UIUCDCS-R-80-1036

Master

THE LANCZOS BIORTHOGONALIZATION ALGORITHM AND OTHER OBLIQUE PROJECTION METHODS FOR SOLVING LARGE UNSYMMETRIC SYSTEMS

Y. Saad

December 1980

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

Supported in part by DOE grant DE-AC02-76ER02383.A003. Professor Saad is on leave from Laboratoire d'Informatique et de Mathematiques Appliquees de Grenoble.

DISCLAIMER

1. INTRODUCTION

In an earlier paper [13], some algorithms based on orthogonalization techniques have been proposed for solving large unsymmetric systems. Of particular interest is the incomplete orthogonalization method without correction, where at every step the solution is taken such as to make the new residual orthogonal to the p previous residuals where p is some small integer. As will be seen, this method can be regarded as an oblique projection method. By nonorthogonal or oblique projection method we mean a method which seeks a solution $\tilde{x}$ of $Ax = b$ by requiring that $\tilde{x}$ belongs to a certain subspace K (called the right space) and that the residual $b - A\tilde{x}$ be orthogonal to another subspace L (called the left subspace).

The best example of an oblique projection method for solving linear systems is provided by the method of Lanczos [7] which is a version of the well known conjugate gradient method in the symmetric case [3]. In that method the right space K is a Krylov subspace $K = \text{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$ where v_1 is a starting vector, while L is a Krylov subspace associated with A^H, $L = \text{span}\{w, A^H w_1, \ldots, (A^H)^{m-1} w_1\}$. The Lanczos algorithm has been neglected for a long time because of its instability as a method for tridiagonalizing a nonsymmetric matrix and computing its eigenvalues, although recently this fact has been reconsidered by Parlett and Taylor [11]. For solving linear systems, however, the method can be quite useful, especially when it is used in conjunction with a preconditioning technique. We should point out that the presence of A^H in the definition of L does not mean at all that the Lanczos method solves the normal equations $A^H Ax = A^H b$.

It is not the purpose of this paper to introduce a specific method effective for any large unsymmetric system, but rather to present and analyze a class ofmethods based upon oblique projection processes. Some of the algorithms presented are already known or can be trivially derived from known algorithms.

Section 2 sets the basic definitions and notations of the oblique projection methods and treats the important example of the Lanczos method. In section 3 other oblique projection methods, such as the Incomplete orthogonalization method and the Orthomin (p) method [14], are considered. The convergence properties of the algorithm are studied in section 4 and some numerical experiments are described in the last section comparing some of the methods treated.

2. OBLIQUE PROJECTION METHODS AND THE LANCZOS ALGORITHM

2.1. Oblique Projection Methods. Basic Theory and Notations

Let us consider the linear system

$$b - Ax = 0 \tag{2.1}$$

where A is an n x n nonsymmetric matrix. Let $V_m \equiv [v_1,\ldots, v_m]$ and $W_m \equiv [w_1,\ldots, w_m]$ be two systems of m linearly independent vectors in $\mathbb{C}^n$. The span of V_m (resp. W_m) will be denoted by K_m (resp. L_m) and will often be referred to as the right (resp. left) space. An oblique projection method onto K_m and orthogonally to L_m is any process that obtains an approximate solution x_m to problem (2.1), which belongs to K_m and which satisfies the relations:

$$b - Ax_m \perp w_j, \quad j = 1,\ldots, m \tag{2.2}$$

If a good guess x_0 at the solution is available, it is more appropriate to seek an approximate solution of the form

$$x = x_0 + z \tag{2.3}$$

where z belongs to K_m and where x is required to satisfy the same condition (2.2). In that case the new unknown z is the solution of the problem

$$r_0 - Az \perp w_j, \quad j = 1,\ldots, m \tag{2.4}$$

where r_0 is the initial residual $b - Ax_0$.

Note that the first formulation is a particular case of the second with $x_0 = 0$ and that the second formulation can be reduced to the first because it amounts to solving the problem

$$r_0 - Az = 0 \tag{2.5}$$

by the oblique projection method.

The second formulation is important for restarting the algorithm. The more general formulation (2.3), (2.4) will be often adopted. It will be assumed throughout that r_0 belongs to K_m. Another important assumption that we shall make is that

(H): no vector of L_m is orthogonal to either K_m or AK_m,

or equivalently that

$W_m^H V_m$ and $W_m^H A V_m$ are both inversible.

In that case the problem (2.5) has a solution z_m which can be obtained as

$$z_m = V_m y_m \tag{2.6}$$

where y_m is given by

$$y_m = (W_m^H A V_m)^{-1} W_m^H r_0 \tag{2.7}$$

In section 4 we will give an interpretation of the oblique projection method in terms of operators.

Indeed it will be seen that the above method replaces problem (2.1) by an approximate problem involving an operator of rank less than n.

2.2. The Lanczos Method of Biorthogonalization

2.2.1. The biorthonormalization process

A very attractive example of the oblique projection process described above is the method proposed by Lanczos in [7]. In that method, Lanczos suggested a simple way to generate biorthogonal systems W_m, V_m such that the matrix $W_m^H A V_m$ in (2.7) has a tridiagonal form. A simple version of his algorithm can be described as follows:

Algorithm 1

1. Choose v_1 and w_1 such that $(v_1, w_1) = 1$.

2. For $j = 1, 2, \ldots, m$ do

• $\hat{v}_{j+1} := Av_j - \alpha_j v_j - \beta_j v_{j-1}$ (2.10)

• $\hat{w}_{j+1} := A^H w_j - \alpha_j w_j - \delta_j w_{j-1}$ (2.11)

(when $i = 1$ take $\beta_1 v_0 := \delta_1 w_0 := 0$)

with $\alpha_j := (Av_j, w_j)$ (2.12)

• choose δ_{j+1} and β_{j+1} such that

$$\delta_{j+1}\, \beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1}) \tag{2.13}$$

• $$v_{j+1} := \hat{v}_{j+1}/\delta_{j+1} \tag{2.14}$$

$$w_{j+1} := \hat{w}_{j+1}/\beta_{j+1} \tag{2.15}$$

It can be shown easily that when the algorithm does not break down for a null inner product $(\hat{v}_{j+1}, \hat{w}_{j+1})$, then the vectors v_i and w_i satisfy the biorthonormality property:

$$(v_i, w_j) = \delta_{ij}, \quad i,j = 1,\ldots, m \tag{2.16}$$

Some interesting choices for δ_{j+1} and β_{j+1} in (2.13) are the following:

a. $\delta_{j+1} = |(\hat{v}_{j+1}, \hat{w}_{j+1})|^{1/2}$, $\beta_{j+1} = \delta_{j+1}\, \mathrm{sign}(\hat{v}_{j+1}, \hat{w}_{j+1})$ (2.17)

b. $\delta_{j+1} = \|\hat{v}_{j+1}\|$; $\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})/\delta_{j+1}$ (2.18)

This makes v_{j+1} of norm unity.

c. $\delta_{j+1} = \left|(\hat{v}_{j+1}, \hat{w}_{j+1})\|\hat{v}_{j+1}\|/\|\hat{w}_{j+1}\|\right|^{1/2}$ (2.18)

$$\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})/\delta_{j+1} \tag{2.19}$$

This last choice makes v_{j+1} and w_{j+1} having the same norm. Practically, the formulae (2.17) are to be preferred as they are more economical. Numerically, the purpose of (2.17), (2.19) is to attempt to balance the norms of the vectors v_{j+1} and w_{j+1}. It is, however, necessary to remark that the product $\|w_{j+1}\| \|v_{j+1}\|$ will not depend upon which of a, b, or c is applied because

$$\|v_{j+1}\| \|w_{j+1}\| = \frac{\|\hat{v}_{j+1}\| \|\hat{w}_{j+1}\|}{\delta_{j+1} \beta_{j+1}} = \frac{\|\hat{v}_{j+1}, \hat{w}_{j+1}\|}{|(\hat{v}_{j+1}, \hat{w}_{j+1})|}$$

$$\|v_{j+1}\| \|w_{j+1}\| = \frac{1}{\cos \theta(\hat{v}_{j+1}, \hat{w}_{j+1})}$$

where $\theta(x, y)$ denotes the acute angle between the vectors x and y. The angle $\theta(\hat{v}_{j+1}, \hat{w}_{j+1})$ is a function of A, v_1, w_1 and j only because, as will be seen later on, the vectors $\hat{v}_{j+1}$, $\hat{w}_{j+1}$ are uniquely determined by v_1, w_1 apart from a normalizing factor. This angle can be equal to $\pi/2$ causing the algorithm to stop. As shown in an example by Wilkinson [15, p. 390], this can occur even when A is well conditioned, and should not be incurred to any shortcoming in the matrix A. It is interesting to note that v_{j+1} and w_{j+1} can be written as $\hat{v}_{j+1} = p_j(A)v_1$, $\hat{w}_{j+1} = p_j(A^H)w_1$ where p_j denotes a polynomial of degree j, so that a sufficient condition for the feasibility of Lanczos algorithm is that

$$\forall p \ , \ d^0 p \leq m \ , \ (p(A)v_1, p(A^H)w_1) \neq 0 \qquad (2.21)$$

This generalizes the condition of the symmetric case which requires that the degree of the annihilating polynomial of v_1 must not exceed m.

2.2.2. Solution of the linear system by the Lanczos method.

The previous algorithm builds a system of biorthonormal vectors $\{v_i, w_i\}_{i=1,m}$ but does not provide explicitly an approximate solution for

(2.1). In [7] Lanczos has proposed an interesting way to build up such an approximation. His algorithm, which was published at about the same time as the conjugate gradient method of Hesleness and Stiefel [3], can be considered as a version of the C.G. algorithm. The algorithm proposed by Lanczos decouples each of the relations (2.10) and (2.11) in two other relations involving a new sequence of vectors (the conjugate directions) in which the solution is easily expressed. The approximate solution x_m provided by Lanczos' algorithm belongs to $\{x_0\} + K_m$ and its residual, which is proportional to the vector v_{m+1}, is orthonormal to the left space L_m. It is therefore equal to the solution that would be obtained by an oblique projection method using the subspaces $K_m = \text{span}(V_m)$ as a right space and $L_m = \text{span}(W_m)$ as a left space, where V_m, W_m is the biorthonormal system built by Algorithm 1.

In the following we show how the approximate solution can also be obtained directly as a combination of the vectors v_i. Note that another algorithm simpler and closer to the C.G. method will be given in the next subsection.

