

**NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:**  
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

THE NUMERICAL TREATMENT  
OF LARGE EIGENVALUE PROBLEMS\*

G. W. STEWART

Departments of Computer Science  
and Mathematics  
Carnegie-Mellon University

February, 1974

---

\*This paper was supported in part by the Office of Naval Research  
under Contract N00014-67-A-0128-0018

This paper surveys techniques for calculating eigenvalues and eigenvectors of very large matrices.

## 1. INTRODUCTION

1. The object of this paper is to give a concise review of current methods for solving algebraic eigenvalue problems involving matrices so large that they cannot be maintained in array form in the high speed storage of a computer. Such problems arise in a variety of applications, and I hope that this survey will prove useful to people who must actually solve large eigenvalue problems as well as to the researcher seeking to devise new techniques. Owing to limitations of space, it has been possible to describe the various computational methods and their properties only in barest outline; the reader must go to the literature for more detail. The approach is not historical, and I have often preferred recent references with complete bibliographies in documenting this survey. I have not attempted to compare algorithms for two reasons. First the field is rapidly changing, and too little is known to make flat assertions about the superiority of one algorithm over another. Second the properties of the methods described here depend on many factors, e.g. the type of problem, the computer, the operating system, and even the language in which the method is coded. A method that is superior in one environment may be inferior in another.

2. Many people have encouraged me in writing this survey. I should particularly like to thank Jane Cullum, Gene Golub, Velvel Kahan and Richard Underwood for freely discussing their own work in the area with me. I am also indebted to Axel Ruhe for sending me post haste copies of valuable reports by him and his colleagues at the University of Umeå.

3. It will be assumed that the reader is familiar with the standard theory of the algebraic eigenvalue problem, as well as the better algorithms for computing the eigenvalues of small, dense matrices. For background refer to [22], [61], [62], and [57].

4. Large eigenvalue problems arise in many scientific and engineering problems. As a rule such problems are sparse, that is they have few nonzero elements. The types of matrices involved can be roughly divided into three classes: structured matrices, modifications of structured matrices, and unstructured matrices. We shall be concerned principally with the last class.

5. Highly structured matrices arise from finite difference and finite element approximations to the solution of continuous problems [58], [60], [64]. Much work has been done on the efficient solution of linear systems involving such matrices [9], [14], [21], [60], [64], and these techniques can be applied to the solution of their eigenvalue problems (see [51]).

Another class of structured sparse matrices is the class of band matrices having small band width. A variety of techniques already exist for dealing

with such matrices. The reader is referred to [61], [62].

6. In some problems a highly structured matrix is modified slightly. For example, a tridiagonal matrix may be altered by appending a full row and column. In some cases it is possible to relate the solution of the structured problem to that of the modified problem in a computationally efficient way (see [15] for an example of a diagonal matrix modified by adding a matrix of rank one). Even when this is not possible, the structure of the underlying problem may often be used to economize the iterative methods described below.

7. In many instances, the matrix will have no special structure. This is not to say that its elements will be distributed randomly; rather their distribution reflects an underlying physical problem that is not completely regular. For example, in structural problems the nonzero entries in the matrix represent the connectivity graph of the elements in the structure. Such a matrix may tend to be banded, but with troublesome exceptions. When the exceptions become so numerous that it is impossible to take advantage of the structure of the matrix, recourse must be had to methods that assume little more than sparsity. It is with such methods that this paper is principally concerned.

8. A large part of the literature on large eigenvalue problems is devoted to problems of the form  $Ax = \lambda Bx$  (for background see [28]). Here  $A$  and  $B$  are usually symmetric and  $B$  is positive definite, in which case all the eigenvalues are real. The numerical treatment of such generalized problems is complicated by the fact that there is no analogue of the useful power method, which is the basis of many techniques for the ordinary eigenvalue problem (see Section 5 below).

9. A greater number of the methods described in this paper apply only to symmetric matrices, or at least have not been extended to nonsymmetric matrices. There are two reasons for this. First, the nonsymmetric eigenvalue problem is considerably more difficult than the symmetric eigenvalue problem. Even in the small dense case, the nonsymmetric problem raises points that continue to trouble specialists (e.g., what should one compute in the place of eigenvectors when the matrix is defective). The second reason is that many of the large eigenvalue problems that arise naturally are symmetric, which has encouraged a great deal of work in the area.

10. In connection with what it is possible to compute, it is useful to distinguish between large problems and gigantic problems. We have implicitly defined a large problem as one in which the order  $n$  is such that  $n^2$  exceeds the high speed storage capacity of the computer in question (Section 1.1). If this excess is not too great, it is reasonable to ask for the full complement of eigenvalues and eigenvectors, perhaps computed at great expense. A gigantic problem is one in which the high speed storage capacity of the computer is but a small multiple of  $n$ . In

\* This paper was supported in part by the Office of Naval Research under Contract N00014-67-A-0128-0018.

this case, it is not reasonable to ask for a complete eigensystem, since the array of eigenvectors will not in general be sparse and cannot be easily manipulated. Moreover, the methods best suited to gigantic problems give at most a few eigenvectors usually corresponding to extreme eigenvalues (eigenvalues lying at either end of the spectrum). It is fortunate that in many problems all that is required is a few extreme eigenvalues and their eigenvectors.

## 2. SPARSE MATRIX TECHNOLOGY

1. In recent years a substantial amount of work has been devoted to computing LU decompositions of large sparse matrices as a preliminary to solving linear systems. This is not the place to survey this work; however, some of the techniques developed are applicable to the large sparse eigenvalue problem, and we shall mention them in this section. For further details and references the reader is referred to [47], [43], and [63].

2. In order to operate with a large sparse matrix, it must be represented in the computer, preferably in its high speed storage. Even a gigantic problem, if it is sufficiently structured or sparse, may be so represented. For example, the five point difference approximation to the  $\Delta$  operator over an equally spaced mesh yields a matrix whose nonzero elements have the values 4 and -1 and are regularly distributed. Hence an element of this matrix can be retrieved simply from a knowledge of its position in the matrix. As another example, very large band matrices can be represented by recording their nonzero diagonals in linear arrays [62]. Matrices that depart slightly from an otherwise representable matrix may be represented by recording the modification in a suitable manner. For example the sum  $A + uv^T$  of a tridiagonal matrix  $A$  and a rank one matrix  $uv^T$  will not in general be sparse. But it can be represented by recording  $A$  and the vectors  $u$  and  $v$ .

