UIUCDCS-R-81-1047

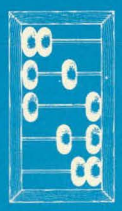
UILU-ENG 81 1701
~~CSO-2383-0077~~ STET
DOE/ER/02383

KRYLOV SUBSPACE METHODS FOR SOLVING LARGE UNSYMMETRIC LINEAR SYSTEMS

by

Y. Saad

January 1981



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

DISCLAIMER

UIUCDCS-R-81-1047

KRYLOV SUBSPACE METHODS FOR SOLVING LARGE UNSYMMETRIC LINEAR SYSTEMS

by

Y. Saad

January 1981

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

Supported in part by the U. S. Department of Energy grant DE-AC02-76ER02383.A003.
Y. Saad was on leave from the Laboratoire d'Informatique et de Mathematiques Appliquees de Grenoble.

1. INTRODUCTION

Few efficient iterative methods have been developed for treating large nonsymmetric linear systems. Some methods amount to solving the normal equations $A^H Ax = A^H b$ associated with the system $Ax = b$ or with some other system derived by a preconditioning technique.

This, unfortunately, is sensitive to the conditioning of $A^H A$ which is in general much worse than that of A. Techniques using Tchebycheff iteration [12] do not suffer from this drawback but require the computation of some eigenvalues of A.

A powerful method for solving symmetric linear systems is provided by the conjugate gradient algorithm. This method achieves a projection process onto the Krylov subspace $K_m = \text{Span}\ (r_0, Ar_0, \ldots, A^{m-1}r_0)$ where r_0 is the initial residual vector. Although the process should theoretically produce the exact solution in at most N steps, it is well known that a satisfactory accuracy is often achieved for values of m for less than N [15]. Concus and Golub [5] have proposed a generalization of the conjugate gradient method which is based upon the splitting of A into its symmetric and skew-symmetric parts.

The purpose of the present paper is to generalize the conjugate gradient method regarded as a projection process onto the Krylov subspace K_m. We shall say of a method realizing such a process that it belongs to the class of Krylov subspace methods. It will be seen that these methods can be efficient for solving large nonsymmetric systems.

The next section describes the Krylov subspace methods from a theoretical point of view. In Section 3 some algorithms are proposed. They are essentially the extensions of the Arnoldi-like methods for solving large eigenvalue problems described in [18]. Section 4 deals with the

convergence of the Krylov subspace methods. Finally, some numerical experiments are described in Section 5.

2. THE KRYLOV SUBSPACE METHODS -- THEORETICAL ASPECTS

2.1. General projection process -- notations

Consider the linear system

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

where A is a (complex or real) N x N matrix and let $V_m = [v_1, \ldots, v_m]$ be a system of m linearly independent vectors in $\mathbb{C}^N$. The projection process onto the subspace $K_m = \text{Span}\,(v_1, \ldots, v_m)$ seeks an approximation $x^{(m)}$ to the solution of (2.1) by requiring that

$$\begin{cases} x^{(m)} \in K_m \\ Ax^{(m)} - b \perp v_j, \quad j = 1, 2, \ldots, m \end{cases} \tag{2.2}$$

Writing $x^{(m)} = V_m \cdot y^{(m)}$, it is immediate that $y^{(m)}$ must satisfy the m x m linear system

$$V_m^H A V_m \cdot y^{(m)} - V_m^H b = 0 \tag{2.3}$$

where V_m^H denotes the transpose of the conjugate of V_m: $V_m^H = \bar{V}_m^T$. Let π_m denote the orthogonal projector onto the subspace K_m. Then another formulation of (2.2) is the following

$$\begin{cases} x^{(m)} \in K_m \\ \pi_m (Ax^{(m)} - b) = 0 \end{cases} \tag{2.4}$$

It will be assumed for simplicity that $b \in K_m$. We shall denote by A_m the restriction of $\pi_m A$ to K_m, so that $x^{(m)}$ is the solution in K_m of the equation

$$A_m x - b = 0 \tag{2.5}$$

(Note that $b \in K_m$ so that $\pi_m b = b$.)

The problem (2.1) is therefore replaced by the m-dimensional problem (2.5). In order to study the convergence properties of this process, one may express the error in terms of the distance between the exact solution x^* and the subspace K_m, that is in terms of $\|(I - \pi_m)x^*\|$. See [8].

Note here that when A is Hermitian definite positive, the convergence is more easily studied by using the fact that the approximate solution $x^{(m)}$ minimizes the error function $E(x) = (x - x^*)^H A(x - x^*)$ over all elements x in K_m. Unfortunately, this property does not extend to the nonsymmetric case so it becomes necessary to make a different approach. Suppose that the exact solution x^* is close to K_m, in that $\pi_m x^*$ is close to x^*, then it is possible to show that $x^{(m)}$ is close to πx_m^* (hence to x^*) by showing that the residual of $\pi_m x^*$ for the problem (2.5) is small. More precisely,

Proposition 2.1

Let $\gamma_m = \|\pi_m A(I - \pi_m)\|$. Then the residual of $\pi_m x^*$ for problem (2.5) satisfies

$$\|b - A_m \pi_m x^*\| \leq \gamma_m \|(I - \pi_m)x^*\| \tag{2.6}$$

Proof

$$\begin{aligned} b - A_m \pi_m x^* &= b - \pi_m A \pi_m x^* \\ &= b - \pi_m A[x^* - (I - \pi_m)x^*] \\ &= \pi_m A(I - \pi_m)x^* \end{aligned}$$

Observing that $(I - \pi_m)$ is a projector we can write

$$\|b - A_m \pi_m x^*\| = \|\pi_m A(I - \pi_m)(I - \pi_m)x^*\|$$

$$\leq \gamma_m \|(I - \pi_m)x^*\|$$

which completes the proof. □

As a consequence, we can state the next corollary which gives a bound for $\|x^* - x^{(m)}\|$.

Corollary 2.1.

Let γ_m be defined as above and let κ_m be the norm of the inverse of A_m. Then the error $x^* - x^{(m)}$ satisfies

$$\|x^* - x^{(m)}\| \leq \sqrt{1 + \gamma_m^2 \kappa_m^2}\, \|(I - \pi_m)x^*\| \tag{2.7}$$

Proof

By Proposition (2.1) and the fact that $x^{(m)} - \pi_m x^* = A_m^{-1}(b - A_m \pi_m x^*)$, we get

$$\|\pi_m(x^* - x^{(m)})\| \leq \gamma_m \kappa_m \|(I - \pi_m)x^*\| \tag{2.8}$$

(remark that $\pi_m x^{(m)} = x^{(m)}$). Writing

$$x^* - x^{(m)} = (I - \pi_m)x^* + \pi_m(x^* - x^{(m)}) \tag{2.9}$$

and observing that the two vectors on the right hand side of (2.9) are orthogonal, we obtain

$$\|x^* - x^{(m)}\|^2 = \|(I - \pi_m)x^*\|^2 + \|\pi_m(x^* - x^{(m)})\|^2$$

which, in view of (2.8), gives the desired result (2.7). □

The above results show that the error $\|x^{(m)} - x^*\|$ will be of the same order as $\|(I - \pi_m)x^*\|$ provided that the approximate problem (2.4) is not badly conditioned.

2.2. Krylov subspace methods

Let x_0 be an initial guess at the solution x^* of (2.1) and let r_0 be the initial residual $r_0 = b - Ax_0$. If the unknown x is decomposed as $x = x_0 + z$ then clearly the new unknown z must satisfy

$$Az - r_0 = 0 \tag{2.10}$$

By a Krylov subspace method we shall refer to any method that obtains an approximation $z^{(m)}$ to problem (2.10) by applying a projection process to the system (2.10) onto the Krylov subspace $K_m = \text{Span}\,[r_0, Ar_0, \ldots, A^{m-1}r_0]$.

We shall assume throughout that the vectors $r_0, Ar_0, \ldots, A^{m-1}r_0$ are linearly independent, which means that

$$\dim(K_m) = m \tag{2.11}$$

If $V_m \equiv [v_1, \ldots, v_m]$ is any basis of K_m, then according to Section 2.1, $z^{(m)}$ can be expressed as $z^{(m)} = V_m \cdot y^{(m)}$ where $y^{(m)}$ is the solution of the m x m system

$$V_m^H A V_m \cdot y^{(m)} - V_m^H r_0 = 0 \tag{2.12}$$

and the approximate $x^{(m)}$ of problem (2.1) is related to $z^{(m)}$ by $x^{(m)} = x_0 + z^{(m)}$.

If $z^* = A^{-1}r_0$ denotes the exact solution of the system (2.10), then we notice that

$$x^* - x^{(m)} = z^* - z^{(m)} \tag{2.13}$$

which means that $x^{(m)}$ and $z^{(m)}$ admit the same error vector for (2.1) and (2.10), respectively.

3. PRACTICAL METHODS

Some algorithms based upon the Krylov subspace methods described above will now be presented. We first propose an adaptation of Arnoldi's method [1], [18] to the solution of systems of linear equations. The algorithm constructs an orthonormal basis $V_m = [v_1, \ldots, v_m]$ of K_m such that $V_m^T A V_m$ has Hessenberg form. An iterative version of this method is also given so as to avoid the storage of too large arrays in memory. Then another class of algorithms is derived from the incomplete orthogonalization method described in [18].