Suppose that Algorithm 1 is started with $v_1 = r_0/\|r_0\|$ and $w_1 = v_1$ (1) and let us consider the component vector y_m of z_m given by (2.6). From the algorithm and the biorthonormality property (2.16), it can be easily shown [16] that the m x m matrix $T_m = W_m^H A V_m$ has the tridiagonal form

(1) It is not necessary to start with $w_1 = v_1$ but it is somehow simplifying.

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \delta_2 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_m \\ & & \delta_m & \alpha_m \end{bmatrix} \tag{2.23}$$

(Notice that with the determination (a) of δ_{j+1}, T_m has the interesting additional property that

$$\beta_j = \pm \delta_j , \quad j = 2,\dots, m .)$$

Furthermore, the right hand side of the system (2.6) is equal to βe_1 where $\beta = \|r_0\|$; $e_1 = (1, 0,\dots, 0)^T$ because $W_m^H r_0 = \beta W_m^H v_1 = \beta e_1$.

The approximate solution x_m is therefore quite easy to obtain practically since we have

$$x_m = x_0 + V_m y_m = x_0 + \beta V_m T_m^{-1} e_1 \tag{2.24}$$

Computing the approximate solution by (2.20) and (2.21) requires the storage of the vectors $v_1, v_2,\dots, v_m$ but this does not constitute a major drawback because the formation of the solution by (2.21) takes place only when the convergence has occurred, and therefore the v_i's can be saved in secondary storage until then. This means, however, that we have to provide some means for determining whether the convergence is achieved, without explicitly using the approximate solution. Fortunately, this can be done quite easily thanks to the following formula, well known and tremendously useful in the symmetric case [10], [11] which expresses the residual norm in terms of y_m and $\|\hat{v}_{m+1}\|$

$$\|b - Ax_m\| = \|\hat{v}_{m+1}\| \, |e_m^T y_m| \tag{2.25}$$

Equality (2.25) enables us to compute the residual norm very economically and one can afford to make use of (2.25) periodically to monitor the convergence. Let us mention that it is not even necessary to actually compute y_m in order to estimate the residual norm. (See similar point in [11] and [14].)

Algorithm 2

1. Choose an initial vector x_0, compute

$$r_0 := b - Ax_0;$$

$$\beta := \|r_0\|$$

$$v_1 := w_1 := r_0/\beta$$

2. For $j = 1, 2, \ldots, s_{max}$ do

   a. compute v_{j+1}, w_{j+1}, α_j, β_{j+1}, δ_{j+1}. By formulae (2.10) to (2.15) of Algorithm 1.

   b. Periodically (e.g., when $[j/5]\cdot 5 = j$) update the estimate ρ_j of the residual norm.

   If $\rho_j \leq \varepsilon$ goto 3 else continue

3. Form the approximate solution

$$x_j = x_0 + \beta V_j y_j$$

2.3. Equivalent Versions and the Bi-Conjugate Gradient Method

We shall now give some equivalent versions of the basic Algorithms 1 and 2. We first introduce for a theoretical purpose a generalization of Algorithm 1. Then on the practical side a simpler version of Algorithm 2 will be studied.

2.3.1. A generalization of Algorithm 1

The following algorithm generalizes Algorithm 1 into a whole class of equivalent versions.

Algorithm 3

1. Choose v_1 and w_1 as in Algorithm 1.
2. For $j = 1, 2, \ldots, m$ do

$$\begin{cases} \hat{v}_{j+1} := Av_j - \alpha_j v_j - \beta_j v_{j-1} \\ \hat{w}_{j+1} := A^H w_j - \sum_{i=1}^{j} h_{ij} w_i \end{cases} \tag{2.26}$$

where $\alpha_j = (Av_j, w_j) - (Av_{j-1}, w_{j-1}) + h_{j-1,j-1}$

(when $j = 1$ take $\beta_1 v_0 = 0$ and $\alpha_1 = (Av_1, v_1)$)

and where the h_{ij}'s are arbitrary.

• Normalize v_{j+1} and w_{j+1} by using formulae (2.13), (2.14), (2.15), (2.17) of Algorithm 1.

Obviously, Algorithm 1 is a particular case of the above algorithm with $h_{j,j} = (Av_j, v_j)$; $h_{j-1,j} = \delta_j$ and $h_{ij} = 0$ for $i < j-1$.

For every particular choice of the parameters h_{ij} one obtains a set of vectors $V_m = [v_1, v_2, \ldots, v_m]$ and a tridiagonal matrix T_m defined by (2.23). What is surprising is that, theoretically, whatever the h_{ij}'s are, the above algorithm will always produce the same right vectors v_i, the same tridiagonal matrix T_m and therefore the same approximate solution x_m as is stated in the next proposition.

Proposition 1

Suppose that Algorithm 3 is feasible for a given pair of starting vectors v_1 and w_1. Then

(i) $$(v_j, w_k) = \delta_{jk}, \quad 1 \le k \le j \le m+1 \tag{2.27}$$

(ii) The system $V_m = [v_1, v_2, \ldots, v_m]$ and the matrix T_m produced by Algorithm 3 do not depend upon the choice of the parameters h_{ij} used in (2.26).

Proof

1. The proof is by induction. Suppose that $(v_j, w_k) = 0$; $k \leq j$ and let us show that $(v_{j+1}, w_k) = 0$; $k \leq j+1$. We consider three cases:

   a. <u>$k = j-1$</u>

$$(\hat{v}_{j+1}, w_{j-1}) = (Av_j, w_{j-1}) - \alpha_j(v_j, w_{j-1}) - \beta_j(v_{j-1}, w_{j-1})$$

$$= (Av_j, w_{j-1}) - \beta_j$$

$$= (v_j, A^H w_{j-1}) - \beta_j = (v_j, \beta_j w_j + \sum_{i=1}^{j-1} h_{ij-1} w_i) - \beta_j$$

$$= \beta_j + (v_j, \sum_{i=1}^{j-1} h_{ij-1} w_i) - \beta_j = 0$$

   b. <u>$k = j$</u>

$$(\hat{v}_{j+1}, w_j) = (Av_j, w_j) - \alpha_j(v_j, w_j) - \beta_j(v_{j-1}, w_j)$$

$$= (Av_j, v_j) - \alpha_j - \beta_j \; (v_{j-1}, \beta_j^{-1} [A^H w_{j-1} - \sum_{i=1}^{j-1} h_{ij-1} w_i])$$

$$= -h_{j-1,j-1} + h_{j-1,j-1} = 0 .$$

   c. <u>$k < j-1$</u>

$$(\hat{w}_{j+1}, w_k) = (Av_j, w_k) - \alpha_j(v_j, w_k) - \beta_j(v_{j-1}, w_k)$$

$$= (Av_j, w_k)$$

$$= (v_j, A^H w_k) = (v_j, \hat{w}_{k+1} + \sum_{i=1}^{k} h_{ik} w_i)$$

$= 0$ by the induction hypothesis.

2. a. In order to prove the second part of the proposition we shall need the following lemma:

<u>Lemma 1</u>

If the first m steps of Algorithm 1 can be realized then the $k \times k$ moment matrices M_k whose general terms are $m_{ij} = (A^{i+j-2} v_1, w_1)$, $i,j = 1,\ldots, k$, are regular for $k = 1, 2,\ldots, m$.

Proof of Lemma

Let us set $\bar{W}_k \equiv [w_1 A^H w, (A^H)^{k-1} w_1]$ and $\bar{V}_k = [v_1, Av_1, \ldots, A^{k-1} v_1]$. Since V_k and $\bar{V}_k$ are both bases of the same subspace K_m, there exists a regular k x k matrix S_k such that $\bar{V}_k = V_k \ S_k$. Similarly there exists a k x k regular matrix S_k' such that $\bar{W}_k = W_k S_k'$. But the matrix M_k is equal to

$$\bar{W}_k^H \bar{V}_k = S_k'^H W_k^H V_k S_k = S_k'^H S_k$$

which is regular. □

b. Let us now show that the v_i's are the same apart from a multiplicative factor. We can write $v_{j+1} = \eta_1 A^j v_1 + \eta_2 A^{j-1} v_1, \ldots, + \eta_0 v_1$. Consider the vector $\eta_1^{-1} v_{j+1}$ which we denote by $\tilde{v}_{j+1}$. The vector v_{j+1} can be written as

$$v_{j+1} = A^j v_1 - \sum_{i=0}^{j-1} \xi_i A^i v_1$$

and it satisfies the following equations

$$(\tilde{v}_{j+1}, (A^H)^k w_1) = 0, \qquad k = 0, 1, \ldots, j-1$$

because it is orthogonal to all the subspace spanned by $w_1, w_2, \ldots, w_j$ which is nothing but the left space $L_j = \text{Span}[w_1, (A^H)w_1, \ldots, (A^H)^{j-1} w_1]$. Hence the ξ_i's will be solutions of the linear system of equations:

$$\sum_{i=1}^{j} (A^{i-1} v_1, (A^H)^{k-1} w_1)\xi_i = (A^j v_1, (A^H)^{k-1} w_1), \quad k = 1, \ldots, j \quad (2.28)$$

Since the moment matrices M_k are assumed to be regular, then the solution of (2.28) is unique showing that $\tilde{v}_{j+1}$ does not depend upon the choice of the h_{ij}'s in Algorithm 3.