3. If an unstructured matrix is sufficiently sparse, it may also be represented in the high speed storage of a computer by recording only its nonzero elements along with information enabling one to retrieve these elements. How this is done will depend on what is to be done with or to the matrix. If, for example, all that is required is to form a matrix-vector product (cf. Section 5 below), a simple packed row representation of the matrix will suffice. If arbitrary matrix elements are to be retrieved or if the matrix itself is to be altered, more sophisticated representations will be needed. Various kinds of linked lists are in common use [27], [7], [8], [18]. A promising alternative is the arc graph representation [44], [45].

4. When the matrix involved is dense or when the number of its nonzero elements exceeds the capacity of the high speed storage of the computer, the matrix must be retained on a slower backing store and its elements brought into high speed storage by blocks. Again how this is done will depend on the problem to be solved, and little can be said in general. For a careful analysis of paging strategies in solving linear systems see [46]. The behavior of some standard matrix algorithms on systems with automatic paging has been discussed in [31], [32].

5. In some cases it is necessary to solve repeatedly problems in which the distribution of nonzero elements of the matrix does not change, although the value of the elements themselves may be changed. In such cases the matrix may often be preprocessed to good effect. If the matrix is to be manipulated, a preprocessing algorithm can be used to arrange the computations so that the fill-in of new nonzero elements is reduced [19], [54]. Moreover if an algorithm is to be used whose data accesses depends only on the structure of the nonzero elements (e.g. Gaussian elimination), a preprocessing algorithm can

actually compile efficient code that takes advantage of the known distribution of the zeros [18], [10]. Both preprocessing techniques should find application to the quasi-direct eigenvalue techniques to be discussed next.

## 3. QUASI-DIRECT METHODS

1. Since the general algebraic eigenvalue problem is equivalent to the solution of a polynomial equation, all numerical methods for its solution must be iterative. However, most of the better algorithms for dense small eigenvalue problems initially reduce the matrix to some compact form as a preliminary to iteration. These direct reductions start with the given matrix  $A_0$  and produce a sequence of similarity transformations

$$A_{k+1} = P_k^{-1} A P_k \quad (3.1)$$

in such a way that some  $A_k$  (usually  $A_1$ ) has a convenient form (usually tridiagonal or Hessenberg) that makes iterating for the eigenvalues easy. We shall call such a combination of direct reduction and iteration a quasi-direct method. For details on the methods mentioned in this section, the reader is referred to [61].

2. When a large sparse matrix can be represented in the high speed storage of a computer, it is possible to attempt to reduce it to compact form by one of the kind of reductions indicated above, say Householder's reduction of a symmetric matrix to tridiagonal form or the reduction by elementary transformations to Hessenberg form. The chief enemy of such reductions is fill-in, which is likely to be greater for eigenvalue problems than for linear systems for two reasons. First the similarity transformations in (3.1) are more complicated than the one-sided transformations used to solve linear systems. Second, for symmetric matrices orthogonal transformations must be used to preserve symmetry, and these transformations tend to create more fill-in than their nonorthogonal counterparts. To mitigate fill-in one may attempt during the course of the reduction to pivot so that fill-in is reduced [59]. The author knows of no algorithm for examining the structure of a matrix to minimize fill-in, as is often done for linear systems (cf. Section 2.5).

3. For symmetric matrices a promising algorithm is the Lanczos reduction with reorthogonalization as implemented in [16] (for a general discussion of the Lanczos algorithm see Section 5.5). This algorithm generates a sequence of orthonormal vectors  $x_1, x_2, \dots, x_n$  such that if  $x_k = (x_{k1}, \dots, x_{kn})$  then  $T_k = X_k^T A X_k$  is tridiagonal. The eigenvalues of  $A$  are the same eigenvalues of  $T_k$ , and can be easily computed. The method has the advantage that it requires only that one be able to form the vector  $Ax$  for any given vector  $x$ . Thus fill in is not a problem for the method, and it can even be used on dense matrices that are too large to fit in the high speed storage of the computer. The main disadvantage of the algorithm is that the vectors  $x_1, x_2, \dots, x_k$  must all be used to generate  $x_{k+1}$ . However, these vectors may be kept on a backing store and brought in as needed (it should be noted that the  $x_k$ 's are not changed from step to step so there is no need for messy and expensive updating on the backing store).

The method can also be used on the symmetric  $Ax = \lambda Bx$  problem whenever  $B$  can be cheaply factored in the form  $B = LL^T$  with  $L$  lower triangular. For then the required eigenvalues are the eigenvalues of the symmetric matrix  $C = L^{-1}AL^{-T}$ . The product  $Cx$  can be computed inexpensively by solving triangular systems and multiplying by  $A$  without ever explicitly forming  $C$ , which will in general not be sparse.

There should be an analogous process for the Hessenberg algorithm to reduce a nonsymmetric matrix to Hessenberg form, but, since the final matrix will

not be sparse, it may be of limited value. Whether the general Lanczos biorthogonalization algorithm [67] can be made stable by some kind of reorthogonalization is an open question.

4. Once a matrix has been reduced to a condensed form, its eigenvalues must be evaluated. For small dense matrices this is usually done by the QR algorithm. For large matrices the QR algorithm has some disadvantages. First the transformations used by the algorithm tend to destroy sparsity, so that it is not a good choice for a sparse Hessenberg matrix. Second, although the algorithm is satisfactory enough for finding all the eigenvalues of a symmetric tridiagonal matrix, if only a few eigenvalues are required there are better ways (see Section 3.6). Finally, if eigenvectors are to be computed one must save either the original matrix or the QR transformations.

5. If the matrix  $A$  is Hessenberg, the function  $f(\lambda) = \det(A - \lambda I)$  and its derivatives can be cheaply evaluated by Hyman's method [61]. This suggests using an iterative method to find the zeros of  $f$ , which of course are the eigenvalues of  $A$ . This is perhaps the best way of getting at the eigenvalues of a sparse Hessenberg matrix, and various strategies and iterative techniques have been treated in detail in [61] and [38].

