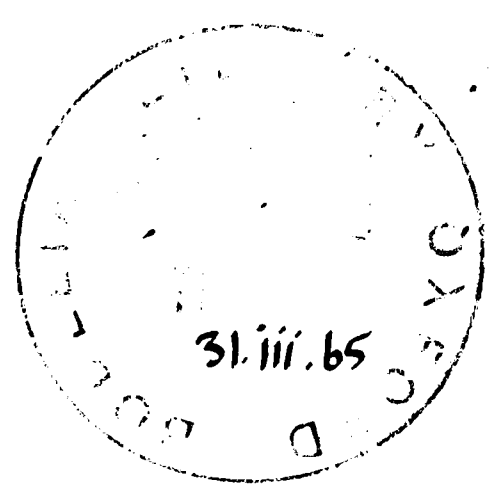
COMPUTATIONAL PROBLEMS IN LINEAR ALGEBRA

J.K. REID

The Queen's College

Oxford



Thesis submitted for the degree of

Doctor of Philosophy

in October 1964

ABSTRACT

In this thesis we consider the problems that arise in computational linear algebra when the matrix involved is very large and sparse. This necessarily results in the emphasis throughout being mostly on iterative methods, but we begin with a chapter in which the usual direct methods for the solution of linear equations are summarized with the very large problem in mind. Methods for band-matrices and block tridiagonal matrices are considered and compared. Kron's method of tearing is described briefly.

The basic and well-established techniques of iterative solution of linear equations are described. Sufficient conditions for convergence are stated and the convergence is established under these conditions. A section is devoted to the problem of deciding when the iterations may be terminated. The method of Kaczmarz is given more emphasis than is usual in the literature, a symmetric version is introduced and rigorous proofs of convergence are given.

Matrices with "property A" are defined and the usual associated results are given. The methods of Carré and Kulsrud for finding the optimum relaxation parameter in SOR are described and compared with a new technique. We also consider methods for finding the best relaxation parameter in symmetric SOR.

Block iterative methods are described briefly, including Varga's result on the comparison of different splittings.

Gradient and semi-iterative methods are considered in detail, including the cases of non-symmetric and symmetric non-definite matrices, and are compared for numerical stability.

For the eigenvalue problem we again summarize some well-known methods with an emphasis on the very large matrix. We give an example of the accurate results that can be expected from Lanczos' method of minimised iterations without re-orthogonalisation and show

how such results can be guaranteed. Inverse iteration, the Rayleigh quotient iteration and Rutishauser's L-R algorithm are considered. We also devote a chapter to iterative methods that yield continually improving approximations to an eigenvector and corresponding eigenvalue. These are very economical in storage but have received little attention in the literature.

We introduce the concept of "numerical rank", since rank really has no meaning for a matrix given by rounded numbers. We give means of finding bounds for this numerical rank.

The extension of the methods to complex matrices is considered briefly and we conclude with a description of some problems that may give rise to large sparse matrices. We consider in particular the so-called "L-membrane" problem and use a conformal transformation. In this way we also illustrate the problem of the solution of an elliptic equation over an area with a curved boundary. We consider the iteration of Federenko, which has its most obvious application in the field of elliptic equations. The thesis is concluded with two further examples, from linear analysis of variance and from surveying, each of which is likely to involve large sparse matrices.

PREFACE

This thesis is an exposition of research undertaken in the Oxford University Computing Laboratory between October 1961 and August 1964. The main claim to originality lies in the opinions expressed throughout, but here is a list of items we consider original.

All numbers quoted anywhere; a few have already been published elsewhere, but they have been checked by an independent computation.

The work on small relaxation parameters.

The extension of Stein's work on near-symmetric matrices.

A new emphasis on Kaczmarz iteration, including a rigorous proof of its convergence and its extension to symmetric Kaczmarz iteration.

A proof of the property A result using matrix rather than determinant algebra.

A new technique for finding w_{opt} in SOR.

The method of conjugate gradients for a general matrix.

A new proof of convergence of Rutishauser's L-R algorithm, the generalisation of the algorithm to block form and an error analysis of the symmetric positive-definite case.

The use of SOR and Kaczmarz iteration for the eigenvalue problem.

The idea and definition of "numerical rank".

Virtually the whole of Chapters 11 and 12 since these are essentially practical in emphasis.

The work on the L-membrane problem has been submitted to J. SIAM as part of a joint paper with Dr. J.E. Walsh of Manchester University.

I would like to take this opportunity of thanking my supervisor, Professor L. Fox, for the many stimulating conversations we have had over three years and for his valuable comments on the presentation of this thesis. I would also like to thank D.S.I.R. for the grant

which made this work possible, and Miss O. Moon who has had the job of interpreting my handwriting.

John K. Reid.

C O N T E N T S

INTRODUCTION

This thesis is concerned with the numerical solution of two basic matrix problems. The first is the set of simultaneous linear equations

$$A \underline{x} = \underline{b} , \tag{1}$$

where A is an nxn matrix and $\underline{x}$ and $\underline{b}$ are nxm matrices. In most cases we will have m = 1. The second problem is the eigenvalue equation

$$A \underline{x} = \lambda \underline{x} , \tag{2}$$

where again A is an nxn matrix.

It will be assumed throughout that n, the order of A, is very large. Our practical work has mostly been on matrices of order a few hundred, but orders in the range of thousands or tens of thousands can arise quite frequently in such problems as three-dimensional elliptic partial differential equations. On the assumption that n is large, we will usually neglect all but the dominant term in any calculation of the number of arithmetic operations or the storage requirement of a particular method.

In the main it will be assumed that most of the elements of A are zero. It is common to find that the average number of non-zero elements in a row is between about five and ten and we call such matrices sparse. The non-zero elements are often grouped about the main diagonal and we refer to such matrices as band-matrices and say that the band-width is $(2r - 1)$ if $a_{ij} = 0$ for $|i - j| \geq r$. We take all matrix elements to be real and the adaptation of the methods to the complex case will be considered briefly in Chapter 10.

We assume throughout that the reader has a theoretical knowledge of matrix algebra, but all the numerical theorems will be proved in detail unless they are included in literature that is easily available.

