reorthogonalization is usually an effective remedy for loss of orthogonality [3]. Practically we shall use the modified Gram-Schmidt method developed in [3]. It performs reorthogonalization only when an important cancellation occurs, and carries on reorthogonalization as long as cancellation persists. We do not intend to describe this algorithm in detail, but we would like to point out that our experiments suggest that the parameters ω and θ invoked in [3] should be so chosen that they ensure strong orthogonality ($\theta = \sqrt{2}$) in the beginning of Arnoldi's process ($j \leqslant [m/2]$ say) and a less perfect orthogonality at the end. Any orthogonalization scheme involved in the present paper will refer to the modified Gram-Schmidt method mentioned above.

(2) If the matrix A is symmetric then H_m reduces to a symmetric tridiagonal matrix and the algorithm (3.1) reduces to the symmetric Lanczos method.

Theorem 3.1 means that the matrix $B_m^{-1}C_m$ of the problem (2.3) is simply the Hessenberg matrix H_m computed by Arnoldi's algorithm. Therefore

COROLLARY 3.1. *The Ritz values of* A *in* K_m *are the eigenvalues* $\lambda_i^{(m)}$ *of* H_m, *and the Ritz vectors are the vectors* $V_m y_i^{(m)}$, *where the* $y_i^{(m)}$ *are eigenvectors of* H_m *associated with the* $\lambda_i^{(m)}$.

As with the symmetric Lanczos method, one might easily compute the residual norms by using the formula $\|(A-\lambda^{(m)}I)\phi^{(m)}\| = h_{m+1,m}|e_m^T y^{(m)}|$,

where e_m is the m-dimensional vector $e_m = (0, 0, \ldots, 0, 1)^T$. This is a direct consequence of the following equality, which derives from the algorithm (3.1):

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T.$$

It is therefore quite simple to check step by step (or periodically) whether the desired accuracy is attained and to stop as soon as it is so. The inverse iteration method is perfectly suited for cheaply computing the successive eigenvectors $y^{(m)}$ of H_m.

On the practical side, there remains the problem of choosing the number m of steps. In theory m should be taken as large as necessary to ensure good accuracy. In practice, however, storage considerations will not allow this.

The storage of the Hessenberg matrix requires about $\frac{1}{2} m^2$ locations, and if one wants to store the vectors v_j in core memory, then $N \times m$ extra locations are needed. When N is large, this may become impossible if we consider that convergence is often achieved for values of m such as $m \simeq \sqrt{N}$. Therefore the number of steps, m, is limited by the available core memory. After computing the Ritz values and Ritz vectors with the maximum possible m according to core memory capacity, one may find the eigenelements have not converged to the desired accuracy.

The simplest way to overcome this difficulty is to repeat the process with v_1 replaced by a Ritz vector or a combination of Ritz vectors. This will be called the iterative Arnoldi algorithm. The idea is essentially the same as that developed in the iterative Lanczos method [11] and the iterative block-Lanczos method [4].

For example, the iterative Arnoldi algorithm for computing the *dominating* eigenvalue λ_1 of A ($|\lambda_1| > |\lambda_j|$ for $j = 2, \ldots, N$) is the following:

ALGORITHM 3.2.

1. Choose m and v_1.
2. Construct V_m and H_m by Algorithm 3.1, and compute the dominating Ritz value $\lambda_1^{(m)}$ and the associated Ritz vector $\phi_1^{(m)}$.
3. If the pair $\lambda_1^{(m)}, \phi_1^{(m)}$ is sufficiently accurate, then stop. Otherwise take $v_1 = \phi_1^{(m)}$ and go back to 2.

REMARKS.

(1) For simplicity the above algorithm is described only for the problem of computing one eigenvalue and the associated eigenvector. p dominant eigenvalues can be computed by the iterative Arnoldi algorithm as well. One

should then take as a new starting vector v_1' in step 3 a combination of $\phi_1^{(m)}, \phi_2^{(m)}, \ldots, \phi_p^{(m)}$. We suggest the following combination:

$$\alpha \cdot v_1' = \sum_{j=1}^{p} \left\| \left(A - \lambda_j^{(m)} I\right) \phi_j^{(m)} \right\| \operatorname{Re}\left(\phi_j^{(m)}\right),$$

where α is a normalizing factor. The first reason for this choice is to avoid complex arithmetic. It is important to note that $\operatorname{Re}(\phi_i^{(m)})$ is a combination of the complex pair of eigenvectors $\phi_i^{(m)}$ and $\overline{\phi_i^{(m)}}$, so that v_1' represents $\phi_i^{(m)}$ in the same way as $\overline{\phi_i^{(m)}}$. The second reason is to attempt to balance the accuracy of the desired eigenvectors. The residual norms introduced in the above combination favor the eigenvectors which converge slowly. Thus the slow convergence is set off by an initial vector which is richer in the corresponding exact eigenvectors.

(2) The dominant eigenvalues are not the only ones which converge rapidly. In effect, the best accuracy is first obtained for those eigenvalues which lie on the outer part of the spectrum. This generalizes the well-known property of the Lanczos method for symmetric matrices, which states that the best accuracy is first achieved for the largest eigenvalues as well as the smallest (see [6], [11], and [16]).

Our experiments reveal that the iterative Arnoldi algorithm is competitive with the simultaneous iteration method of [5] and [18], even when full reorthogonalization is used. (See our experiments in Sec. 4.1.)

As with the symmetric Lanczos method, we also observed that the convergence is slowed down when Arnoldi's algorithm is used iteratively. This means that in general, if Algorithm 3.1 were run with a dimension m of the form $m = p \times m'$, then it would provide much better accuracy than with p iterates of Algorithm 3.2 with the dimension m'.