3.1. The method of Arnoldi

Arnoldi's algorithm builds an orthonormal basis $v_1, \ldots, v_m$ of $K_m = \text{Span}\,[r_0, Ar_0, \ldots, A^{m-1} r_0]$ by the recurrence

$$h_{k+1,k}\, v_{k+1} = A v_k = \sum_{i=1}^{k} h_{ik}\, v_i \qquad (3.1)$$

starting with $v_1 = r_0/\|r_0\|$ and choosing h_{ik}, $i = 1, \ldots, k+1$ in such a way that $v_{k+1} \perp v_1, \ldots, v_k$ and $\|v_{k+1}\| = 1$. In exact arithmetic the algorithm would be as follows.

Algorithm 3.1

1. Compute $r_0 = b - Ax_0$ and take $v_1 := r_0/\|r_0\|$.
2. For k := 1 until m do

$$w := Av_k - \sum_{i=1}^{k} h_{ik}\, v_i \text{ with } h_{ik} := (Av_k, v_i) \qquad (3.2)$$

$$h_{k+1,k} := \|w\| \qquad (3.3)$$

$$v_{k+1} := w/h_{k+1,k}$$

See [18] for some remarks on the practical realization of this algorithm. It is easily seen that $[v_1, v_2, \ldots, v_m]$ is an orthonormal basis of K_m and that the matrix $V_m^H A V_m$ is the Hessenberg matrix H_m whose nonzero elements are the h_{ij} defined by (3.2) and (3.3). As a consequence the vector $V_m^H r_0$ in (2.7) is equal to $\beta \cdot V_m^H v_1 = \beta e_1$ where $\beta = \|r_0\|$. Thus the system (2.7) becomes

$$H_m \cdot y^{(m)} = \beta \cdot e_1 \tag{3.4}$$

and the approximate solution $x^{(m)}$ defined in Section 2.2 reads $x^{(m)} = x_0 + z^{(m)}$ where

$$z^{(m)} = \beta V_m H_m^{-1} e_1 \tag{3.5}$$

The following estimate for the residual norm $\|b - Ax^{(m)}\|$ is very useful as a stopping criterion

$$\|b - Ax^{(m)}\| = h_{m+1,m} \, |e_m^H y^{(m)}| \tag{3.6}$$

Equality (3.6) follows immediately from the relation

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

which can be derived from the algorithm and from equality (2.8).

An interesting practical method would be to generate the vectors v_k and the matrix H_k, $k = 1, 2, \ldots, m, \ldots$, to compute periodically the estimate $h_{m+1,m} \, |e_m^H y^{(m)}|$ of the norm of the residual and to stop as soon as this is small enough. As was suggested in [15] for the symmetric case, there are various ways of updating $|e_m^H y^{(m)}|$ without even actually computing the vector $y^{(m)}$. Let us give a few indications about the problem of computing the estimation $|e_m^H y^{(m)}|$, since it will appear in several parts along the paper. Parlett [15] suggests utilizing a recurrence relation proposed by Paige and Saunders [14], which is based upon the LQ factorization of H_m.

Another interesting possibility is to perform the more economical factorization provided by the Gaussian elimination with partial pivoting on the matrix H_j. The factorization of H_j can be easily performed by using the information at the previous step. Supposing that no pivoting has been necessary for steps 1 through j-1, and writing the LU factorization of H_j, $H_j = LU$, it can be easily seen that $\rho_j = h_{j+1,j} |e_m^H y^{(m)}|$ is simply

$$\rho_j = h_{j+1,j} \beta \left| \left(\prod_{i=1}^{j-1} \ell_i \right) / u_{jj} \right|$$

where the ℓ_i, $i = 1, \ldots, j-1$, are the successive pivots. More generally it can be shown that when no pivoting has been necessary at steps i, $i \in I$, where $I \subset \{1, 2, \ldots, j-1\}$, then ρ_j becomes

$$\rho_j = h_{j+1,j} \beta \left| \left(\prod_{i \in I} \ell_i \right) / u_{jj} \right| .$$

This means that ρ_j can be updated at each step at a negligible cost. Finally, after it is decided that the estimate of the residual norm is small enough, the final factorization of H_m will be used to fully solve the system (3.4). The Gaussian elimination with partial pivoting gives satisfactory results in general, but one might as well use a more stable decomposition, as the LQ decomposition in [14], [15] although at a high cost.

As m increases, the process of computing the v_i becomes, unfortunately, intolerably expensive and core memory demanding. To remedy this, one can use the algorithm in an iterative way, as is described next.

3.2. Iterative Arnoldi method

Due to core memory capacities, the number m of steps in Algorithm 3.1 is inevitably limited. After having computed the approximate solution $x^{(m)}$ with the maximum number of steps allowed, one may find that the accuracy is still unsatisfactory. This naturally raises the question of how to improve the accuracy of $x^{(m)}$. The simplest idea is to restart the algorithm with x_0 replaced by the approximation $x^{(m)}$ obtained. The idea is similar to that of the m step steepest descent in the symmetric case. (See [6].) One can restart as many times as necessary to ensure satisfactory accuracy. We now give a more detailed description of this iterative version. Let us start with an initial guess x_0 and form $r_0 = b - Ax_0$. Then construct H_m and V_m by algorithm (3.1) and compute the approximate solution $x_1^{(m)} = x_0 + z_1^{(m)}$. The estimation (3.6) can be used to determine whether the process must be stopped or restarted. Suppose a restart is necessary. Then take $x_1 = x_0 + z_1^{(m)}$ and compute $r_1 = b - Ax_1$. (Remark that r_1 is also equal to the residual $r_0 - Az_1^{(m)}$.) Construct again V_m and H_m starting with $v_1 = r_1/\|r_1\|$ in Algorithm 3.1. Then an approximate solution $z_2^{(m)}$ to the equation $Az = r_1$ is obtained yielding the new approximation $x_2 = x_1 + z_2^{(m)}$ to the solution x* and so forth.

At the s-th iteration the approximate solution x_s is equal to $x_0 + z_1^{(m)} + \ldots + z_s^{(m)}$. Thus the algorithm can be formulated as follows. (The subscript (m) is dropped for simplifications.)

Algorithm 3.2

1. Start. Choose m and x_0; $r_0 := b - Ax_0$.

2. For s := 0, 1,..., do

   • compute v_1, v_2,..., v_m and H_m by Algorithm 3.1 starting with $v_1 = r_s/(\beta := \|r_s\|)$

   • Solve the system $H_m \cdot y = \beta \cdot e_1$

   • $z_{s+1} := V_m \cdot y$

   • $x_{s+1} := x_s + z_{s+1}$

   • $r_{s+1} := r_s - Az_{s+1}$

   • If $h_{m+1,m} |e_m^T y| < e$, stop else continue

3.3. Incomplete orthogonalization methods

3.3.1. The construction of the vectors v_1,..., v_m by Algorithm 3.1 amounts to orthogonalizing the vectors Av_k against all previous vectors v_1,..., v_k. This is costly and some numerical observations suggest to orthogonalize Av_k against the preceding p+1 vectors rather than all (see [18]).

The system produced is such that $(v_i, v_j) = \delta_{ij}$ for i,j satisfying $|i-j| \leq p$.

Algorithm 3.3

1. Choose p and m such that p < m; compute $r_0 := b - Ax_0$ and $v_1 = r_0/\|r_0\|$.

2. For j := 1, 2,..., m do

   $i_0 := \max(1, j-p+1)$

   $w := Av_j - \sum_{i=i_0}^{j} h_{ij} v_i$ with

$$h_{ij} := (Av_j, v_i) \tag{3.7}$$

$$v_{j+1} := w/(h_{j+1,j} := \|w\|) \tag{3.8}$$

Under the assumption (2.11), this algorithm will not stop before the m-th step and will produce a system of vectors $v_1, \ldots, v_m$ locally orthogonal and a (banded) Hessenberg matrix of the form

$$\tilde{H}_m = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

whose nonzero elements are computed from (3.7) and (3.8). The generalized Lanczos approximation $z^{(m)}$ must satisfy the equations

$$V_m^H A V_m y^{(m)} - V_m^H r_0 = 0 \tag{3.9}$$

$$z^{(m)} = V_m \quad y^{(m)}$$

In the present case, however, the matrix $V_m^H A V_m$ does not have any particular structure as before, so we need to transform (3.9) into a simpler problem.

Let us set $\hat{H}_m = (V_m^H V_m)^{-1} V_m^H A V_m$. Note that this is just the matricial representation of the linear operator $A_m = \pi_m A|_{K_m}$ (see Section 2.1), in the basis $\{v_1, v_2, \ldots, v_m\}$. It was shown in [18] that $\hat{H}_m$ <u>differs from</u> $\tilde{H}_m$ <u>only in its last column</u>. More precisely

<u>Theorem 3.1</u>

<u>Let</u> $s_m = h_{m+1,m}(V_m^H V_m)^{-1} V_m^H v_{m+1}$. <u>Then</u>

$$\hat{H}_m = \tilde{H}_m + s_m e_m^H \tag{3.10}$$

<u>Proof</u>. From Algorithm 3.3 we get the basic equation

$$A V_m = V_m \tilde{H}_m + h_{m+1,m} v_{m+1} e_m^H$$

which yields (3.10) on multiplying by $(V_m^H V_m)^{-1} V_m^H$. □

Multiplying (3.9) by $(V_m^H V_m)^{-1}$ gives the equivalent equation

$$\hat{H}_m y^{(m)} - (V_m^H V_m)^{-1} V_m^H r_0 = 0$$

Observing that $(V_m^H V_m)^{-1} V_m^H r_0 = \beta e_1$ where $\beta = \|r_0\|$, we obtain the system

$$\hat{H}_m y^{(m)} - \beta e_1 = 0 \tag{3.11}$$

If we set $\hat{y}^{(m)} = \beta \hat{H}_m^{-1} e_1$ and $\tilde{y}^{(m)} = \beta \tilde{H}_m^{-1} e_1$, then by the Shermann and Morrison formula [7] these two vectors are related by

$$\hat{y}_m = \tilde{y}_m - \sigma \tilde{H}_m^{-1} s_m \tag{3.12}$$

where $\sigma = e_m^H \tilde{y}^{(m)} / (1 + e_m^H \tilde{H}_m^{-1} s_m)$.