CHAPTER 1

DIRECT METHODS FOR THE SOLUTION OF LINEAR EQUATIONS

Gaussian Elimination and Triangular Decomposition

Gaussian elimination and triangular decomposition are now very well established methods and will not be described in detail here (see N.P.L., 1961, for example). Some useful comparisons will be made, however, of the numbers of arithmetic operations and the storage requirements of the method in its various forms.

In Gaussian elimination, with row and column interchanges, the operations performed on the matrix A can be expressed in the matrix form

$$L^{-1} P A Q = U, \tag{1}$$

where P and Q are permutation matrices, L^{-1} is lower triangular with units along its diagonal and U is upper triangular. The matrix L^{-1} is generally not calculated explicitly since

$$L^{-1} P = L_{n-1} P_{n-1} L_{n-2} P_{n-2} \cdots L_1 P_1 \tag{2}$$

where L_i is lower triangular with units along its diagonal and zeros elsewhere except in the i^{th} column, and P_i is a permutation matrix. The operations represented by $L_i P_i$ are applied successively. If triangular decomposition is performed (1) takes the form

$$P A Q = L U. \tag{3}$$

In the Gaussian elimination process (1) the row and column interchanges are represented by P and Q and are necessary to prevent any large growth of the elements during the reduction. Any such growth will adversely affect the accuracy of the solution (see Wilkinson, 1961). It is quite common practice to take Q = I, that is perform row interchanges only. The maximum possible growth-rate is considerably increased, but it has been found in practice that a very large growth-rate is rarely encountered, and in any case merely

by looking at each reduced matrix we can always compute this growth and use it in Wilkinson's result.

If our computing machine has no facility for double-length computation of scalar products, then the processes (1) and (3) are exactly equivalent. On the other hand (3) will produce a more accurate solution if exact computation of scalar products is used. With full interchanges a considerable amount of additional computation is needed for process (3), but this is not the case if only row interchanges are used. Wilkinson (1961) proposed that (1) should first be performed in order to find P and Q and then (3) can be used in its full form. This procedure will require double the amount of arithmetic, which seems very wasteful, and the difficulty can be overcome with great ease if there is plenty of room in the fast store. We merely perform (1), keeping the reduced matrices to double-length precision, which is exactly equivalent to performing (3) with exact computation of scalar products. Alternatively, if we do not have so much room in the computing store, then we use the backing store to keep both A and the multipliers and the reduced form of (1). If $\underline{x}_a$ is the vector obtained by (1), we form the residual vector

$$\underline{r}_a = \blacklozenge\, \underline{b} - A\,\underline{x}_a \,, \tag{4}$$

where the matrix product is formed using double-length accumulation of the scalar products. A correction is now found by solving the set of equations

$$A\,\underline{x}_c = \underline{r}_a \,, \tag{5}$$

using the known multipliers and reduced form. We take as next approximation

$$\underline{x}'_a = \underline{x}_a + \underline{x}_c \,, \tag{6}$$

and repeat the process. It is unlikely that more than one or two

cycles will be needed.

There are two special cases of matrices for which no interchanges are necessary. Wilkinson (1961) shows that if A is diagonally dominant by columns, that is

$$|a_{ii}| > \sum_{j \neq i} |a_{ji}| , \tag{7}$$

then the growth-rate is certainly less than two, and all the reduced matrices are also diagonally dominant. It follows that the rows are correctly ordered for partial pivoting. The other special case is when A is positive definite. In this case (3) can be replaced by its symmetric version

$$A = LL^T . \tag{8}$$

Wilkinson (1961) obtains good error bounds for this technique. If our machine has facilities for double-length computation of scalar products, then this version is to be preferred. Where scalar products are accumulated in single-length arithmetic the process is equivalent to Gaussian elimination without interchanges and where the pivotal rows are scaled by dividing by the square root of the pivot. It is clear that if Gaussian elimination with floating-point arithmetic is used, exactly the same error bounds will apply. Indeed we will get a marginally better result with Gaussian elimination since fewer operations are performed. In both cases full advantage of the symmetry can be taken. Only those elements of A on and above (or below) the diagonal need be stored. The same applies to all the reduced matrices of the Gaussian elimination method since these are also symmetric. If we wish to work with fixed-point arithmetic, then the symmetric decomposition (8) is to be preferred since it can be readily verified that if A is scaled so that its greatest element is less than unity, then all the elements of L are less than unity in modulus.

A method of elimination was proposed in NPL (1961) for the case where the matrices A and $\underline{b}$ are just too large to be stored in the fast-store of the computer. At a typical stage the first (r-1) equations have been reduced to an upper triangular format. The next equation is read into the store and its first element is compared with the first element of the first equation. If it is greater in modulus, these two equations are interchanged. A multiple of the resulting first equation is now added to the resulting r^{th} so that the latter has a zero as its first element. Now the second element of the r^{th} equation is compared with the second element of the second equation and if it is greater in modulus these equations are interchanged. The second element of the resulting r^{th} equation is now eliminated. And so the process continues. As in the case of ordinary Gaussian elimination, with row interchanges, all the multipliers will be less than unity and the maximum possible growth ratio is 2^{n-1}.

An advantage of this method is that the storage needed is reduced from $n^2 + mn$ to $\frac{1}{2}n(n+3)+1+nm$. However the growth rate is likely to be greater than if the ordinary Gaussian elimination were used, since the multipliers are likely to be larger.

In the case that A is diagonally dominant, as we have already seen, no interchanges are needed. In this case we gain in storage without altering the accuracy.

In the case that A is symmetric and positive-definite this method offers no advantage since we need only store the upper part of the matrices anyway.

Number of Arithmetic Operations and Storage Requirements

It will be useful to have an estimate of the number of arithmetic operations performed and the storage requirement for these various methods. Unfortunately the machine time is not

entirely taken up with arithmetic, "red-tape" and finding numbers in storage may take a comparable time, but it is difficult to include these in a table that is applicable to any machine. These estimates are given in Table 1. The time taken for a multiplication is represented by μ, the time for an addition by α and the time for a division by δ. On most machines division takes considerably longer than multiplication and in all but the NPL technique the number of divisions is minimized if each pivot is replaced by its reciprocal, and we assume that this is done. We take the time for comparing the moduli of two numbers to be α.