Next we must show that if we normalize the vector $\tilde{v}_{j+1}$ so that it makes with $\tilde{w}_{j+1}$ an inner product equal to unity, we obtain the same result with any choice of the h_{ij}'s. Let us consider the inner product $(\tilde{v}_{j+1}, \tilde{w}_{j+1})$:

$$(\tilde{v}_{j+1}, \tilde{w}_{j+1}) = (\tilde{v}_{j+1}, (A^H)^j w_1 - \nu_1 (A^H)^{j-1} w_1, \ldots, \nu_j w_1)$$

$$(\tilde{v}_{j+1}, \tilde{w}_{j+1}) = (\tilde{v}_{j+1}, (A^H)^j w_1)$$

because $\tilde{v}_{j+1}$ is orthogonal to $(A^H)^k w_1$, $k \leq j-1$. Therefore, the normalizing factor does not depend upon the parameters h_{ij}, which finally proves the fact that the v_i's are independent of the h_{ij}'s.

c. To complete the proof, there remains to show that T_m is independent on the parameters h_{ij}. From the algorithm we have

$$AV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T \tag{2.29}$$

where T_m is the tridiagonal matrix obtained from Algorithm 3 for a given choice of the parameters h_{ij}. On multiplying both sides of (2.31) by $\bar{W}_m^H$, where $\bar{W}_m = [w_1, A^H w_1, \ldots, (A^H)^{m-1} w_1]$, we get

$$\bar{W}_m^H AV_m = \bar{W}_m^H V_m T_m + \beta_{m+1} \bar{W}_m^H v_{m+1} e_m^T \tag{2.30}$$

Because of (2.27), v_{m+1} is orthogonal to all the subspace L_m and so $\bar{W}_m^H v_{m+1} = 0$. Furthermore by a proof similar to that of Lemma 1, it can be shown that the matrix $\bar{W}_m^H V_m$ is regular such that from (2.30) we have

$$T_m = (\bar{W}_m^H V_m)^{-1} \bar{W}_m^H AV_m$$

This and the fact that the v_i's are independent upon the h_{ij}'s shows that T_m is independent upon the h_{ij}'s. □

One might wonder whether it is possible to find among all the possible choices of the parameters h_{ij} one which makes the algorithm more efficient than Algorithm 1.

Although the result of Proposition 1 may seem powerful, it has little practical value as it turns out that the most stable and economical version is just the Algorithm 1. Its only practical interest lies in the problem of reorthogonalization. In effect the above results show that it is only important that v_{j+1} be orthogonal to the previous w_i, $i \leq j$. Therefore, if reorthogonalization is needed in Algorithm 1, one must apply it only on the set of vectors v_i; that is, one only needs to reorthogonalize the v_i's against the w_i's.

This, however, might be more important for eigenvalue problems than for the solution of linear equations.

2.3.2. The biconjugate gradient algorithm

The solution x_m provided by Algorithm 3 can be obtained by a conjugate gradient-like method which may be derived in the same way as the C.G. Method is derived from the Lanczos algorithm in the symmetric case [see, e.g., Paige and Saunders [9]].

Algorithm 4

1. Choose an initial guess x_0 of the solution
2. Compute $r_0 = b - Ax_0$ and take $p_0^* := r_0^* := p_0 := r_0$
3. For $k = 1,2,\ldots, m,\ldots$ compute

$$x_{k+1} := x_k + \alpha_k p_k$$

$$r_{k+1} := r_k - \alpha_k A p_k \tag{2.31}$$

$$r_{k+1}^* := r_k^* - \alpha_k A^H p_k^* \tag{2.32}$$

$$p_{k+1} := r_{k+1} + \beta_k p_k \tag{2.33}$$

$$p^*_{k+1} := r^*_{k+1} + \beta_k\, p^*_k \tag{2.34}$$

with

$$\alpha_k := (r_k,\ r^*_k)/(Ap_k,\ p^*_k) \tag{2.35}$$

$$\beta_k := (r_{k+1},\ r^*_{k+1})/(r_k,\ r^*_k) \tag{2.36}$$

The purpose of the above determination of α_k and β_k is to make the residual satisfy the relation $(r_{k+1},\ r^*_k) = 0$ and the direction p_k satisfy $(p_{k+1},\ A^H p^*_k) = 0$. In fact, the following proposition can be shown.

Proposition 2

The vectors r_k, r^*_k and p_k, p^*_k produced by Algorithm 4 are such that:

a. $(r_k,\ r^*_j) = 0$ for $j \neq k$ (biorthogonality property)

b. $(Ap_k,\ p^*_j) = 0$ for $j \neq k$ (biconjugacy property)

Proof

Clearly, because of the duality of the vectors r_k and r^*_k, p_k and p^*_k, it is sufficient to show that

$$(r_k,\ r^*_j) = (Ap_k,\ p^*_j) = 0\ , \qquad j < k \tag{2.37}$$

The proof is by induction. For $k = 1$ (2.37) is satisfied.

Suppose that (2.37) is satisfied and let us show that $(r_{k+1},\ r^*_j) = (Ap_{k+1},\ p^*_j) = 0$, $j < k+1$. For $j = k$ this is true by construction so we must show it for $j < k$.

a. $(r_{k+1},\ r^*_j) = (r_k - \alpha_k\, Ap_k,\ r^*_j)$

$$= (r_k,\ r^*_j) - \alpha_k (Ap_k,\ r^*_j)$$

Since $(r_k,\ r^*_j) = 0$ and using (2.34) we get

$$(r_{k+1}, r_j^*) = - \alpha_k(Ap_k, r_j) = - \alpha_k(Ap_k, p_j^* + \beta_{j-1} p_{j-1}^*)$$

$$= - \alpha_k(Ap_k, p_j^*) - \alpha_k \beta_{j-1}(Ap_k, p_{j-1}^*) = 0$$

b. $(Ap_{k+1}, p_j^*) = (p_{k+1}, A^H p_j^*)$

$$= (r_{k+1} + \beta_k p_k, A^H p_j^*) \text{ by (2.33)}$$

$$= (r_{k+1}, A^H p_j^*) \text{ by the induction assumption}$$

$$(Ap_{k+1}, p_j^*) = (r_{k+1}, A^H p_j^*) = \frac{1}{\alpha_j} (r_{k+1}, r_j^* - r_{j+1}^*)$$

$$= 0 \text{ since } j < k \ \square$$

Let us mention that all the relations that hold for the classical C.G. method will hold for the bi-conjugate gradient method if the vectors on the right parts of the inner products are replaced by the corresponding vectors p_k^*, r_k^*, etc.

It is important not to confuse this algorithm with the bidiagonalization method [8], where one essentially solves the normal equations. The bidiagonalization methods are projection methods on the subspaces $\text{Span}[r_0, (A^H A)r_0, \ldots, (A^H A)^{m-1} r_0]$ while here we are dealing with an oblique projection method on the subspace $K_m = \text{Span}[r_0, Ar_0, \ldots, A^{m-1} r_0]$.

That Algorithm 4 is theoretically equivalent to Algorithm 3 can be simply established as follows:

The solutions obtained by both algorithms satisfy $x_k = x_0 + z_k$ where z_k is such that

$$\begin{cases} z_k \in K_k = \text{Span}[r_0, Ar_0, \ldots, A^{k-1} r_0] \\ r_k = r_0 - Az_k \perp L_k = \text{Span}[r_0, A^H r_0, \ldots, (A^H)^{k-1} r_0] \end{cases}$$

Therefore, $z_k = \sum_{i=1}^{k} \eta_i A^{i-1} r_0$ for both methods and the η_i's are solutions of the linear system

$$(r_0 - A \sum_{i=1}^{k} \eta_i A^{i-1} r_0, (A^H)^{j-1} r_0) = 0 , \qquad j = 1, 2, \ldots, k \tag{2.38}$$

Assuming that the moment matrix M'_k, whose general elements m'_{ij} are $m'_{ij} = (A^{i+j-1} r_0, r_0)$, is regular[2], we conclude that the vectors z_k produced by both algorithms are the same because of the unicity of the solution of the system (2.38). □

On the practical side, Algorithm 4 presents the advantage of requiring less storage than Algorithm 2. It can be coded with six vectors of length N in core memory while the Lanczos algorithm needed five vectors in main memory and m vectors in secondary storage (when m is large, the latter may involve substantial input/output operation times).

Furthermore, the number of arithmetic operations required is slightly in favor of Algorithm 4 because there is no tridiagonal system to solve. Finally, because stable methods can be used to solve the m x m system, Algorithm 2 is, in general, more stable than Algorithm 4.

2.4. Feasibility of the Lanczos Algorithm and the Biconjugate Algorithm

Thus far we have not discussed under which conditions the Algorithms 2 and 4 are feasible. The moment matrices M_k and M'_k mentioned in the previous subsection play an important role as is seen in the next proposition.

---

(2) In Section 2.4 we shall see that this assumption is necessary for the feasibility of Algorithm 2.

Proposition 3

Let M_k and M'_k be the k x k moment matrices whose general terms are defined by $m_{ij} = (A^{i+j-2} v_1, v_1)$ and $m'_{ij} = (A^{i+j-1} v_1, v_1)$, respectively. Then the m-th approximate solutions x_m can be computed by Algorithm 2 if and only if

a. $$\det(M_k) \neq 0 \ , \ k = 1, 2, \ldots, m \tag{2.39}$$

b. $$\det(M'_m) \neq 0 \tag{2.40}$$

Proof

1. First we must show that if Algorithm 1 is feasible then (2.39), (2.40) are satisfied. That (2.39) is true has already been established in Lemma 1. Using the same matrices S_k and S'_k defined in that Lemma, it is also easy to prove (2.40).

2. Second we must show that under the assumptions (2.39) and (2.40), it is possible to compute x_m by Algorithm 2. Let us establish by induction that v_k, w_k can be computed for $k = 1, 2, \ldots, m$. This is trivially true when $k = 1$. Suppose that it is true for k-1 and consider the vectors $\hat{v}_k$ and $\hat{w}_k$. All that is needed in order to compute v_k, w_k is that $(\hat{v}_k, \hat{w}_k) \neq 0$. Suppose this is not true; that is, that

$$(\hat{v}_k, \hat{w}_k) = 0 \tag{2.41}$$

The vector $\hat{v}_k$ can be expressed as

$$\hat{v}_k = \sum_{i=1}^{k} \delta_i A^{i-1} v_1 \tag{2.42}$$

Since $\hat{v}_k$ is orthogonal to $w_1, w_2, \ldots, w_{k-1}$ (with $w_1 = v_1$), it is also orthogonal to $v_1, A^H v_1, \ldots, (A^H)^{k-2} v_1$ and (2.41) shows that it is also orthogonal to $A^{k-1} v_1$ because the vector $\hat{w}_k$ can be written as

$\hat{w}_k = \sum_{i=1}^{k} \delta_i'(A^H)^{i-1}v_1$ with $\delta_k' \neq 0$. Hence $\hat{v}_k$ is orthogonal to $v_1, (A^H)v_1, \ldots, (A^H)^{k-1}v_1$, which can be expressed as $(\sum_{i=1}^{k} \delta_i A^{i-1}v_1, (A^H)^{j-1}v_1) = 0, j = 1, \ldots, k$ or $M_k d = 0$ where $d = (\delta_1, \ldots, \delta_k)^H$ is a non null vector. This contradicts the fact that $\det(M_k) \neq 0$. Let us show that the solution x_m can be computed by the formula $x_m = x_0 + \beta V_m T_m^{-1} e_1$, that is that T_m is nonsingular. We can use the same argument as in Lemma 1. Let S_m, S_m' be two nonsingular k x k matrices such that $\bar{V}_m = V_m S_m$, $\bar{W}_m = W_m S_m'$ where $\bar{V}_m = [v_1, Av_1, \ldots, A^{m-1}v_1]$, $\bar{W}_m = [v_1, A^H v_1, \ldots, (A^H)^{m-1}v_1]$. We have $T_m = W_m^H AV_m = (S'^H)^{-1} \bar{W}_m^H A\bar{V}_m S_m^{-1} = (S'^H)^{-1} M_m' S_m^{-1}$ which in view of (2.40) gives $\det(T_m) \neq 0$ and completes the proof. □

An important remark which can be derived immediately from the proof is that the condition (2.39) ensures that $v_1, v_2, \ldots, v_m$ and $w_1, w_2, \ldots, w_m$ can be built while (2.40) ensures that the tridiagonal matrix T_m is nonsingular. It is therefore obvious that the proposition can be generalized as follows:

The approximations $x_{k_1}, x_{k_2}, \ldots, x_{k_m}$ can be built by Algorithm 2 iff $\det(M_j) \neq 0$, $j = 1, 2, \ldots, k_m$ and $\det(M_{k_j}') \neq 0$, $j = 1, 2, \ldots, m$. For the biconjugate gradient method we have the following analogue of the above result.