On the practical side the only difficulty lies in the computation of the corrective column s_m. Note that $s_m = h_{m+1,m} V_m^+ v_{m+1}$ and that s_m is the solution of the least square problem (see [19])

$$\min_s \|V_m s - h_{m+1,m} v_{m+1}\| \tag{3.13}$$

for which many efficient algorithms are available (see [3], [13]). It should be added that only a moderate accuracy is needed in practice, so the bidiagonalization algorithm BIDIAG described in [13] is suitable for solving (3.13) with moderate accuracy. We can now give an algorithm based upon all the above observations.

Algorithm 3.4. Incomplete Orthogonalization with Correction

Start. Choose two integers p and m with $p < m$. Compute $r_0 := b - Ax_0$, $\beta = \|r_0\|$; $v_1 := r_0/\beta$.

Iterate. Comment compute $\tilde{H}_m$ and $v_1, \ldots, v_m$.

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

$i_0 := \max(1, j-p)$

$$w := Av_j - \sum_{i=i_0}^{j} (h_{ij} := (Av_j, v_i)) \times v_i$$

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

Correct:

1. Compute least square solution s_m of (3.13).

2. Compute $\tilde{y}_m := \beta \tilde{H}_m^{-1} e_n$

$$x := \tilde{H}_m^{-1} s_m$$

$$\sigma := e_m^H \tilde{y}_m / (1 + e_m^H x)$$

$$\hat{y}_m := \tilde{y}_m - \sigma x$$

3. Form the approximate solution

$$x^{(m)} = x_0 + V_m \cdot \hat{y}_m$$

We shall now give some additional practical details.

1. If necessary, the vectors $v_1, v_2, \ldots, v_m$ may be stored in auxiliary memory, one by one as soon as they are computed. Only the p vectors $v_j, v_{j-1}, \ldots, v_{j-p+1}$ must be kept in main memory for more efficiency.

2. The storage of $\tilde{H}_m$ now requires only the storage of (p+1) x m elements instead of the previous m^2.

3. For the choice of the integer p we should first point out that p is limited by the available core memory. In theory the larger p, the better. If p is large, the system $(v_1, \ldots, v_m)$ will, in practice, be close to orthogonality and the solution of the least square problem (3.13) in step correct becomes easier [at the limit if p = m then the solution is just $h_{m+1,m} V_m^H v_{m+1} = 0$]. But in that case the computations in the step iterate are more expensive. If p is too small, on the other hand, it is very likely that the problem (3.13) will become difficult to solve (if not impossible numerically) as the vectors $(v_1, \ldots, v_m)$ will become nearly linearly dependent. Note that this depends also upon m. When m = p the system is orthonormal and as m increases it is observed that the system departs from

orthogonality, in a slow manner at the beginning. All these observations suggest that p must first be chosen according to the main memory capacity and some arbitrary limitation $p \leq p_{max}$. Afterwards, a maximum number of steps m_{max} should be fixed. Then a test must be included at the end of the step iterate in order to shift to the correction step as soon as the system $\{v_1, v_2, \ldots, v_{j+1}\}$ is suspected to be too far from orthonormal, as for example

$$\underline{\text{if}}\ |(v_{j+1}, v_1)| \geq \eta\ \underline{\text{goto}}\ \text{correct}$$

where η is a certain tolerance. The heuristic criterion given above is not the best.

4. When the matrix A is symmetric, then by taking p = 2 we obtain a version of the conjugate gradient method which is known to be equivalent to the Lanczos algorithm (see [14]). In that case the vectors $v_1, \ldots, v_m$ are theoretically orthogonal. Suppose now that A is nearly symmetric and take p = 2 again. By a continuity argument it is clear that the system $(v_1, \ldots, v_m)$ will be nearly orthonormal, making the choice p = 2 optimal in a certain sense. This suggests that when it is known that A is close to a symmetric matrix, p could be taken small (or even p = 2). However, it is not easy to give a rigorous meaning to the notion of nearly symmetric and it is even more difficult to monitor automatically the choice of the parameter p.

3.3.2. In the following we develop another algorithm which is, in particular, more appropriate for the cases of almost symmetric matrices. As pointed out above, the correction step can be expensive and one may ask whether an acceptable accuracy could be achieved by ignoring the corrective step and replacing the approximate solution $x^{(m)} = x_0 + V_m \hat{y}_m$ by

$$\tilde{x}^{(m)} = x_0 + V_m \tilde{y}_m \tag{3.14}$$

The answer is yes, provided that V_{m+1} is not too far from orthonormal. In effect, writing $\tilde{H}_m = \hat{H}_m - s_m e_m^H$, we can derive the following analogue of (3.12)

$$\tilde{y}_m = \hat{y}_m + \frac{h_{m+1,m} e_m^H \hat{y}_m}{1 - e_m^H \hat{H}_m^{-1} s_m} \hat{H}_m^{-1} s_m \tag{3.15}$$

It is remarkable that, by (3.6), the term $h_{m+1,m} e_m^H \hat{y}_m$ is equal to the residual norm $\|r_0 - Az^{(m)}\|$ except for the sign, and hence it becomes smaller as m increases. If $\{v_1, \ldots, v_{m+1}\}$ is nearly orthonormal then $V_m^H v_{m+1}$ is nearly zero and so will be s_m in general. This shows that in general the second term on the right hand side of (3.15) can be neglected (in comparison with $\hat{y}_m$) as long as V_{m+1} remains nearly orthonormal. This fact is confirmed by the experiments and it is observed that the residual norms behave in the same manner as the residual norms obtained for the incomplete orthogonalization method applied to the eigenvalue problem (see [18], section 4.2).

The residual norms $\|r_0 - A\tilde{x}_m\|$ decrease rapidly until a certain step and then start oscillating and decreasing more slowly. This suggests restarting immediately after a residual norm is larger than the previous one. Here again the formula (3.6) remains very useful for estimating the residual norm. This leads to the following algorithm.

Algorithm 3.5. Incomplete Orthogonalization without Correction

Start. $\tilde{x} := x_0$; $\tilde{r} := b - Ax_0$; $\beta := \|\tilde{r}\|$; $v_1 := \tilde{r}/\beta$;

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

1. compute

$$h_{j+1,j}\, v_{j+1} = Av_j - \sum_{i=i_0}^{j} h_{ij}\, v_i$$

where i_0 and the h_{ij}'s are as in Algorithm 3.4.

2. Update the factorization of H_j and the estimate ρ_j of the residual norm (see §3.1).

3. Test for convergence performed every q steps only (e.g., every q = 5 steps).

   a. If $\rho_j < \varepsilon$ goto restart.

   b. If $\rho_j > \rho_{j-q}$ goto restart; otherwise take m := j and continue.

Restart:

$$\tilde{z}^{(m)} := \beta V_m H_m^{-1} e_1$$

$$\tilde{x} := \tilde{x} + \tilde{z}^{(m)}$$

$$\tilde{r} := \tilde{r} - A\tilde{z}^{(m)}$$

$$\beta := \|\tilde{r}\|$$

$$v_1 := \tilde{r}/\beta$$

If $\beta \leq \varepsilon$ stop else goto iterate

The numerical experiments (§5) will reveal that this last algorithm is to be preferred to the iterative Arnoldi Algorithm and to the incomplete orthogonalization method with correction. Surprisingly, it is often the case that no restart is necessary, even for matrices that are not nearly symmetric.

We shall conclude this section by a remark concerning the application of preconditioning techniques to the algorithms described above. Suppose that we can find a matrix M for which linear systems are easily solvable and such that $M^{-1}A$ is closer to the identity than A. In this case it is advantageous, in general, to replace the system $Ax = b$ by the new system $M^{-1}Ax = M^{-1}b$ before applying one of the previous methods. There are two reasons for this. The first is that the rate of convergence of the second system will, in general, be higher than that of the first because the spectrum will be included in a disk with center one and with small radius, and the next section will show that in that case the smaller the radius, the higher the rate of convergence. The second is that $M^{-1}A$, which is close to the identity matrix, is clearly close to a symmetric matrix (the Identity) so that the application of incomplete orthogonalization without correction is most effective (cf §5.5).

4. RATES OF CONVERGENCE FOR THE KRYLOV SUBSPACE METHODS

4.1. Introduction

We shall now consider the problem of the convergence of the approximate $x^{(m)}$ toward the exact solution x^*. We first point out that the convergence is achieved in at most N steps where N is the dimension of A. (This is immediate from the fact that K_N is the whole subspace $\mathbb{C}^N$ and from the definition 2.2.) Therefore, the problem is not to show the convergence but rather to establish theoretical error bounds showing that one can obtain a satisfactory accuracy for values of m much less than the dimension N, which is supposed to be very large. Another way of stating the problem is to suppose that A is an operator on a Hilbert space ($N = \infty$) such that the convergence, the rate of convergence..., of the

infinite sequence $x^{(m)}$ can be discussed. We shall not, however, adopt this extension in the present paper.