Method	Work on A	Work on R.H.S.	Storage
Gaussian Elimination with full interchanges	$\frac{1}{3}n^3\mu + \frac{2}{3}n^3\alpha + O(n^2)$	$n^2m(\alpha + \mu) + O(n)$	$n(n + m)$
Gaussian Elimination with row interchanges or LU split with row interchanges	$\frac{1}{3}n^3\mu + \frac{1}{3}n^3\alpha + O(n^2)$	$n^2m(\alpha + \mu) + O(n)$	$n(n + m)$ [or $n(\frac{n}{2} + m) + O(n)$ by NPL technique]
Gaussian Elimination symmetric positive definite case or LL^T split	$\frac{1}{6}n^3\mu + \frac{1}{6}n^3\alpha + O(n^2)$	$n^2m(\alpha + \mu) + O(n)$	$n(\frac{n}{2} + m) + O(n)$ [or $n(\frac{n}{2} + 1) + O(n)$ if each column of $\underline{b}$ processed separately]
Calculation of $\underline{x}_0$ given $\underline{x}_a$	nil	$2n^2m(\alpha + \mu) + O(n)$	$n(2n + 3m)$ [~~or $n(2n + m)$ on aux. store and $n(n + m)$ in high-speed store~~]
Gaussian Elimination A diagonally dominant or LU split	$\frac{1}{3}n^3\mu + \frac{1}{3}n^3\alpha$ $O(n^2)$	$n^2m(\alpha + \mu) + O(n)$	$n(n + m)$

Table 1

Number of arithmetic operations and storage for direct methods.

Band Matrices

It often happens that the matrix A is a band-matrix with

$$a_{ij} = 0 \quad \text{if} \quad | i - j | \geq r, \tag{9}$$

so that its band-width is (2r-1). It is easily seen that if full interchanges are used, the band character of the matrix is destroyed and it will have to be treated as a full matrix. This will increase the storage and arithmetic very considerably. We therefore do not recommend the use of full interchanges for a band-matrix. If no interchanges are used the matrices L and U are both band-matrices of band-width r. If row interchanges are used, U remains a band-matrix with band-width increased to (2r-1) but L does not, although no column has more than r non-zero elements. This will make the organization of the L-U split non-trivial and for this reason we feel that Gaussian elimination is to be preferred.

We can further exploit the band nature of the matrix to reduce the amount of fast storage needed if we have a convenient backing store available. Most modern machines have such a store in the form of magnetic tape units. The band structure means that we need have only r equations in the fast store at once. The use of the NPL technique will further reduce this requirement in the case of a non-symmetric matrix. We show in Table 2 the number of arithmetic operations performed and the storage requirements of these methods.

Table 2

Method	Gaussian elimination with row interchanges	Gaussian elimination or L-U split, A diagonally dominant	Gaussian elimination or $L\text{-}L^T$ split, A positive definite
Work on A	$n(r-1)(2r-1)(\alpha+\mu)$ $+n\delta$	$n(r-1)(r\mu+(r-1)\alpha)$ $+n\delta$	$n(r-1)(\lfloor\frac{r}{2}+1\rfloor\mu+\frac{r}{2}\alpha)$ $+n\delta$
Work on r.h.s.	$nm[(3r-2)\mu+(3r-3)\alpha]$	$nm[(2r-1)\mu+(2r-2)\alpha]$	$nm[(2r-1)\mu+(2r-2)\alpha]$
Storage	$n(2r-1+m)$	~~$n(3r-1+m)$~~ $n(2r-1+m)$	$n(r+m)$
Fast store	$r(2r-1+m)$	$r(3r-1)+rm$ or by PPL technique, $r(r+m+1)-1$	$r(r+m)$
Work to find $\underline{x}_0$ given $\underline{x}_s$	$nm[(5r-3)\mu+(5r-4)\alpha]$	$nm[(4r-2)\mu+(4r-3)\alpha]$	$nm[(4r-2)\mu+(4r-3)\alpha]$
Storage to find $\underline{x}_0$ given $\underline{x}_s$	$n(5r-3+3m)$	$n(4r-2+3m)$	$n(2r+3m)$

In all cases terms $O(n^0)$ have been omitted.

Use of Partitioned Matrices

It often happens that the matrix A is block tri-diagonal, that is it can be partitioned into submatrices in the form

$$A = \begin{bmatrix} A_1 & B_1 & & & \\ C_2 & A_2 & B_2 & & \\ & C_3 & A_3 & B_3 & \\ & \cdots\cdots & \cdots\cdots & \cdots\cdots & \\ & & & \cdots\cdots & \\ & & & C_q & A_q \end{bmatrix} \tag{10}$$

In such cases it has been proposed that the matrix A be split into the product of a block lower triangular matrix and a block upper triangular matrix, that is

$$A = \begin{bmatrix} I & & \\ E_2 & I & \\ & E_3 & I \\ & \cdots\cdots & \end{bmatrix} \begin{bmatrix} F_1 & G_1 & \\ & F_2 & G_2 \\ & & F_3 & G_3 \\ & & \cdots\cdots \end{bmatrix} \tag{11}$$

and the submatrices are calculated from the equations

$$\begin{aligned} &F_1 = A_1 , \quad G_1 = B_1 , \\ &E_i = C_i F_{i-1}^{-1} , \quad F_i = A_i - E_i G_{i-1} , \quad G_i = B_i , \quad i > 1. \end{aligned} \tag{12}$$

For a general matrix this is a rather unsatisfactory method, since it suffers most of the disadvantage of Gaussian elimination without interchanges. We will illustrate with an example. Suppose we are working with four significant decimals, and consider the decomposition

$$\left[\begin{array}{c|c} 1 & 90 \\ \hline 90 & 2 \end{array}\right] = \left[\begin{array}{c|c} 1 & \\ \hline 90 & 1 \end{array}\right] \left[\begin{array}{c|c} 1 & 90 \\ \hline & -8098 \end{array}\right] \tag{13}$$

We would have obtained the same result if the element a_{33} had been any number in the range

$$1.5 < a_{33} < 2.5. \tag{14}$$

With Gaussian elimination and row interchanges we have

$$\begin{bmatrix} 90 & & 2 \\ & 90 & \\ 1 & & 90 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ .01111 & & 1 \end{bmatrix} \begin{bmatrix} 90 & & 2 \\ & 90 & \\ & & 89.98 \end{bmatrix}, \tag{15}$$

and if any element of A is changed by more than 0.005 we will obtain a different answer.