6. If only a few of the eigenvalues of a symmetric tridiagonal matrix  $A$  are to be calculated, it is best done by the method of Sturm sequences combined with linear interpolation. The method is based on the following observation. Let  $f_k(\lambda) = \det(A^{[k]} - \lambda I)$ , where  $A^{[k]}$  is the leading principal submatrix of order  $k$  of the symmetric matrix  $A$ . The number of agreements in sign between consecutive members of the sequence  $f_1(\lambda), f_2(\lambda), \dots, f_n(\lambda)$  is the number of eigenvalues of  $A$  that are strictly greater than  $\lambda$ . Since, in our case,  $A$  is tridiagonal, the numbers  $f_k(\lambda)$  can be easily evaluated and the number of eigenvalues in any interval counted. By bisecting intervals, arbitrarily small intervals containing an eigenvalue can be found. Since  $f_n(\lambda) = \det(A - \lambda I)$  these values can be used in an iterative scheme, such as the secant method, to accelerate convergence. The general method, which is described in [41], has the additional advantage that it can locate a few eigenvalues lying in a specified interval.

The method can also be applied to the symmetric  $Ax = \lambda Bx$  problem, by setting  $f_k(\lambda) = \det(A^{[k]} - \lambda B^{[k]})$ . If  $A$  and  $B$  are band matrices, the  $f_k$  can be evaluated with not too much computational expense [41], [17].

7. It should be noted that the techniques of Section 3.5-6 often can be applied to matrices that are simple modifications of a highly structured matrix. For example, the determinant of the matrix  $A + uv^T$  can be easily calculated from the determinant of  $A$ :  $\det(A + uv^T) = \det(A) + v^T u$ . Hence if  $A$  is upper Hessenberg,  $\det(A + uv^T - \lambda I)$  can be cheaply evaluated.

8. If it is desired, eigenvectors of the condensed matrix may be computed by the inverse power method and then transformed back to eigenvectors of the original matrix. Since the condensed matrix  $A$  is usually Hessenberg or tridiagonal, the linear systems generated by the inverse power method may be cheaply solved, although extra storage must be provided to save  $A$ , which must not be destroyed. Care must be taken with multiple and defective eigenvalues [42]. After the eigenvectors of the condensed matrix have been computed, they may be transformed back using the transformations  $P_k$  of (3.1), which must be stored for this purpose. Since the information required to recover  $P_k$  will usually require at most  $n$  locations, the  $P_k$  may easily be saved on a backing store.

#### 4. METHODS THAT REDUCE THE RAYLEIGH QUOTIENT

1. The methods of this section apply to the  $Ax = \lambda Bx$  problem, where  $A$  is symmetric and  $B$  is positive definite. They are based on the fact that the Rayleigh quotient  $\rho(x) = x^T A x / x^T B x$  has as its minimum the smallest eigenvalue  $\lambda_1$  of the problem. This minimum is attained when  $x$  is an eigenvector corresponding to  $\lambda_1$ .

The idea is to generate a sequence of vectors  $\{x_i\}$  related by  $x_{i+1} = x_i + \alpha_i p_i$ , where  $p_i$  is a correction vector and  $\alpha_i$  is a scalar. The  $p_i$  and  $\alpha_i$  are chosen in such a way that the sequence  $\rho(x_i)$  decreases, and hence approaches a limit.

By taking  $B = I$ , the methods of this section yield methods for solving the usual symmetric eigenvalue problem. Moreover if it is attempted to increase rather than reduce  $\rho(x_i)$ , they yield analogous methods for finding dominant eigenvalues.

2. The simplest of the methods is the method of relaxation, in which the vector  $p_i$  is chosen to be one of the coordinate vectors, say  $e_k$ , so that only the  $k$ th component of  $x_i$  is altered. The scalar  $\alpha_i$  is chosen so that  $x_{i+1}$  satisfies the equation  $(a_k^T - \rho(x_i) b_k^T) x_{i+1} = 0$ , where  $a_k^T$  and  $b_k^T$  are the  $k$ th rows of  $A$  and  $B$ . This corresponds to one step of the usual relaxation method for solving the homogeneous system  $(A - \rho(x_i) B)x = 0$ , and the method is at least as old as the widespread use of relaxation techniques [40]. In versions suited for computers the directions  $p_i$  are chosen cyclically from among the coordinate vectors  $e_i$ , usually in the order  $e_1, e_2, \dots, e_n, e_1, \dots$ .

3. The simplicity of the relaxation method particularly recommends it. If a backing store is necessary, the rows of  $A$  can be brought in one by one, or in blocks. The Rayleigh quotient can be updated at each stage, so that it is not necessary to compute  $\rho(x_i)$  *ab initio* for each  $x_i$  [33]. Other quantities necessary to the iteration can be treated similarly [25].

4. It has been known for some time [6] that if  $\rho(x_i) < \min(a_i/b_{ii})$ , then the sequence  $\{\rho(x_i)\}$  decreases, and since it is bounded below by  $\lambda_1$ , it must converge to a value  $\lambda$ . The residual vectors  $(A - \rho(x_i) B)x_i$  also approach zero [50]. It follows that  $\lambda$  must be an eigenvalue. Moreover, if  $\lambda$  is simple, the sequence  $\{x_i\}$  approaches an eigenvector when it is suitably scaled. However, unless  $\rho(x_i)$  is less than the second largest eigenvalue, it cannot be guaranteed that  $\lambda = \lambda_1$  (but see Section 4.7 below).

5. The relaxation method can be overrelaxed in the usual way by taking  $\alpha_i$  greater than necessary to solve the equation  $(a_k^T - \rho(x_i) b_k^T) x_{i+1} = 0$ . In some cases this will improve convergence. If the problem has sufficient structure, it is possible to carry over some of the theory of successive overrelaxation for linear systems [50], [51].

6. Closely related to the relaxation method is the method of "coordinate-relaxation" [11]. It differs from the relaxation method in that the parameter  $\alpha_i$  is chosen so that  $\rho(x_{i+1})$  is minimized. The value  $\alpha_i$  can be obtained as one of the roots of a quadratic equation whose coefficients can be cheaply generated. Empirical evidence indicates that the method of coordinate relaxation is superior to the relaxation method in the early stages of the iteration [55], [50].