In view of relation (2.13) it is equivalent to study either the convergence of $x^{(m)}$ to x^* or the convergence of $z^{(m)}$ to z^*. In addition, Corollary 2.1 shows that the convergence can be studied in terms of $\|(I - \pi_m)z^*\|$ where π_m is the orthogonal projection onto the Krylov subspace $K_m = \text{Span}\,[r_0, Ar_0, \ldots, A^{m-1} r_0]$. Let us denote by P_k the space of polynomials of degree not exceeding k. Then, a useful expression for the distance $\|(I - \pi_m)z^*\|$ can be derived by remarking that K_m is nothing but the subspace of $\mathbb{C}^N$ constituted by all the elements $q(A)r_0$ where q belongs to P_{m-1}.

Proposition 4.1. The distance $\|(I - \pi_m)z^*\|$ between z^* and the Krylov subspace K_m satisfies

$$\|(I - \pi_m)z^*\| = \min_{\substack{p \in P_m \\ p(0)=1}} \|p(A)z^*\| \tag{4.1}$$

Proof. The following equalities are easy to show

$$\|(I - \pi_m)z^*\| = \min_{z \in K_m} \|z^* - z\| = \min_{q \in P_{m-1}} \|z^* - q(A)r_0\|$$

$$= \min_{q \in P_{m-1}} \|z^* - q(A)Az^*\| = \min_{q \in P_{m-1}} \|(I - Aq(A))z^*\|$$

$$= \min_{\substack{p \in P_m \\ p(0)=1}} \|p(A)z^*\| \quad \square$$

In order to obtain an upperbound for (4.1) we shall assume that A admits N eigenvectors $\phi_1, \phi_2, \ldots, \phi_N$ of norm one, associated with the eigenvalues $\lambda_1, \ldots, \lambda_N$. Then the solution z^* can then be expressed as

$$z^* = \sum_{i=1}^{N} \alpha_i \phi_i$$

and we can formulate the next theorem.

Theorem 4.1

Set $\alpha = \sum_{i=1}^{N} |\alpha_i|$, where the α_i are the components of the solution z^* in the eigenbasis of A.

Then

$$\|(I - \pi_m)z^*\| \leq \alpha \min_{\substack{p \in P_m \\ p(0)=1}} \max_{j=1,\ldots,N} |p(\lambda_j)| \tag{4.2}$$

Proof. Let $p \in P_m$, with $p(0) = 1$. Then

$$\|p(A)z^*\| = \|p(A) \sum_{i=1}^{N} \alpha_i \phi_i\| = \|\sum_{j=1}^{N} p(\lambda_i)\alpha_i \phi_i\|$$

$$\leq \sum_{i=1}^{N} \|\alpha_i\, p(\lambda_i)\phi_i\| \leq \sum_{i=1}^{N} |\alpha_i|\, |p(\lambda_i)|$$

$$\leq \left[\sum_{i=1}^{N} |\alpha_i|\right] \times \max_{j=1,\ldots,N} |p(\lambda_i)|$$

Therefore, for any polynomial of degree not exceeding m such that $p(0) = 1$ we have

$$\|p(A)z^*\| \leq \alpha \max_{j=1,\ldots,N} |p(\lambda_j)| \tag{4.3}$$

Hence $\min_{\substack{p \in P_m \\ p(0)=1}} \|p(A)z^*\| \leq \alpha \min_{\substack{p \in P_m \\ p(0)=1}} \max_{j=1,\ldots,N} |p(\lambda_j)|$ which by equality (4.1) completes the proof. □

We point out here that from classical results it can be shown that the polynomial realizing the minimum in (4.2) exists and is unique provided that $m \leq N$ (see [11]). We should also add that there is unfortunately no upper bound for α.

We shall set throughout

$$\varepsilon^{(m)} = \min_{\substack{p \in P_m \\ p(0)=1}} \max_{j=1,\ldots,N} |p(\lambda_j)| \tag{4.4}$$

so that inequality (4.2) simplifies to

$$\|(I - \pi_m)z^*\| \leq \alpha\, \varepsilon^{(m)} \tag{4.5}$$

and the result (2.7) becomes

$$\|x^* - x^{(m)}\| = \|z^* - z^{(m)}\| \leq \alpha\sqrt{1 + \gamma_m^2 \kappa_m^2}\, \varepsilon^{(m)}$$

We, therefore, need to show that the sequence $\varepsilon^{(m)}$ decreases rapidly to zero. Note that $\varepsilon^{(N)} = 0$ which shows again that the process will give the exact solution in at most N steps. The rate of convergence of the sequence $\varepsilon^{(m)}$ to zero provides a bound for the actual rate of convergence. Estimating $\varepsilon^{(m)}$ is, unfortunately, a difficult problem in general. The number $\varepsilon^{(m)}$ is the degree of best approximation of the zero function by polynomial of degree m satisfying the constraint $p(0) = 1$, over the set $\lambda_1, \lambda_2, \ldots, \lambda_N$ (see [11]).

4.2. An exact expression for $\varepsilon^{(m)}$

The following theorem gives an expression for $\varepsilon^{(m)}$ in terms of m + 1 eigenvalues of A.

Theorem 4.2

Let $m \leq N-1$. Then there exist m+1 eigenvalues which, without ambiguity can be labelled $\lambda_1, \lambda_2, \ldots, \lambda_{m+1}$ such that

$$\varepsilon^{(m)} = \left[\sum_{\substack{j=1 \\ k \neq j}}^{m+1} \prod_{k=1}^{m+1} \frac{|\lambda k|}{|\lambda j - \lambda k|} \right]^{-1} \qquad (4.6)$$

We omit the proof of this equality. An analogue result will be proved in a forthcoming paper dealing with the convergence of Arnoldi-like methods for computing eigenelements.

The result does not specify which are the eigenvalues $\lambda_1, \ldots, \lambda_{m+1}$ but it still gives an interesting indication. If the origin is well separated from the spectrum then $\varepsilon^{(m)}$ is likely to be very small. Indeed if λ_1 is, for example, the eigenvalue the closest to zero, among those eigenvalues involved in the theorem, then, in general, we shall have $|\lambda_k| > |\lambda_1 - \lambda_k|$, $k = 1, \ldots, N$ as seen in Figure 1. Therefore,

$$\prod_{k=2}^{m+1} \frac{|\lambda k|}{|\lambda k - \lambda 1|} >> 1$$

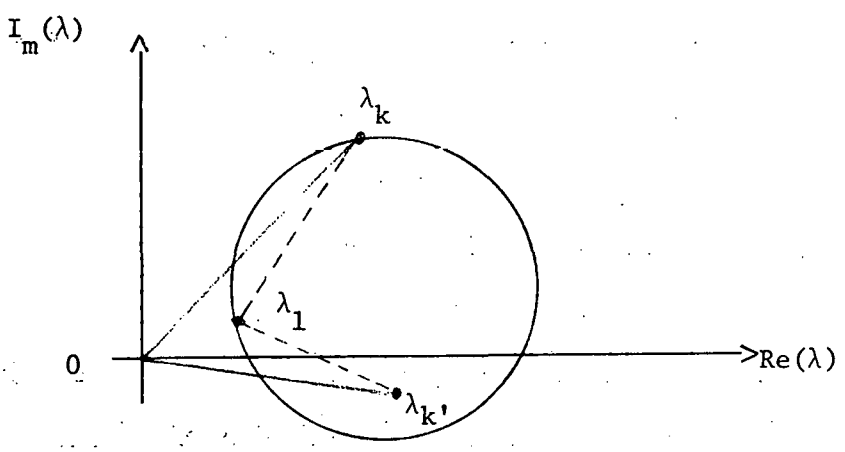


Figure 1.

and it is seen from (4.6) that $\varepsilon^{(m)}$ will be small. There are particular distributions of the eigenvalues where $\varepsilon^{(m)}$ is known exactly (for $m = N-1$). But, in general, the result (2.14) is not useful for giving an estimation of the rate of convergence. Upperbounds for $\varepsilon^{(m)}$ must be established for that purpose.

4.3. Bounds for $\varepsilon^{(m)}$

In the real case one usually obtains bounds for $\varepsilon^{(m)}$ by majorizing the discrete norm $\max_{j=1,N} |p(\lambda_j)|$ by the continuous norm $\max_{z\in I} |p(\lambda)|$ where I is an interval (or the union of two intervals) containing the eigenvalues λ_j and not zero.

In the complex case, however, one encounters the difficulty of choosing an adequate continuum containing all the eigenvalues and not zero. An infinity of choices are possible but except for some particular shapes such as circles, ellipses..., there is no simple expression for the minimax quantity $\min_{\substack{p\in P_m \\ p(0)=1}} \max_{z\in D} |p(z)|$.

We first deal with the simplest case where all the eigenvalues of A are real and positive. The next case to consider is naturally the case where the eigenvalues are almost real. The general case will be considered in subsections 4.3.3 and 4.3.4.