Thus the method is limited to cases where we know a priori that no interchanges are necessary, that is for diagonally dominant matrices and positive-definite symmetric matrices. To compare the amount of arithmetic and storage we will have to consider in more detail the actual operations performed. If the vectors $\underline{x}$ and $\underline{b}$ are partitioned to correspond with the partitioning of A, we have to calculate

$$\left.\begin{aligned} F_1 &= A_1, \quad F_i = A_i - C_i F_{i-1}^{-1} B_{i-1} \\ \underline{Y}_1 &= \underline{b}_1, \quad \underline{Y}_i = \underline{b}_i - C_i F_{i-1}^{-1} \underline{Y}_{i-1} \end{aligned}\right\} \quad i = 2 \ldots q$$

$$\underline{x}_q = F_q^{-1} \underline{Y}_q \qquad \underline{x}_i = F_i^{-1} (\underline{Y}_i - B_i \underline{x}_{i+1}) \quad i = q-1, \ldots 1 \tag{16}$$

Clearly our best plan is to overwrite each F_i by its triangular decomposition,

$$F_i = L_i U_i. \tag{17}$$

We can establish the relationship of this process with the conventional triangular decomposition of A by writing

$$A =$$

$$\begin{bmatrix} I & & \\ E_2 & I & \\ & E_3 & I \\ \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} L_1 & & \\ & L_2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} L_1^{-1} & & \\ & L_2^{-1} & \\ & & \ddots \end{bmatrix} \begin{bmatrix} F_1 & G_1 & \\ & F_2 & G_2 \\ & & \ddots \end{bmatrix} \tag{18}$$

$$= \begin{bmatrix} L_1 & & \\ E_2 L_1 & L_2 & \\ & E_3 L_2 & L_3 \\ & \cdots & \cdots \end{bmatrix} \begin{bmatrix} U_1 & L_1^{-1} G_1 & \\ & U_2 & L_2^{-1} G_2 \\ & & U_3 \\ & & \cdots \end{bmatrix} .$$

The operations actually performed when written in the form of (16) are given by

$$\left.\begin{aligned} &L_1 U_1 = A_1 , \quad L_i U_i = A_i - C_i U_{i-1}^{-1} L_{i-1}^{-1} B_{i-1} \\ &\underline{x}_1' = L_1^{-1} \underline{b}_1 , \quad \underline{v}_i' = L_i^{-1} (\underline{b}_i - C_i U_{i-1}^{-1} \underline{v}_{i-1}') \\ &\underline{x}_q = U_q^{-1} \underline{x}_q' \qquad \underline{x}_i = U_i^{-1} (\underline{v}_i' - L_i^{-1} B_i \underline{x}_{i+1}') \end{aligned}\right\} . \tag{19}$$

The matrices

$$L_i^{-1} G_i = L_i^{-1} B_i$$

and

$$E_i L_{i-1} = C_i F_{i-1}^{-1} L_{i-1} = C_i U_{i-1}^{-1}$$

are calculated and stored in the course of the computation in line 1. The first lines of (16) and (19) are identical and take the same number of operations, but a careful comparison of the rest of the computation shows that in each case we perform 2q matrix by vector multiplications. In (16) however we perform 4q forward or backward substitutions against 2q in (19) so that the form (16) involves more work than (19). However there is no disadvantage in working with

equations (19) and this may provide a convenient way of organising the calculation, instead of treating A as a matrix of variable band-width.

In many cases B_i and C_i are diagonal matrices. Then if all the submatrices A_i are of the same order, the formulation (19) will require exactly the same amount of work as if A were treated as a band-matrix. If the A_i are not of the same order we are again effectively working with A as a matrix of variable band-width.

These remarks remain valid for a positive-definite A, and the method of (16) suffers the further disadvantage of destroying the symmetry. Again the method of (19) is convenient for the treatment of a matrix with variable band-width.

Orthogonal Reductions to Triangular Form

In this process the matrix A is multiplied by a sequence of orthogonal matrices, to produce successively the matrices $A_1, A_2, \ldots, A_{s}, \ldots$, each having one more zero below the diagonal (Givens, 1959) or one more null column below the diagonal (Householder, 1958). For a full matrix the Givens process requires about four times as many arithmetic operations as Gaussian elimination and the Householder about twice as many. Unless A is very ill-conditioned the process described in equations (4) (5) and (6) will require less work and will produce the most accurate solution that is possible working with single-length arithmetic. If these methods are applied to a band-matrix the band structure is not maintained in the reduced matrices. The work is thus very much increased. The methods are, however, very convenient when the matrix is full and we wish to solve the eigenvalue problem also.

Kron's Method of Tearing

We give here a brief explanation of Kron's method and will later consider its application to elliptic partial differential equations. For a fuller account see Kron (1963).

The method depends on expressing A as the sum of two matrices

$$A = B + C, \tag{20}$$

where B is non-singular and C has rank less than n.

It follows that

$$\begin{aligned} B\underline{x} &= \underline{b} - C\underline{x}, \\ \underline{x} &= B^{-1}\underline{b} - B^{-1}C\underline{x} . \end{aligned} \tag{21}$$

Now $B^{-1}C$ has rank less than n and so can be expressed as the product of two rectangular matrices,

$$B^{-1}C = S\,T, \tag{22}$$

where S has n rows and r columns and T has r rows and n columns; r is not less than the rank of C.

From equation (21) we have

$$\underline{x} = B^{-1}\underline{b} - S\,T\underline{x} , \tag{23}$$

from which it follows that

$$T\underline{x} = T\,B^{-1}\underline{b} - T\,S\,T\underline{x} . \tag{24}$$

Writing $\underline{z} = T\underline{x}$, we have

$$\underline{z} = T\,B^{-1}\underline{b} - T\,S\,\underline{z} . \tag{25}$$

If this set is solved we can obtain $\underline{x}$ from (23) as

$$\underline{x} = B^{-1}\underline{b} - S\underline{z} . \tag{26}$$

It is clear that a check should be made that B is not substantially worse conditioned than A.

For this method to show any advantage over those already discussed in this chapter, it is necessary that B be easy to invert and that r be much less than n.