7. The method of coordinate relaxation has been thoroughly analyzed in [25], where it is shown that the  $\alpha_i$  approach the values used in the relaxation method. This means that in the limit the two methods are the same. Moreover, there is given a

criterion for when to shift over to the less expensive relaxation method. Since the method of coordinate relaxation is a descent method, the  $\rho(x_i)$  form a decreasing sequence, which approaches an eigenvalue  $\lambda$ . It cannot be guaranteed that  $\lambda = \lambda_1$ , unless  $\rho(x_i)$  is less than the second largest eigenvalue. However, geometric considerations indicate that convergence to an eigenvalue other than  $\lambda_1$  is unlikely, and when it happens the sequence  $\{x_i\}$  will almost certainly be unbounded, which it is not when  $\rho(x_i) \rightarrow \lambda_1$ . These comments on convergence apply also to the relaxation method.

8. It has also been proposed to apply the method of conjugate gradients to minimize  $\rho(x)$  [2], [12]. This is practical since gradient  $\rho'(x)$  is proportional to the residual  $(A - \rho(x)B)x$  and may be easily calculated. There results a scheme in which the sequences  $\{x_i\}$  and  $\{p_i\}$  are generated concurrently:  $x_{i+1} = x_i + \alpha_i p_i$ ,  $p_{i+1} = \rho'(x_{i+1}) + \beta_i p_i$ . Here  $\alpha_i$  is chosen to minimize  $\rho(x_{i+1})$ . The  $\beta_i$  are chosen to maintain conjugacy among the directions  $p_i$ , and there are several possible formulas (see [48]). One approach is to choose  $\beta_i$  and  $\alpha_i$  simultaneously to minimize  $\rho(x_{i+1})$  [13].

The theory of conjugate gradient methods applied to the Rayleigh quotient is not highly developed. Limited experiment [51] that it is somewhat better than relaxation or coordinate relaxation (here one conjugate gradient step is compared with a complete cycle of relaxation).

9. A natural generalization of the relaxation methods is to write  $A - \rho(x_i)B$  in the form  $M_i - N_i$ , where  $M_i$  is easy to invert. The vector  $x_{i+1}$  is then given by  $x_{i+1} = M_i^{-1} N_i x_i$ . Methods of this kind have been analyzed in [50], [51].

## 5. METHODS BASED ON KRYLOV SEQUENCES

1. The methods of this section use a Krylov sequence of the form  $x, Ax, A^2x, \dots$  to find an approximate eigenvector of  $A$ . Since a Krylov sequence can be generated by sequential vector matrix multiplications, the methods are well suited to large or gigantic problems. Most, but not all of the methods apply only to symmetric matrices (for another survey of these methods see [48]).

2. One of the oldest methods for finding an eigenvector is the power method. It is based on the fact that the members of the Krylov sequence will tend toward an eigenvector of  $A$ . Specifically if the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  of  $A$  satisfy  $|\lambda_1| > |\lambda_2| \geq |\lambda_i|$  ( $i > 2$ ), then under mild restrictions on  $x$  the vectors  $A^k x$  tend, when suitably scaled, toward  $u_1$ , the eigenvector corresponding to  $\lambda_1$ . The convergence is essentially linear with ratio  $|\lambda_2/\lambda_1|$ .

3. The convergence of the power method can sometimes be accelerated by working with the matrix  $A - pI$ , where the scalar  $p$  is chosen to enhance the dominance of  $\lambda_1$ . When  $A$  is symmetric with eigenvalues  $\lambda_1 > \lambda_2 > \dots > \lambda_n$ , the optimal value of  $p$  is  $(\lambda_2 + \lambda_n)/2$ . Of course this value of  $p$  can be computed only if estimates are available for  $\lambda_2$  and  $\lambda_n$  (if  $A$  is positive definite then  $\lambda_n$  can be estimated by zero). In general the choice of  $p$  is something of an art.

4. In the symmetric case, where an interval  $[\alpha, \beta]$  enclosing  $[\lambda_1, \lambda_2]$  is known, the vectors  $y_k = C_{[\alpha, \beta]}^{(k)}(x)$ , where  $C_{[\alpha, \beta]}^{(k)}$  is the  $k$ th Chebyshev polynomial on the interval  $[\alpha, \beta]$ , will in general converge faster than the vectors  $A^k x$  of the power method. The vectors  $y_k$  can be generated by the three term recurrence

$$x_{k+1} = \frac{2}{\beta - \alpha} Ax_k - \frac{\beta + \alpha}{\beta - \alpha} x_k - x_{k-1}$$

which is suitable for large problems. The convergence is at least as fast as the approach of  $1/C_{[\alpha, \beta]}^{(k)}(\lambda_1)$  to zero. This method has been used in [53].

5. An extremely important algorithm for symmetric matrices is the Lanczos algorithm [29]. Here the elements of the Krylov sequence are orthogonalized to yield a sequence  $x_1, x_2, \dots$  of orthonormal vectors. The vectors  $x_k$  can be shown to satisfy a three term recurrence of the form

$$\gamma_{k+1} x_{k+1} = Ax_k - \alpha_k x_k - \beta_k x_{k-1}, \quad (5.1)$$

where  $\alpha_k$  and  $\beta_k$  are chosen to make  $x_{k+1}$  orthogonal to  $x_k$  and  $x_{k-1}$  and  $\gamma_{k+1}$  is chosen so that  $x_{k+1}$  has length unity. It follows from (5.1) that if  $X_k = (x_1, x_2, \dots, x_k)$ , then  $X_k^T A X_k = \text{tridiag}(\beta_k, \alpha_k, \gamma_k) \equiv T_k$ .

6. The matrix  $T_k$  is similar to  $A$ , and consequently its eigenvalues  $\mu_k$  coincide with those of  $A$  (cf. Section 3.3). However some of the eigenvalues of an intermediate  $T_k$  may very closely approximate eigenvalues of  $A$  [26], [36], even when  $k$  is very much less than  $n$ . This is particularly true of isolated eigenvalues and eigenvalues at the extremity of the spectrum. If  $\mu$  is such an eigenvalue of  $T_k$  and  $z$  is its eigenvector, then  $X_k z$  may approximate an eigenvalue of  $A$ . It is an unfortunate aspect of the algorithm that in order to compute eigenvectors in this way the vectors  $x_k$  must either be saved or regenerated.