3.2. *The Incomplete Orthogonalization Algorithm*

3.2.1. The Algorithm, Motivation, and Basic Properties. In what follows we try to develop algorithms which require less core memory than Arnoldi's. The development of such methods is based upon the observation made in many experiments, that the elements h_{ij} of the matrix H_m become slowly smaller as j increases (with i fixed). This suggests that one should orthogonalize the vector Av_j produced after the jth step, against the previous $p+1$ vectors $v_{j-p}, v_{j-p+1}, \ldots, v_j$, rather than against all the previous vectors $v_1, \ldots, v_j$. Such a process will be referred to as the incomplete orthogonalization algorithm. Note that the system $\{v_1, \ldots, v_m\}$ now has no reason to be orthonormal, so that we need to relate this algorithm with the generalized

Lanczos method in a certain way. Before doing so, let us first describe the algorithm in its simplest form. The vector v_1 is again an initial vector of norm 1, and the integers p and m satisfy $p+1<m\leqslant N$.

ALGORITHM 3.3 (Incomplete orthogonalization)

For $j:=1$ until m do

1. $w:=Av_j$.
2. For $i:=\max\{1, j-p\}$ until m do

$$w:=w-v_i\times(h_{ij}:=(Av_j, v_i)). \tag{3.5}$$

3.

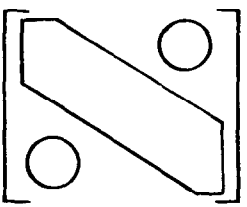
$$v_{j+1}:=w/(h_{j+1,j}:=\|w\|). \tag{3.6}$$

Here again, in practice, stage 2 should be replaced by the more stable version of the Gram-Schmidt algorithm described in [3].

The assumption (3.4) ensures again that the algorithm does not stop before the mth step, in other words that $h_{j+1,j}\neq 0$, $j=1,\ldots,m-1$.

We denote by $\tilde{H}$ the band-Hessenberg matrix whose nonzero elements are those elements h_{ij} in position (i, j) satisfying $i-1\leqslant j\leqslant i+p+1$ and given by (3.5) and (3.6). Therefore

$$\tilde{H}_m= \tag{3.7}$$



With this matrix in hand, one may ask how to define the Ritz elements.

According to (2.1), and since in general the system $\{v_1, v_2,\ldots, v_m\}$ is not orthonormal, the Ritz values are now the eigenvalues of the matrix $B_m^{-1}C_m$, where $B_m=V_m^TV_m$, $C_m=V_m^TAV_m$. But in practice it would be prohibitive to actually perform the computation of the matrices B_m, C_m, and $B_m^{-1}C_m$. A useful property, which will simplify this problem, is that, *except for a rank one perturbation matrix*, $B_m^{-1}C_m$ *is equal to* $\tilde{H}_m$. This is stated in the following analogue of Theorem 3.1.

THEOREM 3.2.

(1) *The system* $\{v_1, \ldots, v_m\}$ *computed from Algorithm* 3.3 *is a basis of the Krylov subspace* K_m *and satisfies the following property (incomplete orthogonality property):*

$$(v_i, v_j) = \delta_{ij} \qquad \text{for} \quad |i-j| \leq p+1, \tag{3.8}$$

where δ_{ij} *denotes the Kronecker symbol.*

(2) *Let* V_m *be the* $N \times m$ *matrix formed by the column vectors* $v_1, v_2, \ldots, v_m$ *and set* $B_m = V_m^T V_m$, $C_m = V_m^T A V_m$, $\hat{H}_m = B_m^{-1} C_m$, *and* $r_m = h_{m+1,m} B_m^{-1} V_m^T v_{m+1}$. *Then*

$$\hat{H}_m = \tilde{H}_m + r_m e_m^T. \tag{3.9}$$

Proof. (1): Under the assumption (3.4), the vectors $v_1, Av_1, \ldots, A^{m-1}v_1$ form a basis of the subspace K_m, so that each vector v_{j+1}, $j \leq m-1$ can be written as $v_{j+1} = \sum_{k=0}^{j} \alpha_k A^k v_1$. It is clear from (3.5) and (3.6) that α_j is just $[\prod_{k=1}^{j} h_{k+1,k}]^{-1}$, which is nonzero. This shows that $v_1, v_2, \ldots, v_m$ are linearly independent. That (3.8) is satisfied can easily be shown by induction.

(2): From the algorithm we can write for $j = 1, 2, \ldots, m$

$$Av_j = \sum_{i=0}^{j+1} h_{ij} v_i, \qquad \text{where} \quad i_0 = \max(1, j-p),$$

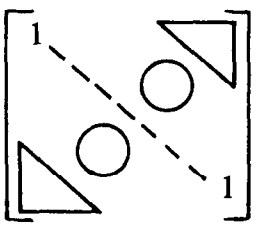
yielding the matrix equation

$$AV_m = V_m \tilde{H}_m + h_{m+1,m} v_{m+1} e_m^T. \tag{3.10}$$

Multiplying on the left by V_m^T gives $C_m = B_m \tilde{H}_m + h_{m+1,m} V_m^T v_{m+1} e_m^T$, and (3.9) results on multiplying again on the left by B_m^{-1}. ∎

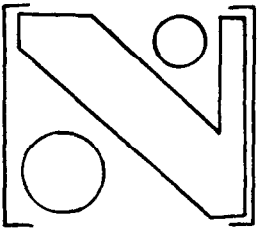
The first part of the theorem states that the matrix B_m has the following form:

$B_m =$



The second part implies that the matrix $B_m^{-1}C_m$ that is required for the solution of problem (2.3) has the form:

$$\hat{H}_m = \tag{3.11}$$



It differs from $\tilde{H}_m$ only in its last column. It is interesting to remark that $r_m = h_{m+1,m}V_m^+ v_{m+1}$, where V_m^+ denotes the pseudoinverse of V_m (see [18]).