<u>Proposition 4</u>

The first m steps of Algorithm 4 can be performed iff:

a. $$\det(M_k) \neq 0 \;, \quad k = 1, 2, \ldots, m \tag{2.43}$$

b. $$\det(M_k') \neq 0 \;, \quad k = 1, 2, \ldots, m \tag{2.44}$$

Proof.

1. Necessary condition. If m steps of Algorithm 4 are realizable, then for $1 \leq k \leq m$ we have four systems $R_k \equiv [r_1, \ldots, r_k]$, $R_k^* \equiv [r_1^*, \ldots, r_k^*]$, $P_k \equiv [p_1, \ldots, p_k]$, $P_k^* \equiv [p_1^*, \ldots, p_k^*]$ such that

$$\begin{cases} (r_i, r_j^*) = 0 \,, & i \leq k,\ j \leq k, \quad i \neq j \\ (r_k, r_k^*) \neq 0 \end{cases}$$

$$\begin{cases} (p_i, A^H p_j^*) = 0 \,, & i \leq k,\ j \leq k, \quad i \neq j \\ (p_k, A^H p_k^*) \neq 0 \end{cases}$$

This means that $(P_k^*)^H R_k$ is diagonal and nonsingular while $(P_k^*)^H A P_k$ is triangular and nonsingular. But R_k, R_k^*, P_n, P_n^* are four different bases of the same subspace K_k and so from the above we can show in a way similar to the first part of the proof of Proposition 3 that M_k and M_k' are nonsingular.

2. Sufficient condition. Suppose that (2.43) and (2.44) are satisfied and let us show by induction that $x_1, x_2, \ldots, x_m$ can be obtained from Algorithm 4 or equivalently that $(r_k, r_k^*) \neq 0$; $(p_k, A^H p_k^*) \neq 0$, $k = 1, \ldots, m$. This is true for $k = 1$. Let us assume that it is true for k-1: $(r_{k-1}, r_{k-1}^*) \neq 0$; $(p_{k-1}, A^H p_{k-1}^*) \neq 0$. That the first relation holds for k can be shown in the same way as in part 2 of Proposition 3. (Note that r_k and r_k^* are proportional to v_k and w_k, respectively.) The second relation to show is $(p_k, A^H p_k^*) \neq 0$. Suppose the contrary is true, then using the notations of the first part of this proof we get that the matrix $(P_k^*)^H A P_k$ is singular and, using again the fact that $\bar{V}_k = P_k S_k^{(3)}$, $\bar{W}_k = P_k^* S_k^{(4)}$ where $S_k^{(3)}$ and $S_k^{(4)}$ are both k x k and nonsingular, we get that M_k' is singular which contradicts (2.40) and completes the proof. □

As a consequence of the remark following Proposition 3, if we assume that only the condition (2.39) is satisfied and that $\det(M'_m) \neq 0$, then Algorithm 4 may break down before the m-th step while Algorithm 2 does not because the tridiagonal systems $T_j y_j = \beta e_j$, need not be solved for $j \neq m$. Only the solution y_m of the last system $T_m y_m = \beta e_1$ is actually necessary to obtain x_m. From this point of view Algorithm 2 is superior to Algorithm 4.

3. OTHER OBLIQUE PROJECTION METHODS

The purpose of this section is to attempt to derive some other oblique projection methods. It will first be seen that the Incomplete orthogonalization method without correction presented in [14] is nothing but an oblique projection method. Then, based upon an analogue of Proposition 1, we shall describe a particular class of the oblique projection methods for the solution of linear systems.

3.1. The Incomplete Orthogonalization Method

Among the methods proposed in [14], the Incomplete orthogonalization method without correction was found to be the most attractive. A simple description of the method is the following:

Algorithm 5

a. Choose two integers p and m and construct a system of vectors $v_1, v_2, \ldots, v_m$ by

1. $v_1 := r_o/(\beta := \|r_0\|)$ with $r_0 = b - Ax_0$

2. For $j = 1, \ldots, m$

$$\hat{v}_{j+1} := Av_j - \sum_{i=i_0}^{j} h_{ij} v_i \tag{3.1}$$

where $i_0 = \max\{1, j-p+1\}$

$$h_{ij} = (Av_j, v_i) \tag{3.2}$$

$$v_{j+1} := \hat{v}_{j+1}/(h_{j+1,j} := \|\hat{v}_{j+1}\|) \tag{3.3}$$

b. Take as approximate solution

$$x_m = x_0 + \beta V_m H_m^{-1} e_1 \tag{3.4}$$

where $V_m \equiv [v_1, \ldots, v_m]$ and where H_m is the (band) Hessenberg matrix whose nonzero elements are the h_{ij} computed by (3.2) and (3.3). Note that $\hat{v}_{j+1}$ is obtained by orthogonalizing Av_j against the previous p vectors.

The above method was founded upon the fact that if we compare the solution (3.4) with that provided by Arnoldi's method (an orthogonal projection method upon the Krylov subspace K_m), we would find that the difference between them is *negligible* provided that the system $[v_1, \ldots, v_m]$ remains not too far from orthogonal [14], a fact which is often observed (see comments following Proposition 6 below).

We now would like to give an interpretation of the method in terms of oblique projection methods. More precisely, we shall exhibit a system of left vectors $w_1, \ldots, w_m$ such that the I.O.M. algorithm will amount to performing an (oblique) projection method onto $K_m = \text{span}[v_1, Av_1, \ldots, A^{m-1}v_1]$ and orthogonally to $L_m = \text{span}[W_m]$.

Consider the system of vectors w_i obtained from $v_1, v_2, \ldots, v_m, v_{m+1}$ as follows

$$w_i = v_i - (v_i, v_{m+1})v_{m+1}, \quad i = 1, 2, \ldots, m \tag{3.5}$$

Each of the vectors w_i is orthogonal to v_{m+1}, so that if we set $W_m \equiv [w_1, \ldots, w_m]$ we get

$$W_m^H v_{m+1} = 0 \tag{3.6}$$

we can then state the next result.

<u>Proposition 5</u>

Let $V_m = [v_1, \ldots, v_m]$ be the system obtained from Algorithm 5 and let $W_m = [w_1, \ldots, w_m]$ be defined by (3.5). Then the approximate solution provided by the Incomplete orthogonalization method is equal to that obtained by the oblique projection method using $K_m = \text{span}[V_m]$ as the right space and $L_m = \text{span}[W_m]$ as the left space.

<u>Proof</u>

From (3.1) it can be shown that

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

which gives, on multiplying both sides by W_m^H, $W_m^H AV_m = W_m^H V_m H_m + h_{m+1,m} W_m^H v_{m+1} e_m^T$. Using (3.6) and assuming that $W_m^H V_m$ is nonsingular we get

$$(W_m^H V_m)^{-1} W_m^H AV_m = H_m \tag{3.7}$$

From (2.6) it is seen that the solution x_m' obtained by the oblique projection method, using as left space $\text{span}[W_m]$ and right space $\text{span}[V_m]$ is given by

$$x_m' = x_0 + V_m (W_m^H V_m)^{-1} W_m^H AV_m W_m^H r_0$$

and since $r_0 = v_1 = \beta V_m e_1$ we have

$$x_m' = x_0 + \beta V_m (W_m^H AV_m)^{-1} W_m^H V_m e_1$$

which in view of (3.7) gives

$$x_m' = x_0 + \beta V_m H_m^{-1} e_1 \ .$$

But this is just the solution (3.4) provided by the I.O.M. method and the proof is complete. □

Notice that in the case when v_{m+1} is orthonormal to $v_1, v_2, \ldots, v_m$, the w_i's coincide with the v_i's which means that Q_m becomes an orthonormal projection. In that case the method would give theoretically the same result as Arnoldi's method [14]. It was precisely the aim of the Incomplete orthogonalization method <u>with correction</u>, described in [14], to attempt to orthogonalize v_{m+1} against all previous vectors $v_1, \dots, v_m$ by finding scalars h_{im}, $i = 1, \dots, m$ such that

$$Av_m - \sum_{i=1}^{m} h_{im} v_i \perp v_i, \quad i = 1, \dots, m.$$

This, however, is difficult to achieve in practice because the previous v_i's, $i = 1, \dots, m$ do no longer form an orthonormal system, and therefore the coefficients h_{im} can be found only by solving a least square problem.

This raises the interesting question to know under which condition on A, the I.O.M. method reduces to an orthogonal projection method. The answer is given by the next proposition.

<u>Proposition 6</u>

Suppose that there exists a polynomial q_p of degree $p - 1$ such that

$$A^H = q_p(A) \tag{3.8}$$

Then the vectors v_i computed from the I.O.M. algorithm (Algorithm 5) are orthonormal and therefore the Incomplete orthogonalization method realizes an orthogonal projection method onto the Krylov subspace K_m (Arnoldi's method).