7. There are several mathematically equivalent ways of computing  $\alpha, \beta, \gamma$  in (5.1), and the numerical properties will depend on which choice is made. An analysis in [36] shows that one should compute  $\alpha_k$  according to either of the formulas  $\alpha_k = x_k^T A x_k$  or  $\alpha_k = x_k^T (Ax_k - \beta_k x_{k-1})$ , and that  $\beta_k$  should be taken equal to  $\gamma_k = \sqrt{x_k^T x_k}$ .

8. The curse of the Lanczos algorithm is the loss of orthogonality and even independence among the vectors  $x_k$ . For example, the matrices  $T_k$  cannot have multiple eigenvalues, so that if  $A$  has a multiple eigenvalue and the process must break down prematurely. Numerically this will be signaled by severe cancellation in the computation of some  $Ax_k - \alpha_k x_k - \beta_k x_{k-1}$ , after which the computed  $x_{k+1}$  will be far from orthogonal to its predecessors. One cure is to reorthogonalize  $x_{k+1}$  with respect to the previous vectors (see [35], [16] and Section 3.3), but this is expensive.

In practice the algorithm without reorthogonalization produces accurate eigenvalue approximations. However, as the process degenerates it tends to start over and produce multiple approximations to the same eigenvalue. This need not be regarded as a defect, since it permits the method to compute multiple eigenvalues, which it could not otherwise do. In order to avoid the acceptance of spurious eigenvalues, all the eigenvectors corresponding to a cluster of eigenvalues can be computed as in Section 5.6, those with large residuals rejected, and a linearly independent subset of the remainder accepted as true eigenvectors of  $A$  [36].

9. There is a version of the Lanczos algorithm for nonsymmetric matrices which requires in addition to the Krylov sequence of Section 5.1 another Krylov sequence in  $A^T$ . Its theoretical and numerical properties are not as well understood as those of the symmetric Lanczos algorithm (for a discussion see [61]).

10. The methods discussed so far in this section

have the defect that the user has no choice of what eigenvalues he gets. The methods now to be discussed have the property that they take an accurate approximation  $\lambda$  to an eigenvalue  $\lambda_1$  of the symmetric matrix  $A$  and attempt to find a linear combination  $y_k = \alpha_0 x + \alpha_1 Ax + \dots + \alpha_k A^k x_k$  of the Krylov sequence that is a good approximation to the eigenvector  $u_1$  corresponding to  $\lambda_1$ . Of course it does not hurt if  $x$  is already a fair approximation to  $u_1$ .

11. The author has considered but not analyzed a method in which  $\alpha_1 = 1$  and the remaining  $\alpha$ 's are chosen to minimize  $\|(A-\lambda I)y_k\|_2$ . This least squares can be solved by orthogonalizing the sequence  $A(A-\lambda I)x, A^2(A-\lambda I)x, \dots, A^k(A-\lambda I)x$ , which gives a three term recurrence ala the Lanczos algorithm. Alternatively the vectors  $Ax, A^2x, \dots, A^kx$  may be orthogonalized with respect to the positive definite matrix  $(A-\lambda I)^2$ , again giving a three term recurrence.

12. In a slightly different approach, Hufford [23] chooses  $y_k$  so that  $\|(A-\rho(y_k)I)y_k\|_2/\|y_k\|_2$  is approximately minimized. Again the vector  $y_k$  can be computed via a three term orthogonalization. A criterion for selecting  $k$  is given, and it is shown that the method has nice convergence properties when  $\lambda$  is sufficiently near  $\lambda_1$ .

13. Given an approximation  $\lambda$  to  $\lambda_1$ , the inverse power method takes as an approximation to  $u_1$  the solution of the system  $(A-\lambda I)y = x$ . In [49] it is proposed to use the method of conjugate gradients [20] to solve this system. The resulting approximations  $y_k$  are linear combinations of the Krylov sequence as described in Section 5.10. Some care must be taken in starting the process, since the method breaks down if  $\lambda = \rho(x)$ . Numerical experiments suggest that the method is quite powerful. In this connection the version of the method of conjugate gradients in [37] may be superior to the usual one.

14. For completeness we mention that the method of Section 4.8 in which the method of conjugate gradients is used to minimize  $\rho(x)$  also expresses an approximate eigenvector as a linear combination of the members of a Krylov sequence. However, this method generally converges to the smallest eigenvalue of  $A$ .

## 6. CALCULATING SEVERAL EIGENVALUES AND EIGENVECTORS

1. Some of the iterative methods discussed in the last two sections give only one eigenvalue and eigenvector at a time, whereas in some applications several are needed. In this section some methods for obtaining more than one eigenvalue are discussed.

2. When an approximate eigenvalue  $\lambda_1$  and its corresponding approximate eigenvector  $x_1$  have been found by an iterative method, it is natural to modify the matrix so that the iterative method cannot converge to  $x_1$ . Such modifications are called deflations. Deflation methods have been treated in detail in [61], and here we mention only one for illustrative purposes.

Suppose that  $A$  is symmetric, and  $x_1$  is an exact eigenvector of norm unity corresponding to the eigenvalue  $\lambda_1$ . Then, since the eigenvectors of  $A$  are orthogonal, it is easily verified that the matrix  $A_1 = (I-x_1x_1^T)A(I-x_1x_1^T)$  has an eigenvalue zero corresponding to the eigenvector  $x_1$ . The remaining eigenvalues and eigenvectors are undisturbed. Moreover, if  $x$  is orthogonal to  $x_1$ , then so is the Krylov sequence  $x, Ax, A^2x, \dots$ , so that the methods of Section 5 cannot converge to  $x_1$ .

In practice  $x_1$  will only approximate an eigenvector of  $A$ . However, it can be shown that the error induced by the inaccuracy of  $x_1$  is equivalent to perturbing  $A$  by a matrix of norm equal to  $\|Ax_1 - \rho(x_1)x_1\|_2$  [57]. Thus  $x_1$  need not even be an accurate eigenvector, as long as it has a small residual.

3. A frequently encountered objection to deflation techniques is that they destroy sparsity. This is in general true if the deflated matrix is formed explicitly. However, in many cases one can work with the matrix in a factored form. For example, if it is desired to compute the vector  $A_1x$ , where  $A_1$  is the deflated matrix of Section 6.2, one can compute sequentially  $y_n = (I-x_1x_1^T)x = x - (x_1^Tx)x_1$ ,  $z = Ay$ , and  $A_1z = z - (x_1^Tz)x_1$ , so that instead of the matrix  $A_1$  all that is required is the vector  $x_1$  and the matrix  $A$ . In fact if  $x$  is orthogonal to  $x_1$ , then  $y = x$  and the first step can be skipped.