4.3.1. Case of a purely real spectrum

Theorem 4.3

Suppose that all the eigenvalues of A are real and positive and let λ_{min} and λ_{max} be the smallest and the largest of them.
Then

$$\|(I - \pi_m)z^*\| \leq \alpha/T_m(\gamma) \tag{4.7}$$

where α is as before, $\gamma = (\lambda_{max} + \lambda_{min})/(\lambda_{max} - \lambda_{min})$ and where T_m is the Tchebycheff polynomial of degree m of the first kind.

This result is an immediate application of a well-known bound for (4.4) when the λ_i are real [2]. It is also possible to establish some

results when the eigenvalues are known to lie in two or more intervals (see [2], [10]).

Inequality (2.11) shows that the Generalized Lanczos method converges at least as rapidly as $[T_m(\gamma)]^{-1} \simeq (\gamma + \sqrt{\gamma^2 - 1})^{-m}$ such that the rate of convergence is bounded by $\gamma + \sqrt{\gamma^2 - 1}$.

Finally note that similar results can easily be obtained if all the eigenvalues are purely imaginary or if they lie on a straight line of $\mathbb{C}$, containing the origin.

4.3.2. Almost purely real spectra

In the following we shall assume that the spectrum lies inside a certain ellipse which has center c on the real line and foci c + e, c - e where e is the eccentricity. Furthermore we shall assume that the origin is not inside that ellipse (see Figure 2).

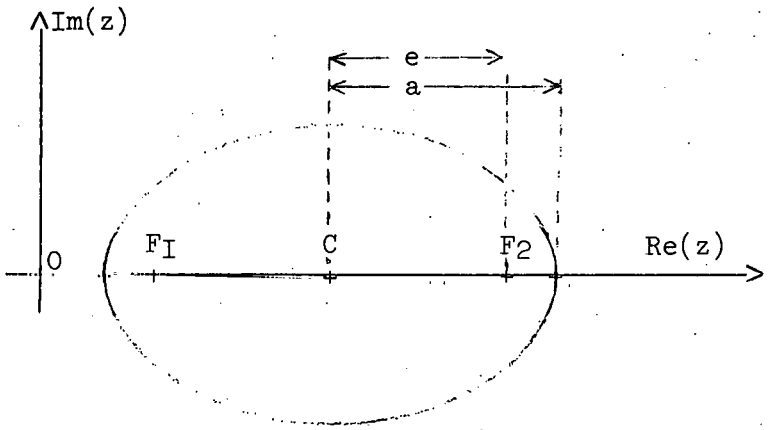


Figure 2.

Let us denote by E the closed domain bounded by the ellipse defined above. Consider the variable transform z' = (c - z)/e;then $\varepsilon^{(m)}$ satisfies the inequality

$$\varepsilon^{(m)} \leq \min_{\substack{p\in P_m \\ p(c/e)=1}} \max_{z'\in E'} |p(z')| \tag{4.10}$$

where the domain E' is bounded by the ellipse centered at origin with eccentricity one and major semi-axis a/e. It was shown by Clayton [4] that the above mini-max is realized for the polynomial $T_m(z')/T_m(c/e)$.

Theorem 4.4

Assume that the eigenvalues of A lie within an ellipse with center c on the real axis, foci c + e, c - e, and with major semi-axis a. Suppose that the origin is not inside this ellipse. Then

$$\varepsilon^{(m)} \leq \frac{T_m(a/e)}{|T_m(c/e)|} \tag{4.11}$$

In view of (4.10) this inequality is a simple corollary of Clayton's result. Since the proof is tedious, we shall give a direct proof of (4.11) and bypass Clayton's result.

Proof. Considering the particular polynomial $T_m(z')/T_m(c/e)$ we get from (4.10)

$$\varepsilon^{(m)} \leq \max_{z'\in E'} \left| \frac{T_m(z')}{T_m(c/e)} \right| \tag{4.12}$$

By the maximum principle, the maximum on the right hand side is realized for z' belonging to the boundary $\partial E'$ of the ellipse E' centered at the origin and having major semi-axis a/e and eccentricity one. Thus (4.2) becomes

$$\varepsilon^{(m)} \leq \frac{1}{|T_m(c/e)|} \cdot \max_{z'\in\partial E'} |T_m(z')| \tag{4.13}$$

Consider now the transform u: $w \leftrightarrow z' = \frac{1}{2}(w + \frac{1}{w})$. It is known [11], [17] that when w belongs to the circle C_ρ centered at the origin and having radius ρ, z' will belong to the ellipse ∂E_ρ having eccentricity

one and major semi-axis $(\rho + \rho^{-1})/2$. We may take $\rho = a/e + \sqrt{(a/e)^2 - 1}$ such that $\partial E\rho$ is just $\partial E'$. $T_m(z)$ can be defined by $T_m(z) = \mathrm{ch}(m.u)$ where u and z are related by $\mathrm{ch}(u) = z$. Setting $e^u = w$ we see that another definition for $T_m(z)$ is $T_m(z) = (w^m + w^{-m})/2$ where w and z are related by $(w + w^{-1})/2 = z$. Hence

$$\max_{z' \in \partial E'} |T_m(z')| = \max_{w \in C\rho} \frac{1}{2} |w^m + w^{-m}|$$

$$= \max_{\theta \in [0, 2\pi]} \frac{1}{2} |\rho^m e^{im\theta} + \rho^{-m} e^{-im\theta}|$$

It is easily seen that the above maximum is just

$$\frac{1}{2}(\rho^m + \rho^{-m}) = \frac{1}{2}\left[\left(\frac{a}{e} + \sqrt{\left(\frac{a}{e}\right)^2 - 1}\right)^m + \left(\frac{a}{e} - \sqrt{\left(\frac{a}{e}\right)^2 - 1}\right)^{-m}\right]$$

$$= T_m\left(\frac{a}{e}\right)$$

which completes the proof. □

The upperbound $T_m(a/e)/T_m(|c/a|)$ for $\varepsilon^{(m)}$ is asymptotically equivalent to

$$\left[\frac{a/e + \sqrt{(a/e)^2 - 1}}{|c/e| + \sqrt{(c/e)^2 - 1}}\right]^m$$

so that an upperbound for the asymptotic rate of convergence is given by

$$\tau = \frac{|c| + \sqrt{c^2 - e^2}}{a + \sqrt{a^2 - e^2}} \tag{4.14}$$

When the eigenvalues are all real, then the ellipse degenerates to the interval $[\lambda_1, \lambda_N]$ and we shall have $e = a = (\lambda_N - \lambda_1)/2$, $c = (\lambda_1 + \lambda_N)/2$ such that τ will become $\gamma + \sqrt{\gamma^2 - 1}$ with $\gamma = (\lambda_N + \lambda_1)/(\lambda_N - \lambda_1)$. This means that the result (2.17) coincides with that of Corollary 2.1 when the spectrum lies on the real line.

Consider now the family of all ellipses having center c and major semi-axis a and let the eccentricity decrease from a to zero. Then

the ellipse will pass from the interval (c - a, c + a) to the circle with center c and radius a. It is easily seen that the bound (4.14) for the rate of convergence will decrease from $\tau_{max} = |c/a| + \sqrt{(c/a)^2 - 1}$ to $\tau_{min} = |c/a|$. Therefore, we may assert that the convergence is likely to be better if the eigenvalues are close to the real line and that when the spectrum has a circular shape the convergence is likely to be slower. Note that the comparison is made for the same relative separation $|c/a|$ from the origin. The above comments are confirmed by a numerical example in section 5.1.

Before considering the more general case where the ellipse containing the spectrum does not stretch along the real axis, let us point out that inequality (4.11) cannot be improved as Clayton's result shows. By this we mean that if one replaces the discrete set $\{\lambda_1, \ldots, \lambda_N\}$ by the set of all points contained in an ellipse of the form described in Figure 2, one cannot find a better inequality than (4.11).

4.2.3. Spectrum contained in an ellipse

If the spectrum lies inside an ellipse with center c and foci c + e, c - e where now both c and e are complex, it is easily seen that the proof of Theorem 4.4 is still valid. Therefore, we can establish that

$$\varepsilon^{(m)} \leq \frac{|T_m(a/e)|}{|T_m(c/e)|} \tag{4.15}$$

Where c, e are the center and the "eccentricity" and are complex, while a the (complex) major semi-axis is such that c + a and c - a are the coordinates of the two points of the ellipse situated on the major semi-axis. Note that a/e is real while c/e is not. The interpretation of (4.15) will, therefore, not be easy in general. It can be shown,

however, that the right hand side of (4.15) converges to zero as $m \to \infty$ (see [12]). The next subsection gives a result which is weaker, in general, but easier to interpret.

4.3.4. Spectrum contained in a circle

In this subsection we shall assume that the spectrum lies in a certain domain bounded by a circle having center c and radius a. Furthermore, let us assume that the origin lies outside the circle (cf. Figure 3).

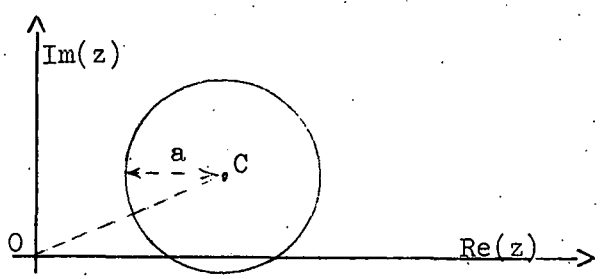


Figure 3.