We illustrate the method by its application to a matrix that has been partitioned into blocks,

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1s} \\ A_{21} & A_{22} & \cdots & \\ \cdots & \cdots & \cdots & \cdots \\ & & & A_{ss} \end{bmatrix} \tag{27}$$

In the notation of (20) we take

$$B = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{ss} \end{bmatrix} \tag{28}$$

and r to be the number of rows in

$$C = A - B = \begin{bmatrix} 0 & A_{12} & \cdots & A_{1s} \\ A_{21} & 0 & \cdots & \\ \cdots & \cdots & \cdots & \cdots \\ & & & 0 \end{bmatrix} \tag{29}$$

that are not entirely zeros. We obtain T from C by omitting from C all its zero rows and S is obtained from B^{-1} by omitting corresponding columns. It should be noted that a full inverse of B need not be computed in this case.

It is difficult to make comparisons with other methods, since so much depends on the matrix A. If there are few non-zero elements in the off-diagonal blocks of (27) and these are not grouped near the diagonal, then the matrix cannot be treated as a band-matrix and Kron's method will show substantial savings in work and storage. The original application was to matrices arising from electrical networks that divide naturally into several regions with few connections between the regions. These provide good examples of the kind of matrix just described. Kron's method may also be tempting when several of the matrices A_{ii} are identical so that the inversion of B is particularly simple.

CHAPTER 2

BASIC ITERATIVE METHODS FOR THE SOLUTION OF LINEAR EQUATIONS

Introduction

The direct methods described in Chapter 1 are very satisfactory when the matrix A can be stored and has few zero elements, or when it is a band-matrix with few zeros inside the band and the band can be stored, or when Kron's method is convenient. Where these conditions do not hold we may decide to use an iterative method. Indeed storage limitation may leave us no alternative. This means that we do not store the matrix A, but rather have a programme that allows us, given an approximation $\underline{x}^{(r)}$ to the solution $\underline{x}$, to calculate a better approximation $\underline{x}^{(r+1)}$. Our decision to use an iterative method will probably be taken on grounds of storage, but if A has an exceptionally large number of zero elements we may also find a considerable reduction in the amount of work to be done. We will show an example of such a case at the end of Chapter 3.

Stationary Methods

We express the matrix A in the form

$$A = M - N \tag{1}$$

where M is non-singular and easily inverted. The iteration scheme is given by

$$M \underline{x}^{(k+1)} = N \underline{x}^{(k)} + \underline{b}. \tag{2}$$

Since the solution satisfies the equation

$$M \underline{x} = N \underline{x} + \underline{b}, \tag{3}$$

the error vector

$$\underline{e}^{(k)} = \underline{x} - \underline{x}^{(k)} \tag{4}$$

satisfies the equation

$$M \underline{e}^{(k+1)} = N \underline{e}^{(k)},$$

that is

$$\underline{e}^{(k+1)} = M^{-1}N \underline{e}^{(k)}. \tag{5}$$

The residual vector

$$\underline{r}^{(k)} = \underline{b} - A \underline{x}^{(k)} = A(\underline{x} - \underline{x}^{(k)}) = A \underline{e}^{(k)} \tag{6}$$

and the displacement vector

$$\underline{\Delta}^{(k)} = \underline{x}^{(k+1)} - \underline{x}^{(k)} = \underline{e}^{(k)} - \underline{e}^{(k+1)} \tag{7}$$

satisfy the similar equations

$$\underline{r}^{(k+1)} = M^{-1}N \underline{r}^{(k)} \tag{8}$$

and

$$\underline{\Delta}^{(k+1)} = M^{-1}N \underline{\Delta}^{(k)}. \tag{9}$$

The ~~number r~~ matrix $T = M^{-1}N$ will be called the iteration matrix. If J is the Jordan canonical form of T and can be obtained from T by the transformation

$$T = S J S^{-1}, \tag{10}$$

then

$$\begin{aligned} \underline{e}^{(k)} &= T^k \underline{e}^{(0)} \\ &= S J^k S^{-1} \underline{e}^{(0)}. \end{aligned} \tag{11}$$

We study the convergence of the iteration by considering J^k. The block diagonal nature of J is preserved in its powers and we therefore need consider only the typical submatrix of J,

$$J_1 = \begin{bmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda \end{bmatrix} . \tag{12}$$

It is easily confirmed by induction that

$$J_1^k = \begin{bmatrix} \lambda^k & {}^kC_1\,\lambda^{k-1} & {}^kC_2\,\lambda^{k-2} & {}^kC_3\,\lambda^{k-3} & \cdots \\ & \lambda^k & {}^kC_1\,\lambda^{k-1} & {}^kC_2\,\lambda^{k-2} & \cdots \\ & & \lambda^k & {}^kC_1\,\lambda^{k-1} & \cdots \\ & & & & \end{bmatrix} \tag{13}$$

and hence that J_1^k converges to the null matrix as k tends to infinity if and only if $|\lambda| < 1$. It follows that a necessary and sufficient condition for the convergence of the iteration for arbitrary $\underline{e}^{(0)}$ is that

$$\rho(T) < 1, \tag{14}$$

where $\rho(T)$ is the spectral radius of T. We say that the matrix T is convergent if (14) holds.

From equation (11) we have

$$\| \underline{e}^{(r)} \| \leq \| T^r \| \; \| \underline{e}^{(0)} \| . \tag{15}$$

where the vector norm and matrix norm are consistent with each other. $\| T^r \|$ gives us an upper bound for the ratio $\| \underline{e}^{(r)} \| / \| \underline{e}^{(0)} \|$. Furthermore there exists an $\underline{e}^{(0)}$ such that equality holds in (15) so that for arbitrary $\underline{e}^{(0)}$ the quantity $\| T^r \|$ will give us a useful measure for comparing two iterations. We are thus led to define the average rate of convergence for an iteration with iteration matrix T over r iterations as

$$R(T^r) = -\frac{1}{r} \log \| T^r \| . \tag{16}$$

By inspection of (13) it is easily seen that

$$\lim_{r\to\infty} \frac{\|J^{r+1}\|}{\|J^{r}\|} = \rho(J) = \rho(T), \tag{17}$$

so that

$$\lim_{r\to\infty} R(T^{r}) = -\log[\rho(T)], \tag{18}$$

and this is defined as the asymptotic convergence rate.

Point Jacobi, Von Mises, Gauss-Seidel and SOR Iterations

Suppose that

$$A = M - N = D - L - U, \tag{19}$$

where D is diagonal, L is strictly lower triangular and U is strictly upper triangular.