The next corollary of Theorem 3.2 is the analogue of Corollary 3.1.

COROLLARY 3.2. *The Ritz values of A in K_m are the eigenvalues $\lambda_i^{(m)}$ of $\hat{H}_m$, and the Ritz vectors are the vectors $V_m \cdot y_i^{(m)}$, where the $y_i^{(m)}$ are eigenvectors of $\hat{H}_m$ associated with the $\lambda_i^{(m)}$.*

So far nothing has been said about the practical computation of the Ritz elements. Computing the eigenelements of the matrix (3.11) does not cause any difficulty. But before that one should have already computed the perturbation $r_m e_m^T$ in (3.9). Now this can be expensive, since the matrix $B_m = V_m^T V_m$ is required to obtain the vector r_m. We can avoid this difficulty in two different ways. The first way is to *ignore the perturbation $r_m e_m^T$*.

Indeed, even though this perturbation is not negligible, one observes the surprising fact that the approximate eigenpairs obtained on replacing $\hat{H}_m$ by $\tilde{H}_m$ are also good approximations to certain eigenelements of A, and that the accuracy improves as m increases until a certain dimension is attained. After that the accuracy starts decreasing slowly, so one should stop at this stage and restart the process if necessary. This process will be described in Sec. 3.2.2.

The second way to avoid the abovementioned difficulty is to attempt to actually compute the last column of the matrix $\hat{H}_m$ by particular means without having to use the matrix B_m explicitly. This will be described in Sec. 3.2.3.

3.2.2. Incomplete Orthogonalization without Correction. Suppose that we neglect the rank one correction matrix $r_m e_m^T$ in (3.9) and consider instead of the Ritz elements $\lambda^{(m)}, \phi^{(m)}$ the *approximate eigenelements* $\mu^{(m)}, \psi^{(m)}$, where $\mu^{(m)}$ is an eigenvalue of $\tilde{H}_m$, and $\psi^{(m)}$ is related to the corresponding

eigenvector $z^{(m)}$ according to $\psi^{(m)} = V_m z^{(m)}$. There are two important points that should be considered concerning the approximation of λ, ϕ by a pair $\mu^{(m)}, \psi^{(m)}$. The first point is to give an explanation of the fact that $\psi^{(m)}$ provides a good approximation to ϕ. The second is to provide reliable stopping criteria in order to prevent the accuracy of the approximate eigenvector $\psi^{(m)}$ from deteriorating by stopping the process at a suitable step.

The next proposition clarifies in a certain measure the first point. let $x^{(m)}, y^{(m)}$ denote an eigenpair of $\hat{H}_m$, $\phi^{(m)} = V_m \cdot y^{(m)}$ denote the Ritz vector, and $\mu^{(m)}, z^{(m)}, \psi^{(m)}$ be defined as above. Also set $\hat{v}_{j+1} = A v_j - \sum_{i=i_0}^{j} k_{ij} v_i$, $i_0 = \max(1, j-p)$.

PROPOSITION 3.1.

(1) *Set* $q_m = (I - \pi_m)\hat{v}_{m+1}$, *where* π_m *is the orthogonal projection on* K_m. *Then the residual of the pair* $\lambda^{(m)}, \phi^{(m)}$ *by* A *satisfies*

$$(A - \lambda^{(m)} I)\phi^{(m)} = e_m^T y^{(m)} \cdot q_m. \tag{3.12}$$

(2) *Let* r_m *be defined as in Theorem* 3.2. *Then the residual of the pair* $\lambda^{(m)}, y^{(m)}$ *by* $\tilde{H}_m$ *satisfies*

$$(\tilde{H}_m - \lambda^{(m)} I) y^{(m)} = -e_m^T y^{(m)} \cdot r_m. \tag{3.13}$$

Proof. (1): Let us first remark that the vector q_m is also equal to $\hat{v}_{m+1} - V_m r_m$. Indeed, $q_m = \hat{v}_{m+1} - \pi_m \hat{v}_{m+1}$ and, as is well known, $\pi_m \hat{v}_{m+1}$ is equal to $V_m (V_m^T V_m)^{-1} V_m^T \hat{v}_{m+1}$, which is also $V_m r_m$. Now from (3.10) and (3.9) one gets $(A - \lambda^{(m)} I) V_m = V_m (\hat{H}_m - \lambda^{(m)} I) + (\hat{v}_{m+1} - V_m r_m) e_m^T$. On multiplying by $y^{(m)}$ one gets

$$(A - \lambda^{(m)} I) V_m y^{(m)} = V_m \cdot (\hat{H}_m - \lambda^{(m)} I) y^{(m)} + (\hat{v}_{m+1} - V_m r_m) e_m^T y^{(m)}.$$

The first term on the right side vanishes, and the second term is $e_m^T y^{(m)} q_m$, which is the desired result.

(2): From (3.10) one gets easily

$$(A - \lambda^{(m)} I) V_m = V_m [\tilde{H}_m - \lambda^{(m)} I] + \hat{v}_{m+1} e_m^T.$$

Multiplying on the right by $y^{(m)}$ yields

$$(A-\lambda^{(m)}I)\phi^{(m)}=V_m(\tilde{H}_m-\lambda^{(m)}I)y^{(m)}+\hat{v}_{m+1}e_m^Ty^{(m)}.$$

By multiplying on the left by V_m^T, we obtain

$$0=B_m(\tilde{H}_m-\lambda^{(m)}I)y^{(m)}+V_m^T\hat{v}_{m+1}e_m^Ty^{(m)},$$

so that $(\tilde{H}_m-\lambda^{(m)}I)y^{(m)}=-B_m^{-1}V_m^T\hat{v}_{m+1}e_m^Ty^{(m)}$ and the proof is complete. ■