Then we have

Theorem 4.4

Suppose that there exists a disk $D(c,a)$ with center c and radius a, that contains all the eigenvalues of A and not the origin. Then

$$\varepsilon^{(m)} \leq \left|\frac{a}{c}\right|^m \tag{4.16}$$

Proof. Consider the particular polynomial $\bar{p}(z) = [\frac{c-z}{c}]^m$. p has degree m and satisfies $\bar{p}(0) = 1$. Hence, by (2.13)

$$\varepsilon^{(m)} \leq \max_{j=1,\ldots,N} |\bar{p}(\lambda_j)| \leq |\frac{c-\lambda_j}{c}|^m \leq |\frac{a}{c}|^m \quad \square$$

The coefficient $|a/c|$ in (2.21) is smaller than one and one can even choose an "optimal" circle for which $|a/c|$ is the least. The optimal center $\bar{c}$ should minimize $\max_{j=1,\ldots,N} |(c - \lambda j)/c|$ over all complex c, $c \neq 0$ and the optimal radius $\bar{a}$ is simply $\max_{j-1,\ldots,N} |\bar{c} - \lambda j|$. The inequality (2.21) is the best bound possible for $\varepsilon^{(m)}$ that can be obtained by replacing the discrete set $\{\lambda_1,\ldots, \lambda_N\}$ by the disk D(c,a) in the formula (2.13). This is due to the next theorem, proved by Zorantonello in [22].

Theorem 2.3

The polynomial $((c - z)/c)^m$ is the polynomial of degree m having least uniform norm over the disk D(c,a) when $a < |c|$. Furthermore

$$\min_{\substack{p\in P_m \\ p(0)=1}} \max_{z\in D(c,a)} = |\frac{a}{c}|^m.$$

5. NUMERICAL EXPERIMENTS

The experiments described in subsections 5.1 to 5.4 have been performed on the Prime 650 computer of the Department of Computer Science at the University of Illinois at Urbana-Champaign. The computations have been made in double precision, using a 48-digit mantissa.

5.1.

The purpose of this first experiment is to illustrate the comments of section 4.3.2 on the convergence properties in the case of

complex eigenvalues. Let us consider the block diagonal matrix A whose diagonal blocks are 2×2 and have the form

$$D_k = \begin{bmatrix} d_k & e_k \\ -e_k & d_k \end{bmatrix} , \quad k + 1, 2, \ldots, n$$

The d_k and e_k are chosen in such a way that the eigenvalues $\lambda_k = d_k \pm ie_k$ of A lie on the ellipse having center $c = 1$ and major semi-axis $a = 0.8$. The eccentricity e varies from $e = 0$ to $e = 0.8$. The real parts d_k of the eigenvalues are uniformly distributed on the interval $[c - a, c + a]$. In other words

$$d_k = 0.2 + \frac{k - 1}{n - 1} ; \quad e_k = (a^2 - e^2)^{1/2} \left[1 - \frac{(d_k - c)^2}{a^2}\right]^{1/2}$$

$$k = 1, 2, \ldots, n$$

where $c = 1$; $a = 0.8$; $0 \leq e \leq 0.8$. The number of blocks is $n = 40$ so that A has dimension $N = 80$.

We compare for different values of e the estimated logarithmic rates of convergence $\rho_{est} = \text{Log}(\tau)$, where τ is given by (4.14), with the "actual" logarithmic rates $-\frac{1}{m} \text{Log}(\|x^* - x^{(m)}\|)$ where x^* and $x^{(m)}$ are the exact and the approximate solutions, respectively. The method used was Arnoldi's algorithm described in section 3.1. The right hand side b of the system $Ax = b$ was the vector $b = Af$ where $f = (1, 1, \ldots, 1)^T$ so the solution is equal to f. The starting vector x_0 was set to zero. The next table gives the results obtained when $m = 30$ for various values of e.

TABLE 1.

e	$\|x^*-x^{(m)}\|$	ρ_{act}	ρ_{est}
0.00	2.68×10^{-3}	0.199	0.223
0.10	2.38×10^{-3}	0.201	0.224
0.20	2.11×10^{-3}	0.205	0.228
0.30	1.69×10^{-3}	0.212	0.237
0.40	1.18×10^{-3}	0.225	0.250
0.50	6.71×10^{-4}	0.243	0.270
0.60	2.62×10^{-4}	0.275	0.303
0.70	4.22×10^{-5}	0.335	0.367
0.75	6.40×10^{-6}	0.398	0.432
0.79	1.62×10^{-7}	0.521	0.555
0.80	1.55×10^{-10}	0.753	0.693

Note that in passing from e = 0.79 to e = 0.80 the spectrum of the matrix A becomes purely real and consists in 40 double eigenvalues, which explains the jump in the actual rate of convergence.

The values ρ_{act} and ρ_{est} of Table 1 are plotted in the next figure.

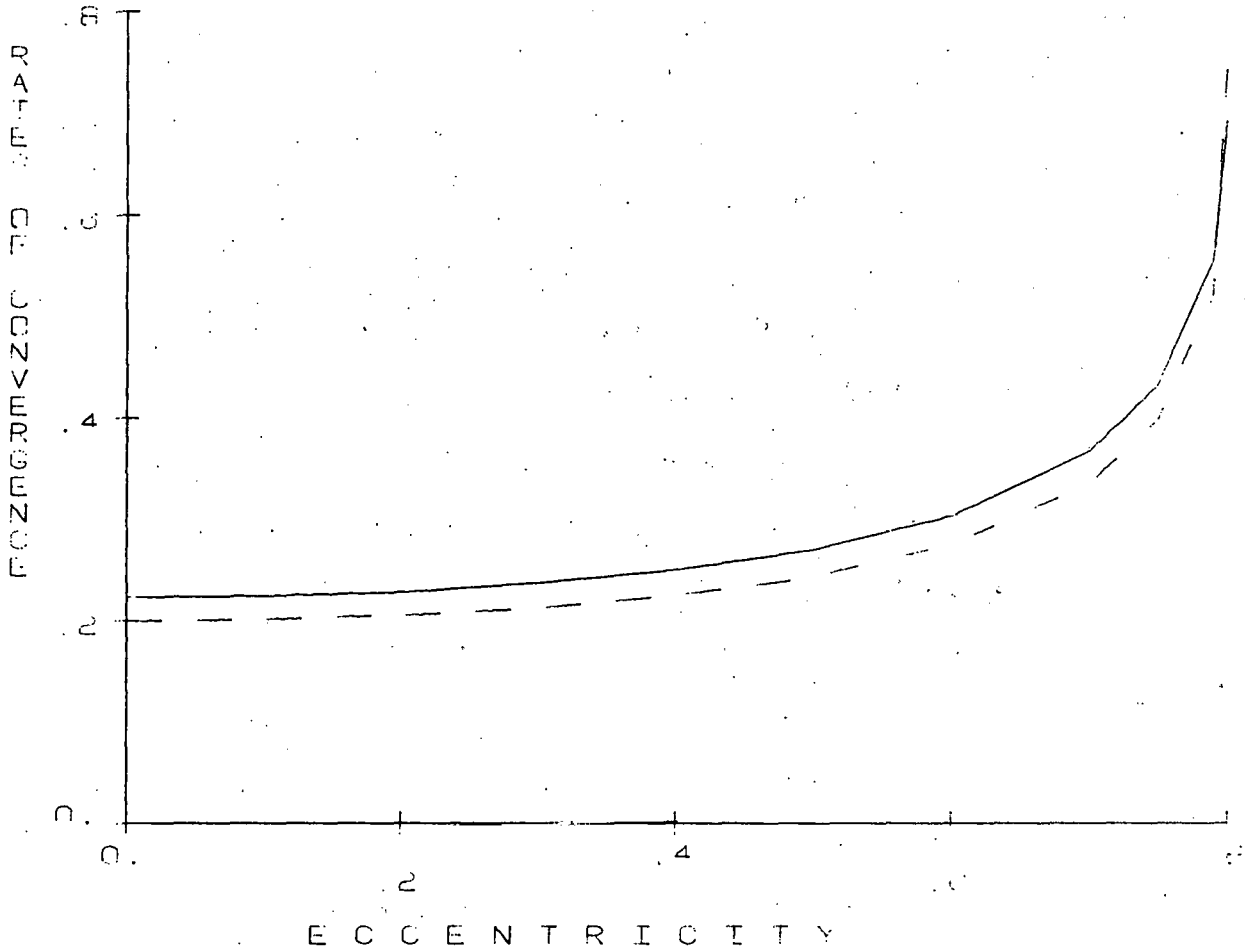


Figure 4.

5.2.

We shall compare in the following experiment the method of conjugate gradients applied to the problem $A^H Ax = A^H b$ with the iterative Arnoldi algorithm. Consider the block-tridiagonal matrices

$$A = \begin{bmatrix} B & -I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & B \end{bmatrix} \quad \text{with} \quad B = \begin{bmatrix} 4 & a & & \\ b & \ddots & \ddots & \\ & \ddots & \ddots & a \\ & & b & 4 \end{bmatrix}$$

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

These matrices come from a discretization of partial differential equations involving a non-selfadjoint operator (see [12], [18]). When δ is small the matrix A is almost symmetric. The conjugate gradient algorithm was run for the following case: $\delta = 0.01$, B has dimension 10 and A has dimension 200. The right hand side b was set to Af where $f = [1, \ldots, 1]^T$ and the initial vector was chosen randomly. We have compared the results with those obtained with the iterative Arnoldi method using 10 steps per iteration (m = 10) and 20 steps per iteration. The initial vector as well as the right hand side are the same as above. Figure 5 shows in a logarithmic scale the evolution of the error norms obtained for the same total number of steps. Notice that although the total number of steps required to achieve convergence is smaller with Arnoldi's method, the total amount of work required in this example is in favor of the conjugate gradient method because the cost of computing Av is not high. The method of Arnoldi will be appropriate whenever the cost of computing Av dominates all the other costs in each step but this will not always be the case. Figure 5 also shows that when the matrix by vector multiplication is costly, it may be advantageous to choose m as large as possible.