In the Jacobi iteration we take M = D, giving

$$\begin{aligned} D\,\underline{x}^{(k+1)} &= D\,\underline{x}^{(k)} + [\underline{b} - A\,\underline{x}^{(k)}] \\ &= L\,\underline{x}^{(k)} + U\,\underline{x}^{(k)} + \underline{b}\,. \end{aligned} \tag{20}$$

This can be generalised by over-relaxation to Von Mises' iteration

$$D\,\underline{x}^{(k+1)} = D\,\underline{x}^{(k)} + w[\underline{b} - A\,\underline{x}^{(k)}], \tag{21}$$

where w is a constant. The matrices M and N are here given by

$$M = w^{-1}D\,, \quad N = w^{-1}D - A, \tag{22}$$

In the Gauss-Seidel iteration M = D - L giving the iteration

$$D\,\underline{x}^{(k+1)} = L\,\underline{x}^{(k+1)} + U\,\underline{x}^{(k)} + \underline{b}, \tag{23}$$

which can be generalised by over-relaxation to the form

$$D\,\underline{x}^{(k+1)} = D\,\underline{x}^{(k)} + w[L\,\underline{x}^{(k+1)} + U\,\underline{x}^{(k)} - D\,\underline{x}^{(k)} + \underline{b}]. \tag{24}$$

This method is known as successive over-relaxation and is usually abbreviated to S.O.R. The matrices M and N are now given by

$$M = w^{-1}D - L, \quad N = w^{-1}D + U - D. \tag{25}$$

In all these methods the elements of $\underline{x}^{(k+1)}$ are calculated one-by-one in the order $x_1^{(k+1)}$, $x_2^{(k+1)}$ ... $x_n^{(k+1)}$. All those elements required for the calculation of any element have already been found by that stage in the calculation. Gauss-Seidel and SOR have the advantage that enough storage for only one vector $\underline{x}$ is necessary, since once $x_{\bullet l}^{(k+1)}$ has been found $x_{\bullet l}^{(k)}$ is no longer needed and may be overwritten.

If A is symmetric we will wish to store D and U only but will require the whole 1^{th} row to find $x_{\bullet l}^{(k+1)}$. We overcome this difficulty by allocating enough storage for another vector and set this to zero at the beginning of each cycle. While we have the i^{th} row of U in use we also regard it as the i^{th} column of L, multiply it by $x_i^{(k)}$ for Jacobi and Von Mises or $x_i^{(k+1)}$ for Gauss-Seidel and SOR and add it to the special vector of storage. When we reach the stage of calculating $x_1^{(k+1)}$ we will already have the 1^{th} component of the vector $L\,\underline{x}^{(k)}$ or $L\,\underline{x}^{(k+1)}$ in the 1^{th} place of the special vector.

There remains the symmetric version of Gauss-Seidel, due to Aitken, and the symmetric version of SOR. If we replace (24) by

$$D\,\underline{x}^{(k+1)} = D\,\underline{x}^{(k)} + w(L\underline{x}^{(k)} + U\,\underline{x}^{(k+1)} - D\,\underline{x}^{(k)} + \underline{b}) \tag{26}$$

and calculate the elements in the order $x_n^{(k+1)}$, $x_{n-1}^{(k+1)}$, $x_{n-2}^{(k+1)}$ ... $x_1^{(k+1)}$, we obtain a very similar iteration. If we alternate between (24) and (26) we obtain the method of symmetric successive over-relaxation (SSOR). This has the advantage that if A is symmetric with non-zero diagonal elements, then the iteration matrix is similar to a symmetric and positive-definite matrix. To prove this we define the square root of a diagonal matrix. If D is the diagonal matrix given by

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix}$$

then

$$D^{\frac{1}{2}} = \begin{bmatrix} d_1^{\frac{1}{2}} & & & \\ & d_2^{\frac{1}{2}} & & \\ & & \ddots & \\ & & & d_n^{\frac{1}{2}} \end{bmatrix} \qquad (27)$$

and $D^{-\frac{1}{2}} = \left[D^{\frac{1}{2}}\right]^{-1} = \begin{bmatrix} d_1^{-\frac{1}{2}} & & & \\ & d_2^{-\frac{1}{2}} & & \\ & & \ddots & \\ & & & d_n^{-\frac{1}{2}} \end{bmatrix}$.

We are trying to solve

$$A\,\underline{x} = \underline{b}\,, \qquad (28)$$

or

$$D^{-\frac{1}{2}}\,A\,D^{-\frac{1}{2}}\,D^{\frac{1}{2}}\,\underline{x} = D^{-\frac{1}{2}}\,\underline{b}. \qquad (29)$$

If we iterate on $\underline{z} = D^{\frac{1}{2}}\,\underline{x}$ it is easily seen that throughout the iteration the correspondence

$$\underline{z}^{(r)} = D^{\frac{1}{2}}\,\underline{x}^{(r)} \qquad (30)$$

still holds. Suppose

$$D^{-\frac{1}{2}}A\,D^{-\frac{1}{2}} = I - L - U, \qquad (31)$$

then the iteration matrix for the SSOR process ~~is~~ on $\underline{z}^{(r)}$ is

$$\begin{aligned} T &= (I - wU)^{-1} (wL - wI + I)(I - wL)^{-1} (wU - wI + I) \\ &= C^{T^{-1}} ([2-w] I - C)C^{-1} (-C^T + [2-w] I), \end{aligned} \tag{32}$$

where

$$C = I - wL. \tag{33}$$

Rearranging,

$$\begin{aligned} T &= (C^T)^{-1} ([2-w]C^{-1} - I)([2-w][C^T]^{-1} - I)C^T \\ &= (C^T)^{-1} \, E \, E^T \, C^T \end{aligned} \tag{34}$$

where

$$E = [2-w]C^{-1} - I. \tag{35}$$

We conclude that T is similar to the positive-definite matrix $E\,E^T$ and since, using (30),

$$\underline{z}^{(k+1)} = D^{-\frac{1}{2}} \, T \, D^{\frac{1}{2}} \, \underline{z}^{(k)}, \tag{36}$$

our result is established. We will see later (Chapter 5) that this allows us to accelerate the convergence.