<u>Proof</u>

We must show that $(v_{j+1}, v_i) = 0$, $1 \le j$ for $j = 1, 2, \dots, m$. The proof is by induction. Suppose $v_j \perp v_i$, $i = 1, 2, \dots, j-1$ (which is clearly true for $j = 1$ by construction) and let us consider $(\hat{v}_{j+1}, v_k)$ where $k \le j$ $(\hat{v}_{j+1}, v_k) = 0$ if $j \le k \le i_0 = \max(1, j-p+1)$ because by construction v_{j+1} is orthonormal to the previous p vectors $v_{j-p+1}, v_{j-p+2}, \ldots, v_j$ (see [13]). For $k \le j-p$ we have $(\hat{v}_{j+1}, v_k) = (Av_j, v_k - \sum_{i=i_0}^{j} h_{ij}(v_i, v_k)$. By the induction assumption $(v_i, v_k) = 0$, $i = i_0, j$, hence

$$(\hat{v}_{j+1}, v_k) = (v_j, A^H v_k) = (v_j, q_p(A)v_k) \tag{3.9}$$

But v_k belongs to K_k and therefore there exists a polynomial s of degree not exceeding k-1 such that $v_k = s(A)v_1$ which implies that the vector $q_p(A)v_k$ in (3.9) can be written as $q_p(A)v_k = t(A)v_1$ where t is the product of the polynomials s and q_p and has degree not exceeding $(p-1) + k-1$. Since $k \le j-p$ the degree of t does not exceed $j-2$ and therefore $q_p(A)v_k$ belongs to K_{j-1} which means that $(v_j, A^H v_k)$ in (3.9) is zero and the proof is complete. □

Any Hermitian or skew-Hermitian matrix will satisfy the conditions of the theorem with $p = 2$. Also, any matrix of the form $A = \alpha I + \beta S$ where S is skew-symmetric will satisfy the condition (3.8) with $p = 2$ as for example when

$$A = \begin{bmatrix} \alpha & \beta & & \\ -\beta & \ddots & \ddots & \\ & \ddots & \ddots & \beta \\ & & -\beta & \alpha \end{bmatrix}$$

In general, however, an arbitrary matrix A does not satisfy (3.8). Nevertheless, a relation of the form (3.8) is often nearly satisfied with a small p which explains why one often gets nearly orthogonal systems by the I.O.M. algorithm with p as small as 5 or 10.

To conclude with the I.O.M. let us mention that it is also possible to write an equivalent algorithm in a form similar to Algorithm 4 which does not require to save the vectors v_i in secondary memory (see [14]).

3.2. A Particular Class of Oblique Projection Methods for Linear Systems

3.2.1. Generalized Hessenberg processes

The results of the previous section can be extended to yield a whole class of oblique projection methods. Suppose that we start with $v_1 = r_0/\beta$ where $\beta = \|r_0\|$ and that we build a sequence of vectors $v_1, v_2, \ldots, v_m$, by the general formula

$$h_{j+1,j}\, v_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij}\, v_i \tag{3.10}$$

where the h_{ij}, $i = 1, 2, \ldots, j+1$, are determined such as to make the vector v_{j+1} satisfy certain conditions such as, for example, $(v_{j+1}, v_i) = \delta_{ij}$, $i = 1, \ldots, j+1$ (which gives the method of Arnoldi). Such processes, called the Generalized Hessenberg processes by Wilkinson [15], have in common the equation

$$AV_m = V_m H_m + h_{m+1,m}\, v_{m+1}\, e_m^T \tag{3.11}$$

where V_m and H_m are defined as before. Let us then consider the solution x_m obtained by applying the formula (3.4) in the same way as in the Incomplete orthogonalization method. Such an approximate solution will have a residual vector proportional to the last vector v_{m+1} obtained from (3.10) because from relation (3.11) we can show that

$$b - Ax_m = h_{m+1,m}\, e_m^T\, y_m\, v_{m+1} \tag{3.12}$$

Thus by requiring that the vector v_{j+1} satisfy certain conditions, when building the sequence $\{v_i\}$ by (3.10), we implicitly require that the same conditions be satisfied for the j-th residual vector.

Let us now show that the generalized Hessenberg processes belong to the class of oblique projection methods.

Suppose that $W_m = [w_1, w_2, \ldots, w_m]$ is any system of vectors such that

$$W_m^H\, v_{m+1} = 0 \tag{3.13}$$

$$\det(W_m^H V_m) \neq 0 \tag{3.14}$$

(Note that W_m is not unique). Then it can be shown by the same proof as that of Proposition 6 that x_m is exactly the solution that would be obtained with an oblique projection method using as right space the space $K_m = \text{span}[v_1, \ldots, A^{m-1}v_1]$ and as left space the space $G_m = \text{span}[W_m]$.

Clearly, the methods of Lanczos and the I.O.M. are particular cases. In the Lanczos method the h_{ij}'s, $i = 1, \ldots, j$, are chosen such that v_{j+1} is orthogonal to all the left space $L_m = \text{span}[w_1, A^H w_1, \ldots, (A^H)^{m-1} w_1]$, and it turns out that this can be realized by the elegant Algorithm 1 of Lanczos in which $h_{ij} = 0$ for $i < j-1$. In the I.O.M., the coefficients h_{ij} are taken such as to make v_{j+1} orthogonal to the p previous v_i's. Some other applications are described next.

3.2.2. <u>ORTHOMIN and the conjugate residual method</u>

Suppose that the coefficients h_{ij} in (3.10) are determined such as to make at each step j the vector v_{j+1} orthogonal to the vectors $Av_1, Av_2, \ldots, Av_j$. The h_{ij}'s can then be obtained by solving the j x j system

$$\sum_{i=1}^{j} (v_i, Av_k)h_{ij} = (Av_j, Av_k) , \qquad k = 1, 2,\ldots, j \tag{3.15}$$

Notice that $(v_i, Av_k) = 0$ for $k < i$, such that the above system is triangular. This is nothing but an oblique projection method with $K_m = \text{span}[v_1, Av_1,\ldots, A^{m-1}v_1]$ as right space and $L_m = AK_m$ as left space. It can be shown that the solution obtained by this method minimizes $\|b - Ax\|$ over the affine subspace $x_0 + K_m$.

This method was first presented in a simplified version by Vinsome [15]. It was also analyzed by Axelsson [1] and by Eisenstat, Elman, and Schultz [2], who give some results on the convergence theory. The simplified version, called ORTHOMIN by Vinsome, produces directly x_m as a sequence of the form $x_m = x_{m-1} + \alpha_m p_m$ where p_m is the direction of search.

Algorithm 6 (ORTHOMIN or Generalized Conjugate Residual Method)

1. Start x_0 initial vector. Compute $r_0 = b - Ax_0$, take $p_0 = r_0$.
2. Iterate

$$x_{k+1} = x_k + \alpha_k p_k \qquad \alpha_k = \frac{(r_k, Ap_k)}{(Ap_k, Ap_k)}$$

$$r_{k+1} = r_k - \alpha_k Ap_k$$

$$p_{k+1} = r_{k+1} - \sum_{i=1}^{k} \beta_{ik} p_i , \qquad \beta_{ik} = \frac{(Ar_{k+1}, Ap_i)}{(Ap_i, Ap_i)}$$

The coefficient α_k is chosen such that the residual r_{k+1} is orthogonal to Ar_k, while the β_{ik}'s are such that Ap_{k+1} is orthogonal to Ap_i, $i \leq k$. Under these conditions it can be shown that $(r_{k+1}, Ar_i) = 0$, $i \leq k$ (which is equivalent to the condition $(v_{k+1}, Av_i) = 0$, $i \leq k$ of the previous version) and hence the residuals are "conjugate." (Notice that since A is nonsymmetric, the conjugacy holds only in one side because it is not true that $(v_{k+1}, Av_i) = 0$ for $i \geq k$. It would be more appropriate to say that the residulas are "semi-conjugate.")

The amount of work and the storage required in Algorithm 6 is prohibitive and unless the algorithm is used iteratively with periodic restarting, it would be of little practical value. Vinsome has then suggested to perform an Incomplete orthogonalization for generating the p_k's. The idea is similar to that of I.O.M. and consists of truncating the sum defining p_{k+1} in Algorithm 6 as follows

$$p_{k+1} = r_{k+1} - \sum_{i=k-p+1}^{k} \beta_{ik} \, p_i$$

Obviously this is still an oblique projection method. If we compare Algorithm 6 with the I.O.M. we will find that while the amount of work is similar, the storage is in favor of the latter. However, ORTHOMIN is certainly easier to study theoretically because of the minimum residual property. Numerical tests will compare the two methods in the last section.

3.2.3. The modified Hessenberg process

In the method of Hessenberg for reducing a matrix to Hessenberg form [16], the h_{ij}'s in (3.10) are chosen such that v_{j+1} has zero components in its j first positions. The h_{ij}'s are found by solving a $j \times j$ triangular system. Therefore, $K_m = \text{span}[v_1, Av_2, \ldots, A^{m-1}v_1]$ and $L_m = \text{span}[e_1, e_2, \ldots, e_m]$. A natural simplification similar to the ideas used in I.O.M. and ORTHOMIN(p) is to save the previous p vectors only, to replace (3.10) by

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

and to determine h_{ij} such as to make p+1 components of the vector v_{j+1} equal to zero. An important question is how to choose the positions in which the zeros must appear. Some experiments have motivated us to prefer the following choice: eliminate the components having the largest modulus among the vectors v_j, v_{j-1},..., v_{j-p+1}

Many other possibilities exist and it may be possible that the above choice is not the best. The modified Hessenberg process described here has the advantage not to require any inner product.

4. CONVERGENCE PROPERTIES

In this section the difficult problem of the convergence of the approximate solution x_m toward the exact solution x^* will be considered. It is important to clarify what is meant by convergence. First, if we assume that the w_i's, $i = 1, 2, \ldots, n$, are linearly independent, then the approximate solution x_m will converge to x^* in at most n steps. This is because if we write the condition (2.2) in the form $W_n^H(b - Ax_n) = 0$, we obtain on multiplying by $(W_n^H)^{-1}$, $x_n = A^{-1}b = x^*$. Therefore, the sequence x_m is a finite sequence and by studying the convergence of x_m we shall mean deriving some properties which will ensure that x_m may be a good approximation to x^* even for m much smaller than the dimension n of the problem. The analysis proposed here is essentially the same as that given in our previous paper [14] and we shall only emphasize on those results that present nontrivial differences.