The relations (3.12) and (3.13) of the proposition can be gathered together into one formula as follows:

$$\|(\tilde{H}_m-\lambda^{(m)}I)y^{(m)}\|=\frac{\|r_m\|}{\|q_m\|}\|(A-\lambda^{(m)}I)\phi^{(m)}\|. \tag{3.14}$$

This equality implies that as long as the factor $\|r_m\|/\|q_m\|$ remains not too large, the eigenelements $\lambda^{(m)}$, $y^{(m)}$ of $\hat{H}_m$ admit a residual which is of the same order of magnitude as $\|(A-\lambda^{(m)}I)\phi^{(m)}\|$. Note that $\|r_m\|$ can be bounded by $\tau(B_m)\cdot h_{m+1,m}$, where $\tau(B_m)$ is the condition number of B_m. Furthermore, $\|q_m\|$ is also equal to $\|(I-\pi_m)Av_m\|$, which measures how independent Av_m is of the previous vectors $v_1,\ldots,v_m$.

In order to obtain a reliable algorithm based upon the incomplete orthogonalization algorithm, it is of prime importance to find a means for checking periodically the accuracy of an approximate eigenpair $\mu^{(m)}, \psi^{(m)}$. This can be done by computing periodically the residual norm $\|(A-\mu^{(m)}I)\psi^{(m)}\|$. In practice, however, this is uneconomical, since it requires that the vector $\psi^{(m)}$ be formed. The following proposition solves this difficulty by providing a simple way for computing this residual norm in terms of the last component of the vector $z^{(m)}$.

PROPOSITION 3.2. *The residual of the approximate eigenpair* $\mu^{(m)}, \psi^{(m)}$ *satisfies*

$$\|(A-\mu^{(m)}I)\psi^{(m)}\|=h_{m+1,m}|e_m^Tz^{(m)}|. \tag{3.14}$$

Proof. This result is an immediate consequence of the equality (3.10). ■

Note here that if $\|z^{(m)}\|=1$, then it is not true that $\|\psi^{(m)}\|=1$, as would be the case of V_m were orthonormal. However, our experience is that the

right side of (3.14) still provides a good estimate of the actual residual norm. The reason for this is that, in general, the dominating components of $z^{(m)}$ are the first ones and that the first vectors of the system V_m are orthonormal (or nearly orthonormal). This suggests the following algorithm.

ALGORITHM 3.4.

1. Choose an integer p, sufficiently large, a starting vector v_1, and a tolerance ε.
2. For $m=1,2,\ldots,$ do:
   (a) Construct the matrix $\tilde{H}_m$ by the Algorithm 3.3 and store the vectors v_m in auxiliary memory (if necessary).
   (b) Compute periodically the desired eigenvalues $\mu^{(m)}$ and eigenvectors $z^{(m)}$ of $\tilde{H}_m$, and estimate the residuals by the formula (3.14). If all the residual norms are less than ε, then stop.
   If two successive residual norms have a quotient greater than a tolerance τ (for example $\tau=1$), then go to 3, else continue.
3. Form an appropriate combination of the approximate eigenvectors $\psi_i^{(m)}$ and go back to 2 with v_1 replaced by this combination.

For the step 2(b) the eigenelements are first computed using the QR method and the inverse iteration method for the eigenvectors. After that one should only make use of a Rayleigh quotient iteration.

For the third step see Remark 1 following Algorithm 3.2.

In this way the algorithm gives satisfactory results and is competitive with the simultaneous iteration method and with the iterative Arnoldi process (see Sec. 4).

3.2.3. Correcting the Matrix $\tilde{H}_m$. We need the following corollary of Theorem 3.2, which gives a simple expression of the required last column of $\hat{H}_m$.

COROLLARY 3.3. *The last column of $\hat{H}_m$ is equal to $V_m^+ A v_m$.*

Proof. Multiplying Eq. (3.10) on the right by e_m gives

$$Av_m = V_m H_m e_m + h_{m+1,m} v_{m+1}.$$

Therefore

$$\begin{aligned} V_m^T A v_m &= V_m^T V_m \tilde{H}_m e_m + h_{m+1,m} V_m^T v_{m+1}, \\ V_m^+ A v_m &= \tilde{H}_m e_m + r_m. \end{aligned} \tag{3.15}$$

Now a comparison with (3.9) shows that the right hand side of (3.15) is just the last column of $\hat{H}_m$. ■

The above corollary shows that *one can get the matrix* $\hat{H}_m$ *from* $\tilde{H}_m$ *by just replacing the last column of* $\tilde{H}_m$ *with the column vector* $s_m = V_m^+ A v_m$. The vector s_m may be computed by the classical formula

$$s_m = \left(V_m^T V_m\right)^{-1} V_m^T A v_m. \tag{3.16}$$

This, however, makes explicit use of the matrix $B_m = V_m^T V_m$ and involves the solution of a full $m \times m$ linear system.

Another way of computing s_m is by solving the least squares problem

$$\underset{x}{\text{minimize}} \| V_m x - A v_m \|, \tag{3.17}$$

the solution of which is the desired vector s_m (see [18]).

Powerful algorithms are available for solving (3.17); e.g., see Björk and Elfving [2] and Paige [12]. Our experience reveals that the vector s_m need not be very accurate.

3.3. *Generalization*

Let $v_1, v_2, \ldots, v_j, \ldots, v_m$ be constructed by the recursion

$$A v_j - \sum_{i=1}^{j+1} h_{ij} v_i = 0, \qquad j = 1, 2, \ldots, m, \tag{3.18}$$

where the h_{ij}, $i = 1, \ldots, j+1$, are chosen so as to make the vector v_{j+1} satisfy certain requirements (see generalized Hessenberg processes in [20, pp. 377–395]). By (3.18) one obtains a sequence of vectors $v_1, v_2, \ldots, v_m$ and a Hessenberg matrix $\tilde{H}_m$ defined as in Sec. 3.2.1.