Termination of the Iteration and Estimation of the Error

As with direct methods of solution of linear equations, a reliable error bound cannot be found without some knowledge of A^{-1}. We assume that we have found the ~~special~~ spectral norm of A^{-1},

$$\| A^{-1} \| = \left[\rho \left(A^{-1T} A^{-1} \right) \right]^{\frac{1}{2}}, \tag{37}$$

where ρ denotes ~~special~~ spectral radius. No great accuracy is needed so the total time for the computation will not be increased substantially. It may in any case be found in the course of estimating relaxation parameters (see Chapters 3 and 5). Otherwise we may use the procedure of Chapter 8, applied to A for the positive-definite case and to

A^TA otherwise.

We need the residual vector for our bound. This is found anyway in the Jacobi and Von Mises iterations. An approximation to it is found anyway in the Gauss-Seidel and Jacobi methods, so additional work need be done only in the last few iterations. Suppose $\underline{r}^{(k)}$ is the exact residual of the set of equations

$$(A + \delta A)\underline{x} = \underline{b} + \delta\underline{b}, \tag{38}$$

that is

$$\begin{aligned} \underline{r}^{(k)} &= \underline{b} + \delta\underline{b} - (A + \delta A)\underline{x}^{(k)} \\ &= A(\underline{x} - \underline{x}^{(k)}) + \delta\underline{b} - \delta A\underline{x}^{(k)} \end{aligned} \tag{39}$$

and

$$\underline{e}^{(k)} = A^{-1}\left[\underline{r}^{(k)} - \delta\underline{b} + \delta A\,\underline{x}^{(k)}\right]. \tag{40}$$

The error bound

$$\|\underline{e}^{(k)}\| \leq \|A^{-1}\|\left[\|\underline{r}^{(k)}\| + \|\delta\underline{b}\| + \|\delta A\|\,\|\underline{x}^{(k)}\|\right] \tag{41}$$

follows at once.

An alternative method is applicable if we can neglect the effect of errors in the elements of A and $\underline{b}$. If the iteration matrix possesses a single dominant latent root, λ_1, and it is real and non-degenerate, then the displacement vector will be have asymptotically as

$$\underline{\Delta}^{(k)} = \lambda_1^k\,\underline{f} + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \tag{42}$$

where $\underline{f}$ is a constant vector and λ_2 is the sub-dominant latent root. Once the displacements show this behaviour, we may estimate the norm of the error as

$$\| \underline{e}^{(k)} \| \approx \frac{\| \underline{\Delta}^{(k)} \|}{1 - \lambda_1} \approx \frac{\| \underline{\Delta}^{(k)} \|^2}{\| \underline{\Delta}^{(k)} \| - \| \underline{\Delta}^{(k+1)} \|} \quad . \tag{43}$$

If the iteration matrix has more than one dominant latent root, the above analysis is inapplicable. However, we may average over a number of iterations by considering

$$\mu = \left[\frac{\| \underline{\Delta}^{(k+p)} \|}{\| \underline{\Delta}^{(k)} \|} \right]^{1/p} \quad . \tag{44}$$

If p is sufficiently large, μ approximates to the ~~special~~ spectral radius of the iteration matrix and we find the error estimate

$$\| \underline{e}^{(k)} \| \approx \frac{\| \underline{\Delta}^{(k)} \|}{1 - \mu} \tag{45}$$

These estimates should not be used too automatically. Machine rounding must ultimately destroy the asymptotic behaviour and the analysis will give accurate results only while these rounding errors are insignificant. The residual method would appear to be the safer.

Convergence of Von Mises Iteration

The Von Mises iteration matrix T is given by

$$\begin{aligned} T &= I - wD^{-1}A \\ &= (1-w)I + w D^{-1}(L + U) . \end{aligned} \tag{46}$$

Hence if ρ is the ~~special~~ spectral radius of the Jacobi iteration matrix, $D^{-1}(L + U)$, the iteration will converge if

$$|1 - w + \rho w| < 1 \text{ and } |1 - w - \rho w| < 1,$$

that is if $$\rho < \min (1, \tfrac{2}{w} - 1), \tag{47}$$

assuming that $0 < w < 2$.

We can obtain convergence if $\rho < 1$ and $0 < w < \frac{2}{1+\rho}$.

Now the spectral radius of T is bounded by

$$\alpha(w) = \max\{|1 - w + w\rho|, |1 - w - w\rho|\}, \tag{48}$$

and if $\delta > 0$ then

$$\begin{aligned} &\alpha(1) = \rho \\ \text{and} \quad &\alpha(1+\delta) \geq -1 + (1+\delta) + (1+\delta)\rho = \rho + \delta\rho + \delta > \rho \\ &\alpha(1-\delta) \geq 1 - (1-\delta) + (1-\delta)\rho = \rho + \delta(1-\rho) > \rho . \end{aligned} \tag{49}$$

It follows that if all that is known about the eigenvalues of $D^{-1}(L+U)$ is the value of ρ, then the best asymptotic rate of convergence is obtained at $w = 1$. Since the spectral radius of a matrix is not greater than either its row or Euclidean norm, convergence of the Jacobi process will be assured if either of these norms is less than unity. Thus we have that the Jacobi process is convergent for matrices that are diagonally dominant by rows. More can be said if A is positive definite. Since

$$D^{-1}A = D^{-\frac{1}{2}} D^{-\frac{1}{2}} A D^{-\frac{1}{2}} D^{\frac{1}{2}} , \tag{50}$$

we have that $D^{-1}A$ is similar to $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. But if A is positive definite then so is $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ since for any vector $\underline{x} \neq 0$

$$\begin{aligned} &\underline{x}^T D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \underline{x} = \underline{z}^T A \underline{z} > 0 \\ &\text{where} \quad \underline{z} = D^{-\frac{1}{2}} \underline{x} . \end{aligned} \tag{51}$$

Thus the roots of $D^{-1}A$ are real and positive and the iteration will converge if

$$w < \frac{2}{\sigma} , \tag{52}$$

where σ is the spectral radius of $D^{-1}A$. If the smallest latent root of $D^{-1}A$ is η, then the best asymptotic rate of convergence is given when

$$1 - w\eta = -(1 - w\sigma), \quad w = \frac{2}{\sigma + \eta} \tag{53}$$

One special case is sometimes quoted. The Jacobi iteration will converge if A is positive definite and all the elements of $D^{-1}(L+U)$ are non-negative. This result depends on the Perron-Frobenius theory of non-negative matrices. The lemma we require is that the matrix $D^{-1}(L+U)$ has a positive eigenvalue equal to its spectral radius. This is proved, for example, by Varga (1962, page 46). The matrix $D^{-1}A$ has all its eigenvalues positive, so the greatest positive root of $D^{-1}(L+U)$ is less than unity. From the lemma it follows that its spectral radius is less than unity so that the Jacobi process will converge.