Let P_m be the orthogonal projector onto the subspace K_m, and Q_m the (oblique) projector onto K_m orthogonally to L_m. We shall study the convergence in terms of the distance $\epsilon_m = \|(I - P_m)z^*\|$ where z^* is the exact solution of the problem (2.5), and where $\|\cdot\|$ denotes the Euclidean norm. This distance between z^* and the subspace K_m has been fully studied in [14] and some bounds for it have been established, showing that in general ϵ_m is a quantity which decreases rapidly to zero.

We shall need an interpretation of the oblique projection method in terms of operator equations. Let us define the operator[3] $A_m = Q_m A P_m$, and make the assumption (H) of §2.1. We then have

[3]Note that here A denotes at the same time a matrix and its associated linear operator.

Lemma 2

The problem

$$\begin{cases} z \in K_m & (4.1) \\ r_0 - A_m z = 0 & (4.2) \end{cases}$$

has as its unique solution the approximate solution z_m provided by the oblique projection method using K_m as right space and L_m as left space.

Proof

It is sufficient to translate problems (4.1), (4.2) into matricial notations. Since $z \in K_m$, it can be written as

$$z = V_m y \tag{4.3}$$

Furthermore, r_0 and z belong to K_m and therefore $P_m z = z$ and $Q_m r_0 = r_0$. The matricial representation of Q_m in the canonical basis is $V_m(W_m^H V_m)^{-1} W_m^H$ and so (4.1), (4.2) give

$$V_m(W_m^H V_m)^{-1} W_m^H r_0 - V_m(W_m^H V_m)^{-1} W_m^H A V_m y = 0$$

which yields

$$y = (W_m^H A V_m)^{-1} W_m^H r_0 \tag{4.4}$$

This means that the problem (4.1), (4.2) has a unique solution and a comparison between (4.3), (4.4) on the one hand and (2.6), (2.7) on the other hand show that the solution is just that obtained by the projection method. □

We shall refer to problem (4.1), (4.2) as the approximate problem. What the lemma shows is that the projection method described in §2.1 amounts to replacing the problem (2.1) by the approximate problem. Our next task is naturally to relate the solutions of the two problems. A simple way to relate z^* to z_m is to give a bound for either the residual of z_m for problem (2.5) or for the residual of z^* for problem (4.2). The latter case is considered in the next proposition.

Proposition 5

Let $\gamma_m = \|Q_m A(I - P_m)\|$ then

$$\|r_0 - A_m z^*\| \leq \gamma_m \varepsilon_m \tag{4.5}$$

Proof

We have

$$\|r_0 - A_m x^*\| = \|Q_m(r_0 - AP_m z^*)\| = \|Q_m(Az^* - AP_m z^*)\|$$

$$= \|Q_m A(I - P_m)z^*\| = \|Q_m A(I - P_m)(I - P_m)z^*\|$$

$$\leq \gamma_m \varepsilon_m \; . \; \square$$

Corollary 1

Let γ_m be defined as above and let $\kappa_m = \|(A_m|_{K_m})^{-1}\|$. Then

$$\|z_m - z^*\| \leq (1 + \gamma_m^2 \kappa_m^2)^{1/2} \varepsilon_m \tag{4.6}$$

Proof

See analogue result in [14].

The number of $\gamma_m \kappa_m$ plays the role of a condition number for the approximate problem. The corollary therefore means that the error made in approximating z^* by z_m (which is the same as the error $x^* - x_m$) will be of the same order as ε_m provided that the approximate problem is not too badly conditioned.

We believe that there is no simple way of bounding either K_m or γ_m because Q_m is an oblique projector. Thus γ_m can be bounded as $\gamma_m \leq \|Q_m\| \|A\|$ where $\|Q_m\|$ is not known. (In the orthogonal projection case we have $\|Q_m\| = 1$.)

Note that we do not have at our disposal optimality properties such as the very helpful ones involved in the conjugate gradient method. An interesting bound for the residual of z_m for problem (2.5) can also be established by adapting a result shown by Vainikko (see [5]) for orthogonal projection methods.

Proposition 6

Assume that $\tau_m = \min_{\substack{x \in AK_m \\ \|x\|=1}} \|Q_m x\|$ is nonzero and let $c_m = \|Q_m\|$, and $\epsilon'_m = \min_{z \in K_m} \|r_0 - Az\|$, then

$$\epsilon'_m \leq \|r_0 - Az_m\| \leq (1 + c_m/\tau_m)\epsilon'_m \qquad (4.7)$$

Proof

Consider the restriction $\tilde{Q}_m$ of Q_m to the subspace AK_m. If $\tau_m \neq 0$ then $\tilde{Q}_m$ is a bijection from AK_m to $Q_m AK_m$. Furthermore from equation (4.2) we get

$$r_0 = Q_m Az_m$$

and since Az_m belongs to AK_m we have

$$Az_m = \tilde{Q}_m^{-1} r_0 = \tilde{Q}_m^{-1} Q_m r_0$$

Hence

$$r_0 - Az_m = (I - \tilde{Q}_m^{-1} Q_m) r_0 \qquad (4.8)$$

Let now x be any vector of AK_m. Then $(I - \tilde{Q}_m^{-1} Q_m)x = 0$ and hence (4.8) can also be written as

$$r_0 - Az_m = (I - \tilde{Q}_m^{-1} Q_m)(r_0 - x) \quad \forall \in AK_m .$$

Thus

$$\|r_0 - Az_m\| \leq \|I - \tilde{Q}_m^{-1} Q_m\| \|r_0 - x\| \quad \forall \in AK_m$$

and

$$\|r_0 - Az_m\| \leq (1 + \|\tilde{Q}_m^{-1}\| \|Q_m\|) \quad {}_{\in K_m} \|r_0 - Ax\|$$

Since $\|\tilde{Q}_m^{-1}\| = \tau_m$ this establishes the second part of (4.7). The first part is obvious. □

It is important to remark that in the case where K_m is the Krylov subspace, then

$$\epsilon_m' = \min_{\begin{cases} p \in P_{m-1} \\ p(0)=1 \end{cases}} \|p(A)r_0\| \tag{4.9}$$

where P_{m-1} denotes the space of polynomials of degree not exceeding m - 1. This quantity is very similar to the quantity ϵ_m and the bounds for ϵ_m' are of the same nature as those for ϵ_m.

It may seem at first that inequality (4.7) is more powerful than the previous inequality (4.6) because the condition number of the approximate problem does not appear in it. This is not true, however, because the number τ_m^{-1} can be shown to be equal to

$$\|A(A_{m\ K_m})^{-1}\|$$

The inverse of $A_{m\ K}$ is therefore implicitly involved in the constant τ_m^{-1} and we have $\tau_m^{-1} \leq \|A\| \kappa_m$ where κ_m is defined in Corollary 1.

5. NUMERICAL EXPERIMENTS

The numerical experiments described in this section have been run on the CDC CYBER 175 at the University of Illinois at Urbana-Champaign. The single precision has been used throughout (mantissa of 48 bits).

5.1. Comparison of I.O.M. and Lanczos

We shall first compare the Incomplete orthogonalization method (see 3.1) with the Lanczos method (Algorithm 2) on the following example.

$$A = \begin{bmatrix} B & -I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & B \end{bmatrix} \text{ with } B = \begin{bmatrix} 4 & a & & \\ b & \ddots & \ddots & \\ & \ddots & \ddots & a \\ & & b & 4 \end{bmatrix} \qquad (5.1)$$

and $a = -1 + \delta$, $b = -1 - \delta$.

B is of dimension 20 and A has dimension $N = 100$. These matrices represent the 5-point discretization of the operator $-\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} + \gamma \frac{\partial}{\partial x}$ on a rectangular region.

The right hand side b is taken to be $b = Ae$ where $e = (1, 1, \ldots, 1)^T$, such that the solution of the system is just e. The parameter δ is taken equal to 0.5 in this first example. The next figure compares the convergence of the I.O.M. algorithm with two values of the parameter p, $p = 2$ (upper curve) and $p = 4$ (middle curve), with Algorithm 2 (lower curve). It is seen that the convergence is faster with the Lanczos algorithm. However, each step of the Lanczos algorithm requires two matrix by vector multiplications while I.O.M. requires only one. It should be mentioned that the I.O.M. algorithm applied here is the Algorithm 5 of [13] and that it includes a restarting strategy. (Two restarts have been necessary for $p = 2$ while no restart has been needed when $p = 4$.)