It is clear that under the assumption (3.4) the system $v_1, v_2, \ldots, v_m$ is still a basis of the Krylov subspace K_m. Here again one may ask *how to correct the matrix* $\tilde{H}_m$ *in order to get the matrix* $\hat{H}_m = (V_m^T V_m)^{-1} V_m^T A V_m$ *so as to achieve a generalized Lanczos process.*

It is easily seen that the second part of the Theorem 3.2 and the Corollary 3.3 remain valid here. Consequently, the answer to the above question is the same: $\hat{H}_m$ can be obtained from $\tilde{H}_m$ by replacing the last column of $\tilde{H}_m$ with $s_m = V_m^+ A v_m$.

Obviously the previous algorithms are particular cases of the generalization (3.18). The Arnoldi method is obtained by choosing the h_{ij}, $i = 1, 2, \ldots,$

$j+1$, such that v_{j+1} is orthogonal to $v_1, \ldots, v_m$ and has norm one. In incomplete orthogonalization, v_{j+1} has norm one and is orthogonal to the $p+1$ previous vectors.

Note that the simplest recurrence of the form (3.18) is the recurrence $Av_j - v_{j+1} = 0$, which constructs the Krylov sequence $v_j = A^{j-1}v_1$, $j = 1, \ldots, m$. However, one expects the computation of $s_m = V_m^+ Av_m$ to be harder in this case. In this sense this choice is unstable. Arnoldi's method and incomplete orthogonalization method aim to avoid instability by constructing an orthonormal or an almost orthonormal system $v_1, \ldots, v_m$ such that the computation of $s_m = V_m^+ Av_m$ becomes easier. Many other algorithms exist and remain to be studied in detail and compared.

4. NUMERICAL EXPERIMENTS

All the experiments described in this paper have been run on the IRIS-80, CII-HB computer of the Grenoble Computing Center. We used double precision (mantissa of 56 bits).

4.1. Numerical Experiments with Iterative Arnoldi Method

We tested Arnoldi's method described in Sec. 3.1 on a class of test matrices borrowed from [19]. These matrices represent the transition matrices of the Markov chains describing a random walk on an $(n+1)\times(n+1)$ triangular grid (Fig. 1). A transition may take place from the node (i, j) to one of the four adjacent nodes $(i+1, j)$, $(i, j+1)$, $(i-1, j)$, $(i, j-1)$. The probability of jumping from the node (i, j) to either of the nodes $(i+1, j)$ or $(i, j+1)$ is

$$pu(i, j) = 0.5 - \frac{i+j}{2n}$$

(This transition can occur only for $i+1 \leqslant n$ and $j+1 \leqslant n$.) The probability of jumping from the node (i, j) to either of the nodes $(i, j-1)$ or $(i-1, j)$ is

$$pd(i, j) = \frac{i+j}{2n},$$

this probability being doubled if either i or j is 0.

The nodes are numbered in the order $(0,0), (1,0), \ldots, (n,0), (0,1), (1,1), \ldots$. It is then known that the transpose of the matrix of transition probabilities admits 1 as eigenvalue. In this particular case -1 is an eigenvalue as well.

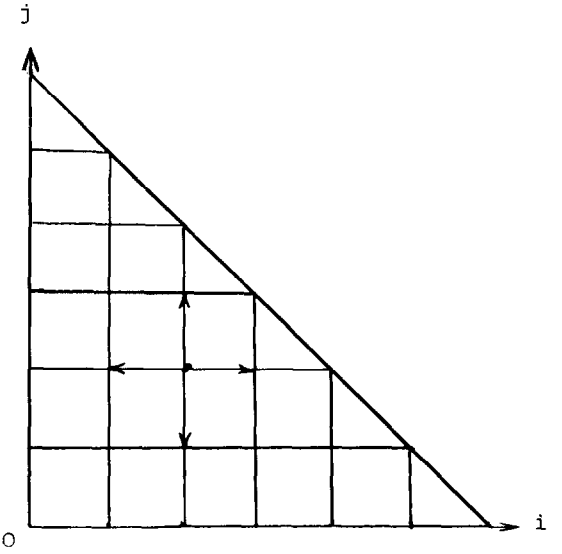


FIG. 1.

The matrix A^T need not be generated explicitly. Instead, one only needs a routine which computes $y=A^Tx$ for any given vector x. This routine, together with a more detailed description of this interesting class of sparse matrices, may be found in [19].

We are interested in the computation of the eigenvector corresponding to the eigenvalue unity, since it represents the steady state probabilities of the chain.

We begin by a comparison between the simultaneous iteration method and the iterative Arnoldi process.

In the first test we chose $n=13$, so that the dimension of A^T is $N=(n+1)(n+2)/2=105$. The simultaneous iteration method was run with different values of the number of columns M on which the iterations are performed. (See [19].) The process was stopped as soon as the Euclidean norm of the residual of the desired eigenvector was less than $\varepsilon=10^{-10}$. Table 1 gives the number IT of iterations that were necessary, as well as the number $M\times$IT of operations $x\rightarrow y=Ax$ and the run times. The first vector of the starting system U_0 was always $e=(1,1,\ldots,1)^T$. The other vectors were generated randomly.

The simultaneous iteration algorithm used here is the one described in [5].

We emphasize that an important drawback of the simultaneous iteration method, in the unsymmetric case, is that it does not, in general, allow

TABLE 1

M	IT	$M\times$IT	Run time (min)
4	196	784	0.37
6	137	822	0.44
8	116	928	0.62
10	98	980	0.62

efficient use of the Tchebycheff iteration as it usually does for symmetric matrices. Nevertheless this can be done when all the eigenvalues are real or when many extreme eigenvalues are real. For example, if the Tchebycheff iteration is used, one obtains when $M=4,6$ the results in Table 2 instead of those of Table 1.