The practical implementation of a deflation technique, will depend on the technique and the iterative method to be used. For an example of deflation in relaxation methods see [55].

4. The method of simultaneous iteration [1], [4], [52], [53], [56] for finding several dominant eigenvectors of a matrix is based on the following observation. Suppose that  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n| > |\lambda_{n+1}| \geq \dots$ , and let  $U$  denote the invariant subspace corresponding to  $\lambda_1, \dots, \lambda_r$ . Let the  $n \times r$  matrix  $X$  be given. Then under mild restrictions on  $X$ , the space spanned by the columns of  $A^k X$  tends toward  $U$  [39]. Thus appropriate linear combinations of the columns of  $A^k X$  tend toward eigenvectors of  $A$ .

In practice all of the columns of  $A^k X$  will tend toward the dominant eigenvector of  $A$ , and the information about the other eigenvectors of  $A$  will be lost. To cure this problem, matrices  $X_k$  having the same column space as  $A$  are generated according to the formula  $X_{k+1} = AX_k R_k$ , where  $R_k$  is chosen to keep the columns of  $X_{k+1}$  strongly independent. The matrix  $R_k$  is commonly chosen to be the upper triangular matrix that makes the columns of  $X_{k+1}$  orthonormal. If  $A$  is symmetric, then the  $i$ th column  $x_i^{(k)}$  of  $X_k$  will usually tend toward the eigenvector corresponding to  $\lambda_k$  at a rate proportional to the  $k$ th power of  $\max\{|\lambda_k/\lambda_{k-1}|, |\lambda_{k-1}/\lambda_{k-2}|\}$ . Thus for symmetric  $A$ , the method of simultaneous iteration obtains several eigenvectors at a time.

5. It is evident that the convergence of  $x_i^{(k)}$  will be slow if  $\lambda_i$  is poorly separated from its neighbors. On the other hand, if  $|\lambda_i| \gg |\lambda_{i+1}|$ , the space spanned by the columns of  $X_k$  will quickly contain a good approximation to the  $i$ th eigenvector of  $A$ . This good approximation can be retrieved by the following Rayleigh-Ritz device. Form the symmetric matrix  $B_k = X_k^T A X_k$  and diagonalize it by a unitary transformation:  $M_k = P_k^T B_k P_k$  (this last step involves the solution of a comparatively small  $r \times r$  symmetric eigenvalue problem). Then the columns of  $Z_k = X_k P_k$  will also converge to eigenvectors of  $A$ , but this time at rates proportional to the  $k$ th power of  $|\lambda_{i+1}/\lambda_i|$ . The diagonal elements of  $M$  are Rayleigh quotients of the  $y$ 's and provide good approximations to the eigenvalues. A program incorporating this acceleration appears in [53].

6. In [5] the above technique is generalized to nonsymmetric matrices. Here another sequence  $Y_k$  is formed from the matrices  $(A^k)^k Y_k$ . The matrices  $X_k$  and  $Y_k$  are required to satisfy the biorthogonality relations  $Y_k^T X_k = I$ . Suppose that  $B_k = Y_k^T A X_k$  can be diagonalized:  $P_k^{-1} B_k P_k = M_k$ . Then under suitable conditions the column of  $X_k P_k$  and  $Y_k P_k^T$  approach left and right eigenvectors.

7. A different approach to the nonsymmetric case has been taken by the author in an effort to avoid the problems posed by degeneracies in the matrix. The matrices  $X_k$  are formed as in the symmetric case, and under suitable conditions  $x_i^{(k)}$  approaches the  $k$ th Schur vector, that is the vector obtained by orthogonalizing the eigenvector  $u_i$  with respect to  $u_1, u_2, \dots, u_{i-1}$ . The convergence can be accelerated by forming  $B_k$  as in Section 6.5 and finding a unitary  $P_k$  such that  $M_k = P_k^T B_k P_k$  is upper triangular with its eigenvalues ordered in descending order of

modulus. The columns of  $Z_k = X_k P_k$  will converge faster than those of  $X_k$ . It should be noted in many applications a knowledge of the Schur vectors is all that is required

8. A promising area of research is the use of block Lanczos schemes. These are obtained by orthogonalizing the block Krylov sequence  $X, AX, A^2X, \dots$ , where  $X$  is an  $n \times p$  matrix. One obtains a three term recurrence of the form

$$X_{k+1} C_{k+1} = AX_k - X_k D_k - X_{k-1} B_k.$$

If  $C_{k+1}$  is chosen to be the lower triangular matrix that orthogonalizes the columns of  $X_{k+1}$ , then  $D_k = X_k^T A X_k$  and  $B_k = C_k^T$ . Moreover, the matrix  $T_k = (X_1, \dots, X_k)^T A (X_1, \dots, X_k)$  is block tridiagonal with entries  $B_k, D_k$ , and  $C_k$ . Eigenvalues and eigenvectors can be found as in the usual Lanczos method; however, eigenvalues of multiplicity less than  $r$  may now be found without difficulty.

Work in this area is being carried out by Gene Golub and Richard Underwood (Stanford University) and by June Cullum (IBM Research, Yorktown Heights) and Velvel Kahan (UC, Berkeley).

## 7. THE USE OF AN INVERSE

1. Some of the standard techniques for dense eigenvalue problems - the inverse power method, for example - require one to be able to form the vector  $x = (A - \lambda I)^{-1} b$  for any vector  $b$ . This is of course equivalent to solving the system  $(A - \lambda I)x = b$ . If  $A$  is sparse such systems can often be solved with reasonable efficiency via the sparse matrix techniques discussed in Section 2. This section is devoted to examining the consequences of this fact for the large eigenvalue problem.

2. The transformation  $A \rightarrow (A - \lambda I)^{-1}$  transforms the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  of  $A$  into the eigenvalues  $(\lambda_1 - \lambda)^{-1}, (\lambda_2 - \lambda)^{-1}, \dots, (\lambda_n - \lambda)^{-1}$  of  $(A - \lambda I)^{-1}$  without altering the eigenvectors. Thus all of the eigenvalues of  $A$  near  $\lambda$  become extreme eigenvalues of  $(A - \lambda I)^{-1}$ . In particular if  $\lambda$  is very near an eigenvalue of  $A$ , the power method with  $(A - \lambda I)^{-1}$  will converge very swiftly to the corresponding eigenvector of  $A$ . This is the inverse power method [42]. Alternatively one can employ the Lanczos algorithm (Section 5.5), or simultaneous iteration techniques (Section 6.4) to find several eigenvalues near  $\lambda$  and their corresponding eigenvectors.