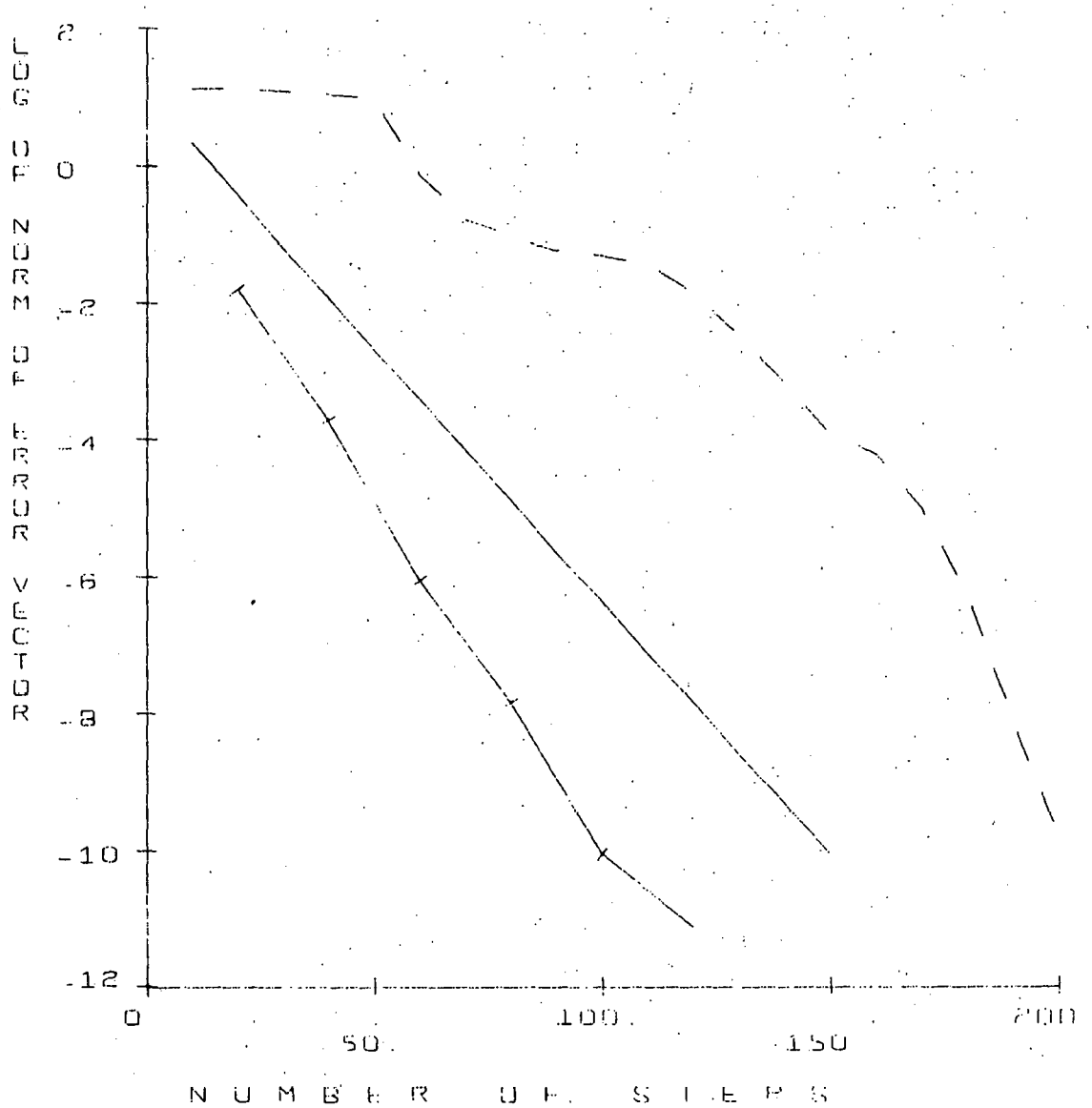


Figure 5. Conjugate gradients for $A^TAx = A^Tb$ (upper curve) and iterative Arnoldi method, p = 10 middle curve, p = 20 lower curve.

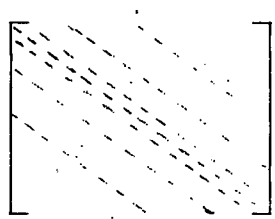
5.3. In the previous example, the matrix treated is nearly symmetric and so the use of the incomplete orthogonalization method without correction is more suitable. Taking $p = 2$, and starting with the same initial vector as in the experiment of 5.2, yielded a rapidly decreasing sequence of residual norm estimates. No restart was necessary and convergence occurred after 90 steps with a residual norm equal to 4.6×10^{-11}. Clearly the amount of work required here is far less than that required by either of the methods compared in 5.2.

5.4. We shall now compare the incomplete orthogonalization methods with and without corrective step on the 100 x 100 block tridiagonal matrix A of §5.2 obtained by taking $\delta = 0.2$. In a first test an iterative method based upon the incomplete orthogonalization algorithm with correction (Algorithm 3.4) was tried. As soon as the estimate $\beta h_{m+1,m} |e_m^H y_m|$ of the residual norm stops decreasing or when the number of steps reaches the maximum number of steps allowed, $m_{max} = 40$, the algorithm is halted, a corrective step is taken and the algorithm is either stopped (if the residual norm is small enough) or restarted. For the present example the algorithm halted first at $m = 20$ and gave a residual norm of 1.8. After the correction step, the residual norm dropped down to 6.2×10^{-3}. In the second iteration the algorithm halted at $m = m_{max} = 40$ and gave the residual norms 9.6×10^{-5} before the correction and 1.14×10^{-6} after.

It is important to mention that, here, the corrective steps necessitate the use of the bidiagonalization algorithm to compute the corrective column s_m, which is usually very expensive.

The results obtained with the incomplete orthogonalization method without correction are by far superior from the point of view of the run times. Algorithm 3.5 was first tested with $p = 2$. At the 1^{st} iteration the residual norms decreased from 7.6 to 1.8 at the 15th step and then a restart was made. At the 2^{nd} iteration the residual norms kept decreasing rapidly to 2.1×10^{-6} at the 60th step. The test with $p = 4$ yielded a steadily decreasing sequence of residual norm estimates and therefore no restart has been necessary. The final residual norm obtained at $m = 60$ was 7.88×10^{-7}.

5.5. Finally we shall describe an experiment on a more difficult example considered in [19]. The runs reported below have been made on a CDC CYBER 175 computer using a word of 60 bits and a mantissa of 48 bits (single precision). The problem $Ax = b$ treated has dimension $N = 1000$ and the nonzero part of A consists in 7 diagonals.

$A =$ 

(The nonzero elements of the 1^{st} row and 1^{st} column of A are A_{11}, A_{12}, $A_{1,10}$, $A_{1,100}$, A_{21}, $A_{10,1}$, $A_{100,1}$.) The problem originated from the simulation of a reservoir, and is known to be badly conditioned. It has been solved in [18] by using Chebychev iteration combined with a

preconditioning technique. The matrix A was first decomposed as A = LU + F where M = LU is an approximate LU decomposition of A provided by one step of the SIP algorithm described in [21]. Then Richardson iteration was run for the problem $M^{-1}Ax = M^{-1}b$, yielding the sequence of approximate solutions

$$x^{(k+1)} = x^{(k)} + t_k M^{-1} r^{(k)} \tag{5.2}$$

where $r^{(k)}$ is the residual $b - Ax^{(k)}$ and t_k is an acceleration parameter. The acceleration parameters were first chosen a priori and as the iteration proceeded, they were periodically adjusted in such a way that the iteration (5.2) matches the (optimal) Chebyshev iteration [12] for the problem $M^{-1}A = M^{-1}b$. After 60 steps the residual norm has decreased by a factor of (see [19]):

$$\|r^{(60)}\|/\|r^{(0)}\| \simeq 2.025 \times 10^{-5}$$

The initial vector x_0 was generated randomly. Note that an important part of the calculations lies in the computation of a few eigenvalues of A, as these are needed for determining the optimal parameters t_k.