In Table 3 we give the results obtained with the iterative Arnoldi method when m takes the values $m=5,10,15,20,25$. The stopping criterion was the same as above, and the starting vector was $v_1=e/\|e\|$. Note that when $m=20$ and $m=25$ the residual norms obtained were 1.7×10^{-13} and 5×10^{-13}, respectively. This indicates that for large values of m one should check the residual frequently.

Let us mention that if the matrix A were *a full matrix*, then the run times would be nearly proportional to the numbers $M\times$IT and $m\times$IT of operations $x\rightarrow y=Ax$ needed, so that a comparison between the tables reveals a sharp superiority of Arnoldi's method.

Many other tests were performed with several sparse matrices of various sizes (the largest was 450), issued from a Markov chain analyzer. These matrices are stochastic, and we sought, as above, the eigenvectors of their transposes, associated with the eigenvalue unity. An important simplification in this case is that the eigenvalue is known. The method is now extensively used for the solution of these problems, and it appears that it is much more efficient than the simultaneous iteration. Note, however, that a competitive method is provided by inverse iteration using either a sparse matrix code or a least squares method [2, 12] for solving the problems

$$\min_{x^{(s+1)}} \|(A-\lambda I)x^{(s+1)}-x^{(s)}\|.$$

TABLE 2

M	IT	$M\times$IT	Run time (min)
4	115	460	0.27
6	94	564	0.37

TABLE 3

m	IT	$m \times$ IT	Run time (min)
5	16	80	0.13
10	9	90	0.20
15	4	60	0.17
20	3	60	0.25
25	2	50	0.25

4.2. *Experiments with the Incomplete Orthogonalization Algorithm*

4.2.1. In the next experiment the incomplete orthogonalization algorithm was tested with a matrix A of the same type as in Sec. 4.1, with $n=16$, so that the dimension is $N=153$. It is instructive to plot for several values of m the residual norms $\|(A-\mu(\psi^{(m)})I)\psi^{(m)}\|/\|\psi^{(m)}\|$, where $\mu(\psi^{(m)})$ denotes the Rayleigh quotient $(A\psi^{(m)},\psi^{(m)})/\|\psi^{(m)}\|^2$. This was done with $p=19$, which means that the band-Hessenberg matrices $\tilde{H}_m$ had bandwidth 20. The starting vector v_1 was generated randomly.

Figure 2 shows the evolution of the above residual norms when m takes the values $5, 10, 15, \ldots, 90$. It shows at the same time the evolution of the residuals that were obtained with Arnoldi's method with $m=5, 10, 45$, using the same starting vector.

A few comments are in order. Let us first recall that the first 20 steps are nothing but Arnoldi steps. In a second stage, after the step $m=20$, actual incomplete orthogonalization steps are performed, and it is observed that the residuals are decreasing until $m=35$. After that, there is a third stage where the residual norms start oscillating. With the last values of m one notices, however, that the accuracy is not lost. Instead there is a slow improvement achieved in an oscillating way. At $m=35$ a second copy of the eigenvalue 1 appears. The curve obtained is a typical one.

The phenomenon observed in the third stage is, in a certain measure, similar to that occurring with the symmetric Lanczos method *without reorthogonalization*. Unfortunately, our theoretical results do not give an explanation to this phenomenon, for the system $(v_1, \ldots, v_n)$ is certainly far from orthonormal at that stage; only the second stage can be interpreted with the help of Proposition 3.1.

However, it should be added that in practice it is wiser, and in general more efficient, to halt at the end of the second stage ($m=35$ in the present example) and to restart if necessary, as in Algorithm 3.4. For example, the results obtained with Algorithm 3.4 on the same matrix are the following:

1st iterations: Halt at $m=40$, $\|\text{res}(\psi^{(35)})\| \simeq 7 \times 10^{-4}$, restart with $v_1 = \psi^{(35)}$;