3. The methods mentioned in Section 7.2 share the property that the shift  $\lambda$  is fixed. Several methods, such as the well known Rayleigh quotient method [34], achieve high order local convergence by varying  $\lambda$ . Such methods have the drawback that a new LU decomposition of  $A - \lambda I$  must be calculated each time  $\lambda$  is changed. However, the sparsity structure of  $A - \lambda I$  does not change with  $\lambda$ , and considerable work may be saved by the preprocessing techniques mentioned in Section 2.5. Another difficulty with these methods is that their convergence to a specific eigenvalue cannot be guaranteed unless an accurate starting approximation is given.

4. The generalized eigenvalue problem  $Ax = \lambda Bx$  is equivalent to the ordinary eigenvalue problem  $B^{-1}Ax = \lambda x$  whenever  $B$  is nonsingular. Consequently if systems involving  $B$  can be readily solved, one can apply some of the techniques previously discussed to the matrix  $B^{-1}A$  (n.b.,  $B^{-1}A$  will not in general be formed explicitly). When  $A$  is symmetric and  $B$  is positive definite, symmetry can be preserved by factoring  $B$  into the product  $LL^T$ , where  $L$  is lower triangular, and working with the symmetric matrix  $L^{-1}AL^{-T}$  [30] (cf. Section 3.3).

The inverse power method applied to  $B^{-1}A$ , requires that one solve the symmetric system

$(A - \lambda B)y = Bx$ . Since in general  $A - \lambda B$  is indefinite, the usual symmetry preserving methods for solving systems may be unstable. It is unfortunate that the stable method in [3] has not been adapted to sparse matrices. A method which uses the inverse power method with several shifts to find the eigenvectors corresponding to eigenvalues lying in a given interval has been described in [24].

## REFERENCES

- [1] F. L. Bauer, Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme, *Z. Angew. Math. Phys.*, vol. 8, 1957, 214-235 (German).
- [2] W. W. Bradbury and R. Fletcher, New iterative methods for solution of the eigenproblem, *Numer. math.*, vol. 9, 1966, 259-267.
- [3] J. R. Bunch and B. N. Parlett, Direct methods for solving symmetric indefinite systems of linear equations. *SIAM j. numer. anal.*, vol. 8, 1972, 639-655.
- [4] M. Clint and A. Jennings, The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration, *Comput. j.*, vol. 13, 1970, 76-80.
- [5] M. Clint and A. Jennings, A simultaneous iteration method for the unsymmetric eigenvalue problem, *J. inst. math. appl.*, vol. 8, 1971, 111-121.
- [6] J. L. B. Cooper, The solution of natural frequency equations by relaxation methods, *Quart. appl. math.*, vol. 6, 1948, 179-183.
- [7] A. R. Curtis and J. K. Reid, The solution of large sparse unsymmetric systems of linear equations, IFIP Congress 71, Num. Math. Section, 1971.
- [8] A. R. Curtis and J. K. Reid, FORTRAN subroutines for the solution of sparse sets of linear equations, report AERE-R6844, United Kingdom Atomic Energy Research Establishment, Harwell, England, 1971.
- [9] F. W. Dorr, The direct solution of the discrete Poisson equation on a rectangle, *SIAM rev.*, vol. 12, 1970, 248-263.
- [10] A. M. Erisman, Sparse matrix approach to the frequency domain analysis of linear passive electrical networks, *Sparse matrices and their applications* (D. J. Rose and R. A. Willoughby eds.), Plenum, New York, 1972, 31-40.
- [11] D. K. Faddeev and V. N. Faddeeva, *Computational methods of linear algebra*, Freeman, San Francisco, 1960.
- [12] I. Fried, Gradient methods for finite element eigenproblems, *AIAA j.*, vol. 1, no. 4, 1969, 739-741.
- [13] I. Fried, Optimal gradient minimization scheme for finite element eigenproblems, *J. sound vib.*, vol. 20, 1972, 333-342.
- [14] A. George, Nested dissection of a regular finite element mesh, *SIAM j. numer. anal.*, vol. 10, 1973, 345-363.
- [15] G. H. Golub, Some modified eigenvalue problems, report Stan-CS-234-71, Stanford University, 1971.
- [16] G. H. Golub, R. Underwood, and J. H. Wilkinson, The Lanczos algorithm for the symmetric  $Ax = \lambda Bx$  problem, report Stan-CS-72-270, Stanford University, 1972.
- [17] K. K. Gupta, Solution of eigenvalue problems by Sturm sequence method, *Int. j. numer. math. eng.*, vol. 4, 1972, 379-404.
- [18] F. G. Gustavson, Some basic techniques for solving sparse systems of linear equations, *Sparse matrices and their applications* (D. J. Rose and R. A. Willoughby eds.), Plenum, New York, 1972, 41-52.
- [19] E. Hellerman and D. C. Rarick, The partitioned preassigned pivot procedure ( $P^4$ ) in *Sparse matrices and their applications* (D. J. Rose and R. A. Willoughby, eds.), Plenum, New York, 1972, 67-76.