Two runs have been made with Algorithm 3.5, the first with p = 2 and the second with p = 4. The same preconditioning matrix M = LU as above has been used. Figure 6 shows the evolution of the residual norms $\|M^{-1}Ax^{(k)} - M^{-1}b\|$ and confirms the remarks ending section 3. In either case, no restart was necessary and at the 60th step, the actual residual norms $\|b - Ax^{(k)}\|$ decreased by a factor of

$$\|r^{(60)}\|/\|r^{(0)}\| \sim 4.44 \times 10^{-7} \text{ for } p = 2$$

and

$$\|r^{(60)}\|/\|r^{(0)}\| \simeq 1.62 \times 10^{-7} \text{ for } p = 4$$

Clearly, here the choice p = 2 is more suitable than p = 4. Note that, with p = 2, each step of Algorithm 3.5 requires about 21 N operations while each step of the first method requires an average of 16.7 N operations per step [19]. Considering that it takes 40 steps for the second method to get the residual norm reduced by a factor of $\|r^{(40)}\|/\|r^{(0)}\| \simeq 3.3 \times 10^{-5}$, it is easily seen that the total number of operations is about 16% less with Algorithm 3.5. Thus, the total numbers of operations are comparable. The first method requires, however, 5 N more memory locations than the second. (These are used to estimate the eigenvalues of $M^{-1}A$.) Let us mention that on another example similar to the present one, the Chebyshev iteration failed to converge, while the I.O.M. gave the solution without any problem with p = 2.

Acknowledgments

The author is indebted to Professor P. Saylor for providing the example treated in section 5.5, and to the referee for his valuable remarks and his careful corrections.

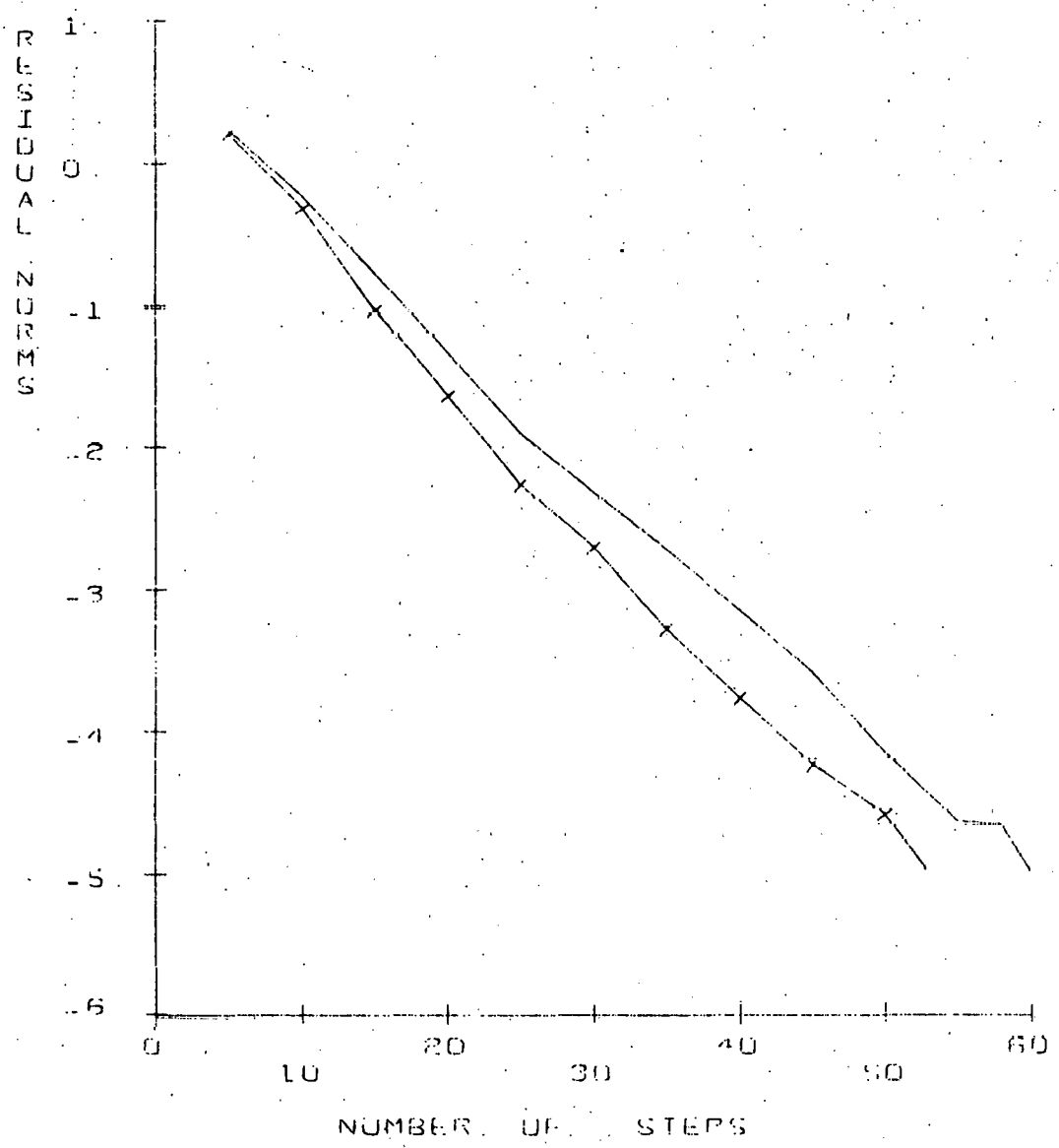


Figure 6. Convergence of algorithm 3.5 on example of §5.5. Upper curve p = 2, lower p = 4.

REFERENCES

[1] ARNOLDI, W.E., The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9, 1951, 17-29.

[2] AXELSSON, O., Solution of linear systems of equations: iterative methods, in Lecture Notes in Mathematics 572, V.A. Barker ed., Springer-Verlag, 1977, 1-51.

[3] BJORK, A. and T. ELFVING, Accelerated projection methods for computing pseudo inverse solutions of systems of linear equations, BIT 19, 1979, 145-163.

[4] CLAYTON, A., Further results on polynomials having least maximum modules over an ellipse in the complex plan, UKAEA Report AEEW-7348, 1963.

[5] CONCUS, P. and G.H. GOLUB, A generalized conjugate gradient method for non-symmetric systems of linear equations, Report STAN-CS-75-535, Computer Science Dept., Stanford University, 1976.

[6] FADDEEV, D.K. and V.N. FADDEEVA, Computational Methods of Linear Algebra. San Francisco: Freeman Company, 1963.

[7] HOUSEHOLDER, A.S., The Theory of Matrices in Numerical Analysis. New York: Blaisdell, 1964.

[8] KRASNOSELSKII, M.A. et al., Approximate Solutions of Operator Equations. Groningen: Wolters-Nordhoof, 1972.

[9] LANCZOS, C.C., Solution of systems of linear equations by minimized iterations, J. Res. N.B.S. 49, 1952, 33-53.

[10] LEBEDEV, V.I., Iterative methods for solution of operator equations with their spectrum on several intervals, Zh. Vychislit. Mat. i Mat. Fiz. 9, 1969, 1247-1252.

[11] LORENTZ, G.G., Approximation of Functions. New York: Holt, Rinehart & Winston, 1966.

[12] MANTEUFFEL, T.A., An iterative method for solving nonsymmetric linear systems with dynamic estimation of parameters, Report UIUCDCS-R-75-758, Dept. Computer Science, U. Illinois U-C, Ph.D. thesis, 1975.

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

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

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

[16] REID, J.K., On the method of conjugate gradients for the solution of large sparse systems of linear equations, in Large Sparse Sets of Linear Equations, J.K. Reid ed., Academic Press, 1971.

[17] RIVLIN, T.J., The Chebyshev Polynomials. New York: J. Wiley & Sons, Inc., 1976.

[18] SAAD, Y., Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, to appear in Lin. Alg. Appl., special issue: Large Scale Matrix Problems.

[19] SAYLOR, P.E., Richardson's iteration with dynamic parameters and the SIP approximate factorization for the solution of the pressure equation, Society of Petroleum Engineers of AIME Fifth Symposium on Reservoir Simulation, Denver, Colorado, 1979, SPE 7688.

[20] STEWART, G.W., Introduction to Matrix Computation. New York: Academic Press, 1973.

[21] STONE, H.L., Iterative solution of implicit approximations of multidimensional partial differential equations, SIAM J. Numer. Anal. 5, 1968, 530-558.

[22] VARGA, R.S., A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials, J. Soc. Indust. Appl. Math. 5, 1957, 39-46.

[23] WRIGLEY, H.E., Accelerating the Jacobi method for solving simultaneous equations by Chebyshev extrapolation when the eigenvalues of the iteration matrix are complex, Computer J. 6, 1963, 169-176.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-81-1047	2.	3. Recipient's Accession No.
4. Title and Subtitle KRYLOV SUBSPACE METHODS FOR SOLVING LARGE UNSYMMETRIC LINEAR SYSTEMS			5. Report Date January 1981 6.
7. Author(s) Y. Saad			8. Performing Organization Rept. No. R-81-1047
9. Performing Organization Name and Address Department of Computer Science University of Illinois 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 Argonne, IL 60439			13. Type of Report & Period Covered technical 14.

15. Supplementary Notes

16. Abstracts

Some algorithms based upon a projection process onto the Krylov subspace $K_m = \mathrm{Span}(r_0, Ar_0, \ldots, A^{m-1}r_0)$ are developed, generalizing the method of conjugate gradients to unsymmetric systems. These methods are extensions of Arnoldi's algorithm for solving eigenvalue problems. The convergence is analyzed in terms of the distance of the solution to the subspace K_m and some error bounds are established showing in particular a similarity with the conjugate gradient method (for symmetric matrices) when the eigenvalues are real. Several numerical experiments are described and discussed.

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

iterative methods
nonsymmetric systems
sparse systems
orthogonal projections
Krylov subspaces

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
unlimited	UNCLASSIFIED	43
	20. Security Class (This Page) UNCLASSIFIED	22. Price