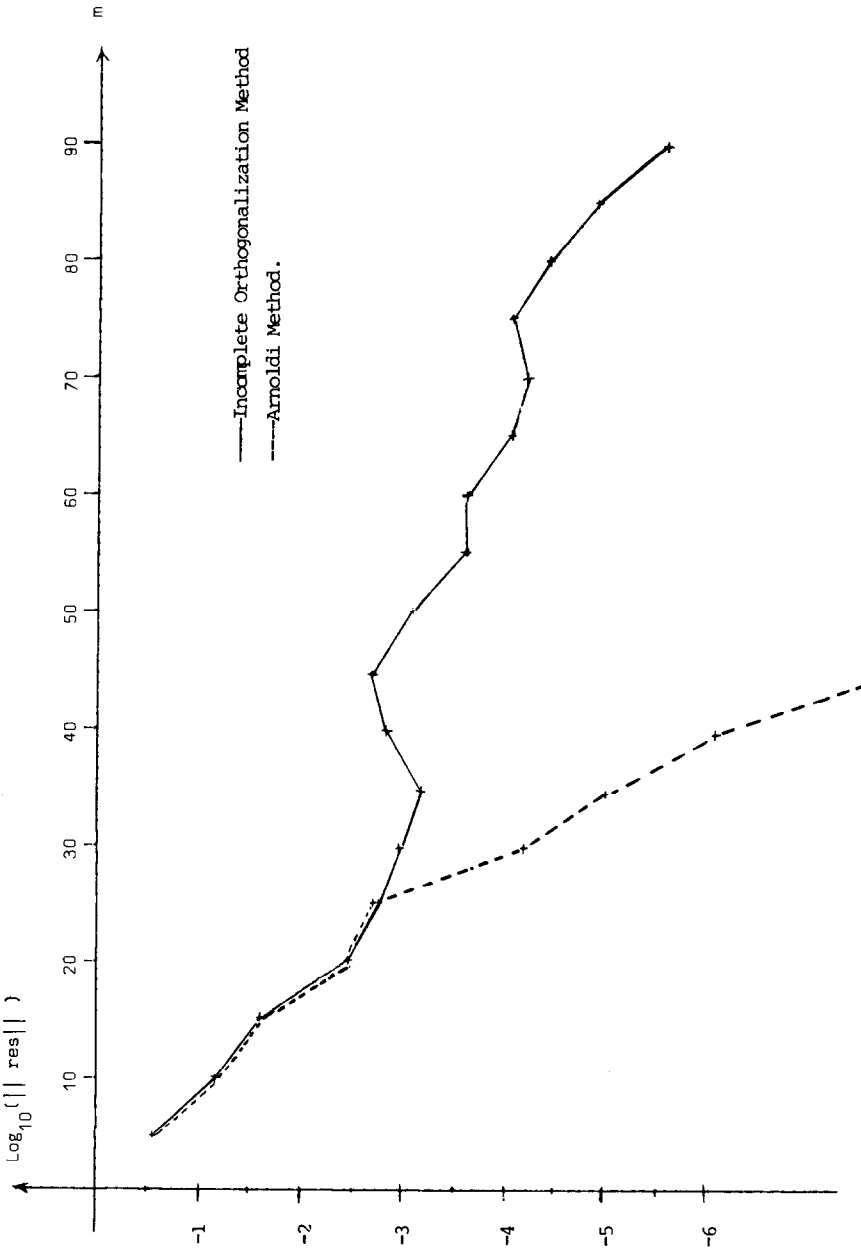

m
10
20
30
40
50
60
70
80
90
-1
-2
-3
-4
-5
-6
Log10(|| res||)
——Incomplete Orthogonalization Method
---Arnoldi Method.

FIG. 2.

2nd iteration: Halt at $m=30$, $\|\text{res}(\psi^{(25)})\| \simeq 5\times 10^{-7}$ (best vector obtained $\psi^{(25)}$).

Therefore, with $40+30=70$ matrix by vector operations, one obtains as residual norm 5×10^{-7}, whereas the last residual obtained without restarting was at $m=90$, only a little less than 2.6×10^{-6} at the price of 90 matrix by vector operations and a higher run time.

In order to show that the equality (3.14) provides a good estimate to the residual norms $\|(A-\mu(\psi^{(m)})I)\psi^{(m)}\|/\|\psi^{(m)}\|$ and to $\|(A-\mu^{(m)}I)\psi^{(m)}\|/\|\psi^{(m)}\|$, we compare these quantities in Table 4 with their corresponding estimates $h_{m+1,m}\|e_m^T z^{(m)}\|/\|z^{(m)}\|$ for $m=30,\ldots,90$. It is observed that, even for large values of m, the quantities $h_{m+1,m}|e_m^T z^{(m)}|/\|z^{(m)}\|$ remain rather good estimates for the residual norms.

4.2.2. An interesting application of the incomplete orthogonalization method takes place when the matrix A is *almost symmetric*, that is, when its skew-symmetric part $\frac{1}{2}(A-A^T)$ is small in comparison with its symmetric part $\frac{1}{2}(A+A^T)$. This is not uncommon when one discretizes eigenvalue problems involving non-self-adjoint partial differential operators.

Let us, for example, consider the following simple problem:

$$-\Delta u(x,y)+\frac{\partial}{\partial x}[a(x,y)u(x,y)]=u(x,y) \qquad \text{for } (x,y)\in\,]0,1[\,\times\,]0,1[\,,$$

$$u(x,y)=0 \qquad \text{on the boundary.}$$

Taking $a(x,y)=1$ and discretizing with centered differences yields the following matrix $A(n)$, where n is the chosen number of interior mesh points

TABLE 4

m	1st residual norm	2nd residual norm	Estimate
30	1.11×10^{-3}	1.16×10^{-3}	1.15×10^{-3}
40	1.19×10^{-3}	1.20×10^{-3}	1.14×10^{-3}
50	4.51×10^{-4}	4.72×10^{-4}	4.36×10^{-4}
60	2.47×10^{-4}	2.50×10^{-4}	2.16×10^{-4}
70	6.18×10^{-5}	6.24×10^{-5}	5.22×10^{-5}
80	3.45×10^{-5}	3.51×10^{-5}	2.94×10^{-5}
90	2.62×10^{-6}	2.62×10^{-6}	2.18×10^{-6}

on each side of the square:

$$A(n)=\begin{bmatrix} B_n & -I & & & & \mathbf{O} \\ -I & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & -I \\ \mathbf{O} & & & & -I & B_n \end{bmatrix}$$

with $$B_n=\begin{bmatrix} 4 & a & & & & \mathbf{O} \\ b & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & a \\ \mathbf{O} & & & & b & 4 \end{bmatrix}$$

and $a=-1+1/2(n+1)$, $b=-1-1/2(n+1)$. B_n and $A(n)$ have dimensions n and $N=n^2$, respectively.

Algorithm 3.4 was run on the 225×225 matrix $A(15)$ obtained by taking $n=15$ mesh points on each side. We took $p=19$ again, and the algorithm was stopped as soon as the *actual* residual norm was less than $\varepsilon=10^{-9}$. The starting vector was again a random vector.

The results obtained are as follows:

1st iteration: Halt at $m=50$, $\|\text{res}(\psi^{(45)})\|\simeq8\times10^{-5}$, restart with $v_1=\psi^{(45)}$;

2nd iteration: Stop at $m=40$, $\|\text{res}(\psi^{(40)})\|\simeq9\times10^{-10}$, eigenvalue obtained: $\lambda_1=7.92218308949660$.