- [20] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. res. nat. bur. standards, vol. 49, 1952, 409-436.
- [21] A. J. Hoffman, M. S. Martin, and D. J. Rose, Complexity bounds for regular finite difference and finite element grids, SIAM j. numer. anal., vol. 10, 1973, 364-369.
- [22] A. S. Householder, The theory of matrices in numerical analysis, Blaisdell, Boston, Mass., 1964.
- [23] G. Hufford, On the computation of selected eigenvalues, J. analys. math., vol. 16, 1966, 423-451.
- [24] P. S. Jensen, The solution of large symmetric eigenproblems by sectioning, SIAM j. numer. anal., vol. 9, 1972, 534-545.
- [25] W. Kahan, Relaxation methods for an eigenproblem, technical report CS44, Stanford University, Department of Computer Sciences, 1966.
- [26] S. Kaniel, Estimates for some computational techniques in linear algebra, Math. comp., vol. 20, 1966, 369-378.
- [27] D. E. Knuth, The art of computer programming, V. I Fundamental Algorithms, Addison-Wesley, Reading, Mass., 1968.
- [28] P. Lancaster, Lambda-matrices and vibrating systems, Pergamon, Oxford, 1966.
- [29] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential operators, J. res. nat. bur. standards, vol. 45, 1950, 255-282.
- [30] R. S. Martin and J. H. Wilkinson, Reduction of the symmetric eigenproblem  $Ax = \lambda Bx$  and related problems to standard form, Numer. math., vol. 11, 1968, 99-110.
- [31] A. C. McKellar and E. G. Coffman, Organizing matrices and matrix operations for a paged memory system, Comm. ACM, vol. 12, 1969, 153-165.
- [32] C. B. Moler, Matrix computation with Fortran and paging, Comm. ACM, vol. 15, 1970, 268-270.
- [33] R. K. Nesbet, Algorithm for diagonalization of large matrices, J. chem. phys., vol. 43, 1965, 311-312.
- [34] A. Ostrowski, On the convergence of the Rayleigh quotient for the computation of characteristic roots and vectors I, II, III, IV, V, VI, Arch. rational mech. anal., vol. 1, 233-241; vol. 2, 243-248; vol. 3, 325-340, 341-347, 472-481; vol. 4, 153-165, 1958-59.
- [35] C. C. Paige, Practical use of the symmetric Lanczos process with reorthogonalization, BIT, vol. 10, 1970, 183-195.
- [36] C. C. Paige, Computational variants of the Lanczos method for the eigenproblem, J. inst. math. appl., vol. 10, 1972, 373-381.
- [37] C. C. Paige and M. A. Saunders, Solution of sparse indefinite systems of equations and least squares problems, report STAN-CS-73-399, Stanford University, Computer Science Department, 1973.
- [38] B. Parlett, Laguerre's method applied to the matrix eigenvalue problem, Math. comp., vol. 18, 1964, 464-465.
- [39] B. N. Parlett and W. G. Poole, A geometric theory for the QR, LU, and power iterations, SIAM j. numer. anal., vol. 10, 1973, 389-412.
- [40] A. Pellew and R. U. Southwell, Relaxation methods applied to engineering problems VI. The natural frequencies of systems having restricted freedom, Proc. roy. soc. (A), vol. 175, 1940, 262-290.
- [41] G. Peters and J. H. Wilkinson, Eigenvalues of  $Ax = \lambda Bx$  with band symmetric A and B, Comp. j., vol. 12, 1969, 398-404.
- [42] G. Peters and J. H. Wilkinson, The calculation of specified eigenvectors by inverse iteration, Handbook for automatic computation, VII linear algebra (J. H. Wilkinson and C. Reinsch, eds.), Springer-Verlag, New York, Heidelberg, Berlin, 1971, 418-439.
- [43] J. K. Reid (ed.), Large sparse sets of linear equations, Academic Press, New York, 1970.
- [44] W. C. Rheinboldt and C. K. Mesztenyi, Arc graphs and their possible application to sparse matrix problems, technical report TR-238, University of Maryland, Computer Science Center, 1973.
- [45] W. C. Rheinboldt and C. K. Mesztenyi, Programs for the solution of large sparse matrix problems based on the arc graph structure, technical report TR-262, University of Maryland, Computer Science Center, 1973.
- [46] L. D. Rogers, Optimal paging strategies and stability considerations for solving large linear systems, Ph.D. thesis, Department of Applied Analysis and Computer Science, University of Waterloo, 1973.
- [47] D. J. Rose and R. A. Willoughby (eds.), Sparse matrices and their applications, Plenum, New York, 1972.
- [48] A. Ruhe, Iterative eigenvalue algorithms for large symmetric matrices, report UMINF-31.72, Department of Information Processing, Umeå Universitet, November 15, 1973.
- [49] A. Ruhe and T. Wiberg, The method of conjugate gradients used in inverse iteration, BIT, vol. 12, 1972, 543-554.
- [50] A. Ruhe, SOR-methods for the eigenvalue problem, report UMINF-37.73, Department of Information Processing, University of Umeå, May 23, 1973. To appear in Math. comp.
- [51] A. Ruhe, Iterative eigenvalue algorithms based on convergent splittings, report UMINF-43.73, Department of Information Processing, University of Umeå, October 30, 1973.
- [52] H. Rutishauser, Computational aspects of F. L. Bauer's simultaneous iteration method, Numer. math., vol. 13, 1969, 4-13.
- [53] H. Rutishauser, Simultaneous iteration method for symmetric matrices, Numer. math., vol. 16, 1970, 205-223.
- [54] M. A. Saunders, Large scale linear programming using the Cholesky factorization, dissertation, Stanford University, Department of Computer Science, 1972.
- [55] I. Shavitt, C. F. Bender, A. Pipano, and R. P. Hosteny, The iterative calculation of the lowest or highest eigenvalues and corresponding eigenvectors, J. comput. phys., vol. 11, 1973, 90-108.
- [56] G. W. Stewart, Accelerating the orthogonal iteration for the eigenvalues of a Hermitian matrix, Numer. math., vol. 13, 1969, 362-376.
- [57] G. W. Stewart, Introduction to matrix computations, Academic Press, New York, 1973.
- [58] G. Strang and G. J. Fix, An analysis of the finite element method, Prentice-Hall, Englewood Cliffs, N. J., 1973.
- [59] R. P. Tewarson, On the transformation of symmetric sparse matrices to the triple diagonal form, Internat. j. comput. math., vol. 2, 1970, 247-258.
- [60] R. S. Varga, Matrix iterative analysis, Prentice Hall, Inc., Englewood Cliffs, N. J., 1962.
- [61] J. H. Wilkinson, The algebraic eigenvalue problem, Oxford Univ. Press (Clarendon), London and New York, 1965.
- [62] J. H. Wilkinson and C. Reinsch (eds.), Handbook for automatic computation, V, II Linear Algebra, Springer-Verlag, New York, 1971.
- [63] R. A. Willoughby (ed.), Sparse matrix proceedings, RAI IBM Research, Yorktown Heights, New York, 1968.
- [64] D. M. Young, Iterative solution of large linear systems, Academic Press, New York-London, 1971.