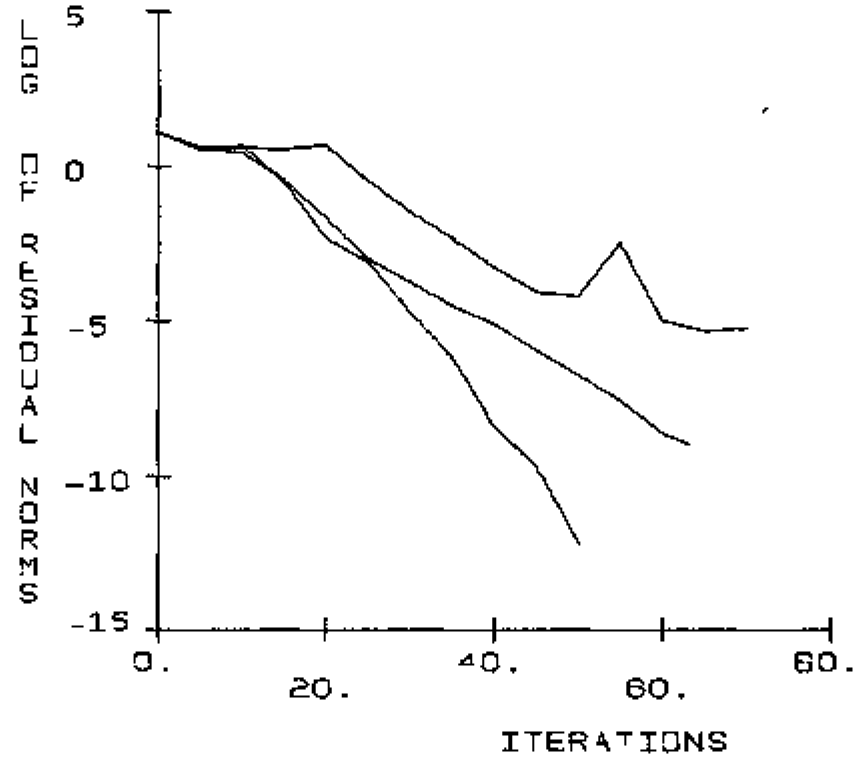

Figure 1. Upper curve: IOM(2), middle: IOM(4), lower: Lanczos

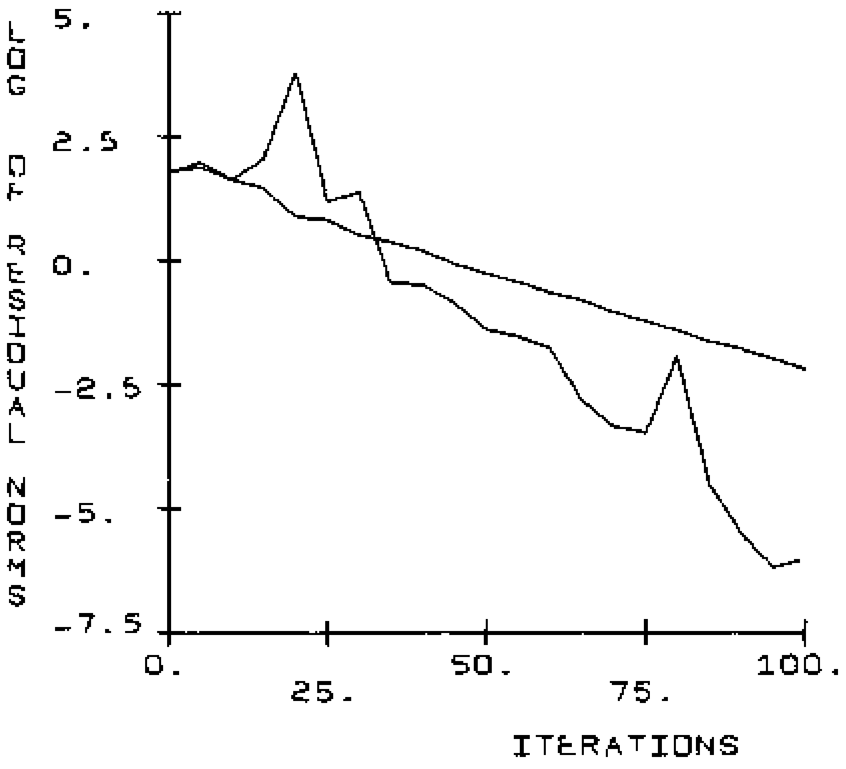

Figure 2. Lanczos and IOM(4)

Figure 2 shows the same example with $\delta = 10$ treated with Algorithm 2 and I.O.M.(4). Notice the peaks presented by the Lanczos method.

The Lanczos algorithm often behaves in a way similar to that of Figure 2, especially in situations where there are large imaginary eigenvalues. It is to remark that these peaks do not seriously affect the overall convergence. When the residual norm increases rapidly after a certain step, it decreases even more rapidly in the following steps.

5.2. I.O.M., Lanczos and ORTHOMIN

It was mentioned by Paige and Saunders [9] and by other authors that, in the symmetric case, the conjugate residual method (or minimum residual method) and the conjugate gradient method often exhibit a similar convergence behavior. As the next experiment will show, we can make a similar remark for the I.O.M. and the ORTHOMIN-G.C.R. methods. Let A be defined as in section 5.1, with the same right hand side and the same δ. Figure 3 shows the convergence behaviors of I.O.M.(4) (upper bound), ORTHOMIN(4) (middle curve), and the Lanczos method (lower curve) for this example.

Recall that the ORTHOMIN(P) requires twice as much memory as I.O.M.(p) and that in each step of ORTHOMIN(P) we have to perform two matrix by vector multiplications against only one such operation for I.O.M.(p). This means that for this example, I.O.M. is superior if we do not take into account the fact that for the I.O.M. there are some additional I.O. operations (necessary for the preservation of the v_i's until convergence). Algorithm 3 converges much faster than I.O.M.(p) and ORTHOMIN( ) but uses two matrix by vector multiplications. However, it has the advantage not to require from the user to supply the parameter p that is needed both in I.O.M. and ORTHOMIN.

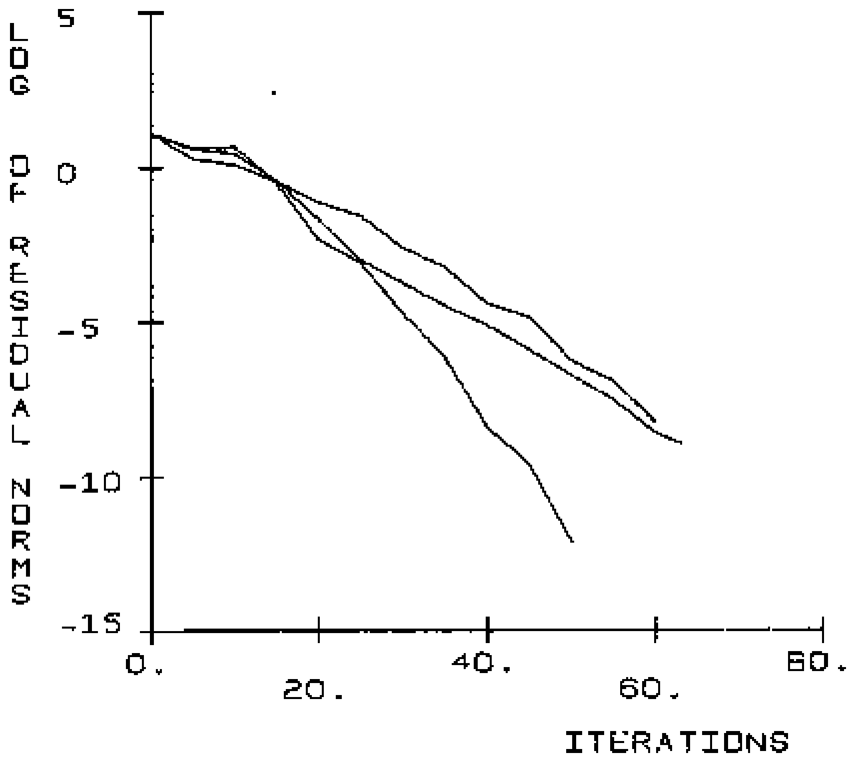

Figure 3. IOM(4), ORTHOMIN(4) and Lanczos

5.3. Complex Eigenvalues and the Lanczos Method

The purpose of the following example is to show how the behavior of the Lanczos method can vary when the shape of spectrum changes. Let B be the 100 x 100 block-diagonal matrix with 2 x 2 blocks c_k defined by

$$c_k = \begin{bmatrix} a_k & -b_k \\ b_k & a_k \end{bmatrix}, \quad k = 1, 2, \ldots, 50$$

with $a_k = k$, $b_k = \delta a_k$ where δ is a parameter. The eigenvalues of B are $\lambda_k^{\pm} = k(1 \pm i\delta)$ where $i = \sqrt{-1}$, $k = 1, 2, \ldots, 50$. When δ is small the eigenvalues are almost real positive and B is almost symmetric. The theory indicates that in that case a fast convergence can be expected because the distance $\|(I - \Pi_m)x^*\|$ decreases rapidly to zero [13]. When δ increases, the spectrum spreads out in $\mathbb{C}$ and in that case the theory does not guarantee a good rate of convergence. Figure 4 shows the behavior of the Lanczos method for the following values of δ: $\delta = 0.1$ (curve a), $\delta = 0.4$ (curve b), $\delta = 0.7$ (curve c), $\delta = 1$ (curve d), $\delta = 10$ (curve e). The graphs obtained confirm the theoretical indications. We emphasize here that in the case where a preconditioning is applied, the eigenvalues of the resulting matrix are closer to 1 than those of the original matrix such that the situations of poor convergence, similar to the case $\delta = 10$ here, can be avoided.

5.4. Generalized Hessenberg Process

Finally we will describe an experiment with a generalized Hessenberg process belonging to the class of methods outlined in section 3. Let us again take the example given in section 5.1 and consider the generalized Hessenberg process which builds a sequence of vectors v_j as follows

$$h_{j+1,j} v_{j+1} = Av_j - \sum_{i=j-p+1}^{j} h_{ij} v_i \tag{5.2}$$

where $h_{j+1,j}$ is a normalizing factor for v_{j+1} and where the h_{ij}, $i \neq j+1$ are chosen such as to make p components of v_{j+1} equal to zero. An important question is to determine which components of v_{j+1} should be zero for more efficiency. Several tests have been made, yielding various rates of convergence, depending on the strategies adopted. It was found that for this example a good strategy consists in eliminating