4.2.3. In this final experiment, we compare the simultaneous iteration method SRRIT developed by G. W. Stewart [19] with Algorithm 3.4, on a matrix taken from the class of matrices described in Sec. 4.1. Taking $n=30$ provides a 496×496 matrix. We first reproduce in Table 6 the results obtained in [19] with the SRRIT routine for this matrix, when the number of columns M used takes the values 2, 4, 6, 8. The convergence criterion used there was that both eigenvectors associated with the eigenvalues $+1$ and -1 admit a residual norm less than 10^{-5}. According to the convergence theory, the two approximate eigenvectors converge with the same rate of convergence. So the results would have been nearly the same if the convergence criterion had dealt only with the eigenvector associated with the eigenvalue unity.

TABLE 6
RESULTS FROM [19]

M	IT	$M \times$ IT
2	1737	3474
4	523	2092
6	325	1950
8	188	1504

TABLE 7

P	IT	NOPE
14	3	95
19	4	110

Two runs were made on the same matrix as above, with the Algorithm 3.4 applied to the computation of the eigenvector associated with the eigenvalue unity. The stopping criterion was that the residual norm be smaller than 10^{-5}. In the first run, p was chosen equal to 14 (bandwidth 15). The trial vector was generated randomly. The results obtained are as follows:

1st iteration: Halt at $m=40$, $\|\text{res}(\psi^{(35)})\| \simeq 1.7 \times 10^{-2}$;
2nd iteration: Halt at $m=35$, $\|\text{res}(\psi^{(30)})\| \simeq 1.6 \times 10^{-4}$;
3rd iteration: Stop at $m=20$, $\|\text{res}(\psi^{(20)})\| \simeq 9.5 \times 10^{-6}$.

With $p=19$, these results became:

1st iteration: Halt at $m=30$, $\|\text{res}(\psi^{(25)})\| \simeq 2.3 \times 10^{-2}$,
2nd iteration: Halt at $m=30$, $\|\text{res}(\psi^{(25)})\| \simeq 3.2 \times 10^{-4}$,
3rd iteration: Halt at $m=30$, $\|\text{res}(\psi^{(25)})\| \simeq 2 \times 10^{-5}$,
4th iteration: Stop at $m=20$, $\|\text{res}(\psi^{(20)})\| \simeq 4.1 \times 10^{-6}$.

We sum up in Table 7 the number NOPE of operations $x \rightarrow Ax$ required in the two cases in order to permit a comparison with the results obtained with the SRRIT routine. This shows again that Algorithm 3.4 requires many less matrix by vector multiplications. However, we should note that it needs more memory. But it still remains advantageous, especially for full matrices (or not very sparse matrices).

The author wishes to thank Professor A. Ruhe for some very useful remarks and for suggesting the generalization of Sec. 3.3. The author is also indebted to one of the referees for improving the presentation of Corollary 3.3.

REFERENCES

1 W. E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.* 9:17–29 (1951).
2 A. Björck and T. Elfving, Accelerated projection methods for computing pseudo inverse solutions of systems of linear equations, *Nordisk Tidskr. Informationsbehandling (BIT)* 19:145–163 (1979).
3 G. W. Daniel, W. B. Gragg, L. Kaufmann, and G. W. Stewart, Reorthogonalization and stable algorithms for updating the Gram-Schmidt *QR* factorization, *Math. Comp.* 30:772–795 (1976).
4 G. H. Golub and R. Underwood, The block Lanczos method for computing eigenvalues, in *Mathematical Software III* (J. R. Rice, Ed.), Academic, New York, 1977, pp. 361–377.
5 A. Jennings and W. J. Stewart, Simultaneous iteration for partial eigensolution of real matrices, *J. Inst. Math. Appl.* 15:351–361 (1975).
6 S. Kaniel, Estimates for some computational techniques in linear algebra, *Math. Comp.* 20(95):369–378 (1966).
7 M. A. Krasnoselskii et al., *Approximate Solutions of Operator Equations*, Wolters-Nordhoof, Groningen, 1972.
8 C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Nat. Bur. Standards* 45(4):255–282 (1950).
9 J. G. Lewis, Algorithms for sparse matrix eigenvalue problems, Ph.D. Thesis, Stanford Univ. Report 77-595, 1977.
10 G. G. Lorentz, *Approximation of Functions*, Holt, Rinehart & Winston, New York, 1966.
11 C. C. Paige, The computation of eigenvalues and eigenvectors of very large sparse matrices, Ph.D. dissertation, Univ. of London, 1971.
12 C. C. Paige, Bidiagonalization of matrices and solution of linear equations, *SIAM J. Numer. Anal.* 11:197–209 (1974).
13 B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J., 1980.
14 A. Ruhe, Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse matrices, *Math. Comp.* 33(146):680–687 (1979).
15 Y. Saad, Calcul de valeurs propres de grandes matrices hermitiennes par des techniques de partitionnement, Thesis, Univ. of Grenoble, 1974.
16 Y. Saad, On the rates of convergence of the Lanczos and the block Lanczos methods, *SIAM J. Numer. Anal.*, to appear.
17 Y. Saad, Etude de la convergence du procédé d'Arnoldi pour le calcul d'éléments propres de grandes matrices non symetriques, in *Seminar of Numerical Analysis*, University of Grenoble, 1979, p. 321.
18 G. W. Stewart, *Introduction to Matrix Computations*, Academic, New York, 1973.
19 G. W. Stewart, SRRIT, a FORTRAN subroutine to calculate the dominant invariant subspace of a real matrix, *ACM Trans. Math. Software*, to appear.
20 J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, 1965.

Received 7 November 1979; revised 15 May 1980