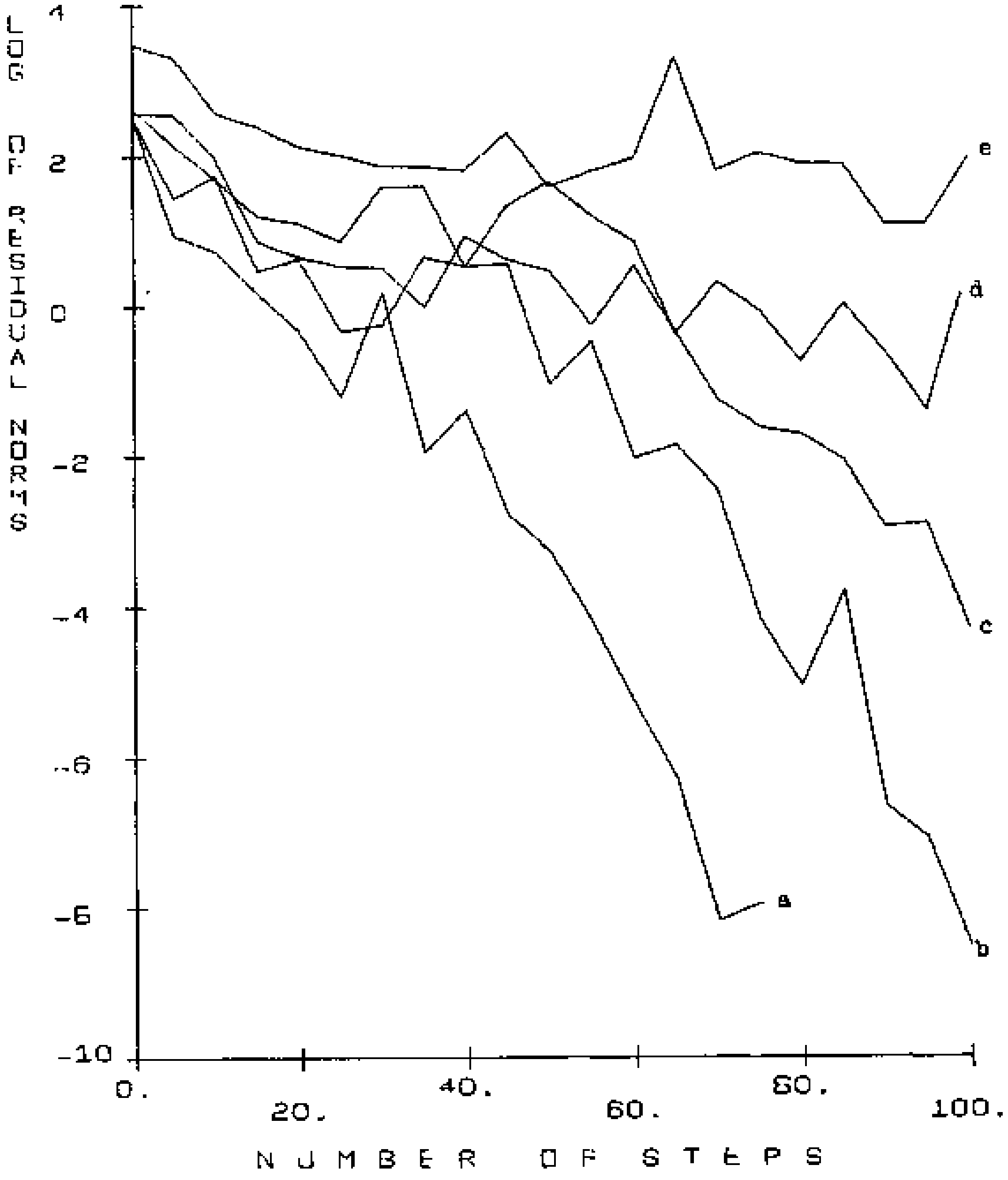

Figure 4. Behavior of Lanczos algorithm on different shapes of spectrum

in (5.2) the components j, j-1,..., j-p+1. A comparison of this strategy when p=2, with I.O.M.(2) and ORTHOMIN(2) is shown in Figure 5. It can be seen that the convergence of the generalized Hessenberg method compares well with that of I.O.M.(2) or ORTHOMIN(2), and the fact that there are no innerproducts involved for building the v_i's makes the Generalized Hessenberg method quite attractive. More general and more powerful strategies remain, however, to be investigated. Another strategy, that has appeared effective, isto eliminate the components in v_{j+1} corresponding to the large components in the previous v_i's.

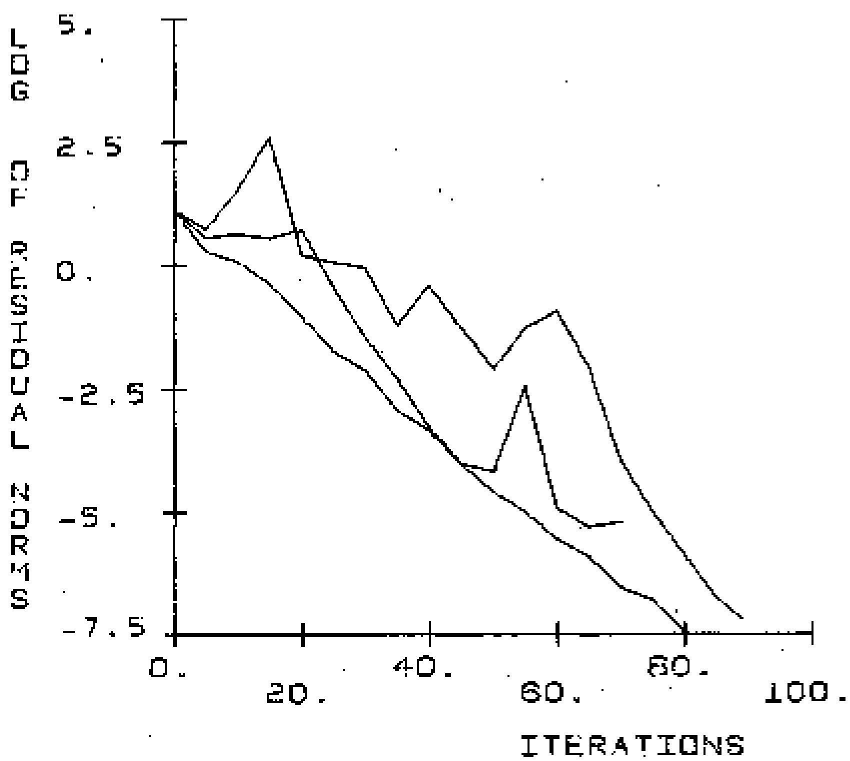

Figure 5. A generalized Hessenberg method (upper curve), IOM(2) (middle curve), and ORTHOMIN(2) (lower curve).

REFERENCES

[1] Axelsson, O., Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations, Lin. Algebra and its Applications 29, 1980, 1-16.

[2] Eisenstat, S.C., H. Elman, M.H. Schultz, A. Sherman, Solving approximations to the convection diffusion equation, Proceedings Fifth SPE Symposium on Reservoir Simulation, Denver, Colorado, 1979, 127-132.

[3] Hestenes, M. and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. NBS 49, 1952, 409-436.

[4] Householder, A.S., The Theory of Matrices in Numerical Analysis. Blaisdell Publishing Company: New York, 1964.

[5] Krasnoselskii, et. al., Approximate Solution of Operator Equations. Walters Nordproof: Groningen, 1972.

[6] Lanczos, C., An iteration method for the solution of the eigenvalue problem of linear differential and integral operator, J. Res. NBS 45, 1950, 255-282.

[7] Lanczos, C., Solution of systems of linear equations by minimized iterations, J. Res. NBS 49, 1952, 33-53.

[8] Paige, C.G., Bidiagonalization of matrices and solution of linear equations, SIAM J. Numer. Anal. 11, 1974, 197-209.

[9] Paige, C.G. and M.A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12, 1975, 617-629.

[10] Parlett, B.N., The Symmetric Eigenvalue Problem. Prentice-Hall, Inc.: Englewood Cliffs, New Jersey, 1980.

[11] Parlett, B.N., A new look at the Lanczos algorithm for solving symmetric systems of linear equations, Lin. Alg. Appl. 29, 1980, 323-346.

[12] Parlett, B.N. and D. Taylor, Lanczos for large nonsymmetric matrices, Sparse Matrix Meeting, March 1980, Seattle, Washington.

[13] Saad, Y., Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. To appear, Lin. Alg. Appl.

[14] Saad, Y., The Krylov subspace methods for solving large unsymmetric systems. To appear Math. Comp.

[15] Vinsome, P.K.W., Orthomin, an iterative method for solving sparse sets of simulteneous linear equations," Proceedings of 4th Symposium on Numerical Simulation of Reservoir Simulation of the SPE, Los Angeles, 1976, 149-160.

[16] Wilkinson, J.E., The Algebraic Eigenvalue Problem. Clarendon Press: Oxford, 1965.

DOE Form IR-427
(1/79)

U.S. DEPARTMENT OF ENERGY

UNIVERSITY-TYPE CONTRACTOR AND GRANTEE RECOMMENDATIONS FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

See Instructions on Reverse Side

1. DOE Report No.
C00-2383-0075

2. Contract No.
DE-AC02-76ER02383.A003

3. Title
THE LANCZOS BIORTHOGONALIZATION ALGORITHM AND OTHER OBLIQUE PROJECTION METHODS FOR SOLVING LARGE UNSYMMETRIC SYSTEMS

4. Type of Document ("X" one)
[X] a. Scientific and technical report
[ ] b. Conference paper:
Title of conference ______
Date of conference ______
Exact location of conference ______
Sponsoring organization ______
[ ] c. Other (Specify Thesis, Translations, etc.)

5. *Recommended Announcement and Distribution* ("X" one)
[X] a. DOE's normal announcement and distribution procedures may be followed.
[ ] b. Make available only within DOE and to DOE contractors and other U.S. Government agencies and their contractors.

6. Reason for Recommended Restrictions

7. Patent Information
Does this information product disclose any new equipment, process or material? [ ] Yes [ ] No

Has an invention disclosure been submitted to DOE covering any aspect of this information product? If so, identify the DOE (or other) disclosure number and to whom the disclosure was submitted. [ ] Yes [ ] No

Are there any patent related objections to the release of this information product? If so, state these objections.

8. Submitted by

Name and Position (Please print or type)
Charles W. Gear, Professor and Principal Investigator

Organization
University of Illinois at Urbana-Champaign

Signature Charles W. Gear

Date December 1980

FOR DOE USE ONLY

9. Patent Clearance ("x" one)
[ ] a. DOE patent clearance has been granted by responsible DOE patent group.
[ ] b. Report has been sent to responsible DOE patent group for clearance.
[ ] c. Patent clearance not required.

BIBLIOGRAPHIC DATA SHEET	1. Report No. R-80-1036	2.	3. Recipient's Accession No.
4. Title and Subtitle THE LANCZOS BIORTHOGONALIZATION ALGORITHM AND OTHER OBLIQUE PROJECTION METHODS FOR SOLVING LARGE UNSYMMETRIC SYSTEMS			5. Report Date December 1980
			6.
7. Author(s) Y. Saad			8. Performing Organization Rept. No. R-80-1036
9. Performing Organization Name and Address Department of Computer Science University of Illinois U-C Urbana, IL 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. DE-AC02-76ER02383.A003
12. Sponsoring Organization Name and Address U.S. Department of Energy Chicago Operations Office 9800 S. Cass Avenue Argonne, IL 60439			13. Type of Report & Period Covered technical
			14.

15. Supplementary Notes

16. Abstracts

Many powerful methods for solving systems of equations can be regarded as projection methods. Most of the projection methods known for solving linear systems are orthogonal projection methods but little attention has been given to the class of nonorthogonal (or oblique) projection methods, which is particularly attractive for large nonsymmetric systems. The purpose of this paper is to present some methods in the general setting of oblique projection methods and to give some theoretical results. Some experiments comparing the various algorithms are reported.

17. Key Words and Document Analysis. 17a. Descriptors

iterative
nonsymmetric
projection methods
linear systems

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 48
unlimited	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71