Convergence of the Iterations for a Small Relaxation Parameter

If w is small, the SOR and Von Mises iterations converge equally fast since the SOR iteration matrix,

$$L_w = (I - w D^{-1}L)^{-1} (w D^{-1}U - (w-1) I) \tag{54}$$

may be reduced by some manipulation to

$$\begin{aligned} L_w &= (I + w D^{-1}L)(w D^{-1}U - (w-1) I) + O(w^2) \\ &= (I - w)I + w D^{-1}U + w D^{-1}L + O(w^2) \\ &= I - w D^{-1}A + O(w^2). \end{aligned} \tag{55}$$

If the eigenvalues of $D^{-1}A$ are $\lambda_k + i\,\mu_k$ then those of L_w are

$$\upsilon_k = 1 - w(\lambda_k + i\,\mu_k) + O(w^2) \tag{56}$$

and

$$\begin{aligned} |\upsilon_k|^2 &= (1 - w\lambda_k)^2 + w^2\mu_k^2 + O(w^2) \\ &= 1 - 2w\lambda_k + O(w^2). \end{aligned} \tag{57}$$

It follows that the processes will converge for sufficiently small positive w if the real parts of the eigenvalues of $D^{-1}A$ are positive.

An alternative criterion for convergence if w is sufficiently small is that $(A + A^T)$ be positive definite. This implies that the elements of D are positive. Suppose that $D^{\frac{1}{2}}$ is the diagonal matrix whose elements are the square roots of those of D, then L_w is similar to the matrix

$$D^{\frac{1}{2}} T D^{-\frac{1}{2}} = I - w D^{-\frac{1}{2}} A D^{-\frac{1}{2}} + O(w^2). \tag{58}$$

Now the spectral radius of a matrix is not greater than its spectral norm. Hence T will be convergent if the matrix

$$\begin{aligned}&(D^{\frac{1}{2}} T D^{-\frac{1}{2}})(D^{\frac{1}{2}} T D^{-\frac{1}{2}})^T \\ &= I - w D^{-\frac{1}{2}} A D^{-\frac{1}{2}} - w D^{-\frac{1}{2}} A^T D^{-\frac{1}{2}} + O(w^2)\end{aligned} \tag{59}$$

is convergent. But $(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} + D^{-\frac{1}{2}} A^T D^{-\frac{1}{2}})$ is positive definite if $(A + A^T)$ is positive definite. Our result is therefore established. This result is given by Broyden (1964), using a different proof.

Convergence of SOR and SSOR for Positive-definite Matrices

Consider one stage in the iteration, where the i^{th} element of $\underline{x}^{(k+1)}$ is calculated. Suppose the corresponding error vectors are $\underline{e}^{(k,i)}$ and $\underline{e}^{(k,i+1)}$, then

$$\underline{e}^{(k,i+1)} = \underline{e}^{(k,i)} - \frac{w \underline{r}_i^{(k,i)}}{a_{ii}}, \tag{60}$$

where $A = [a_{ij}]$ and $\underline{r}_i^{(k,i)}$ in the column vector whose i^{th} element is the i^{th} element of the residual vector and whose other elements are all zero. Consider the quadratic form $\underline{e}^{(k,i)^T} A \underline{e}^{(k,i)}$. Using (60), we find the relation

$$\begin{aligned}\underline{e}^{(k,i+1)^T} A \underline{e}^{(k,i+1)} &= \underline{e}^{(k,i)^T} A \underline{e}^{(k,i)} - \frac{w}{a_{ii}} \Big[\underline{r}_i^{(k,i)^T} \underline{e}^{(k,i)} + \underline{e}^{(k,i)^T} \underline{r}_i^{(k,i)} \\ &\qquad - \frac{w}{a_{ii}} \underline{r}_i^{(k,i)^T} A \underline{r}_i^{(k,i)} \Big] \\ &= \underline{e}^{(k,i)^T} A \underline{e}^{(k,i)} - \frac{w}{a_{ii}} (2-w) \underline{r}_i^{(k,i)^T} \underline{r}_i^{(k,i)}\end{aligned} \tag{61}$$

and if $0 < w < 2$ then

$$\underline{r}^{(k,i+1)T} A \underline{r}^{(k,i+1)} < \underline{r}^{(k,1)T} A \underline{r}^{(k,i)} \tag{62}$$

The sequence $[\underline{r}^{(k)T} A \underline{r}^{(k)}]$ is therefore monotonic decreasing. It is a sequence of positive numbers since A is positive definite and therefore converges to a limit. The difference between $\underline{e}^{(k,i)T} A \underline{e}^{(k,i)}$ and $\underline{r}^{(k,i+1)T} A \underline{r}^{(k,i+1)}$ will become arbitrarily small, so that from (61) it follows that $r_1^{(k,1)} \to 0$ and convergence is assured.

Conversely, if A is symmetric and has a negative eigenvalue, there exists a vector $\underline{e}^{(0)}$ such that

$$\underline{e}^{(0)T} A \underline{e}^{(0)} < 0. \tag{63}$$

But the sequence $[\underline{e}^{(k)T} A \underline{e}^{(k)}]$ is monotonic decreasing and so cannot converge to zero. The process is therefore not convergent for arbitrary $\underline{r}^{(0)}$.

Convergence of SOR for Near-symmetric Matrices

The SOR iteration matrix is

$$L_w = (D - wL)^{-1} (wU - (w-1) D). \tag{64}$$

Using the notation of equations (4), (6) and (7) we find the relations

$$\underline{e}^{(k)} = (I - L_w)^{-1} \underline{\Delta}^{(k)} \tag{65}$$

and

$$\underline{r}^{(k)} + \underline{r}^{(k+1)} = (I + L_w)(I - L_w)^{-1} \underline{\Delta}^{(k)}. \tag{66}$$

Now

$$(I + \mathcal{L}_w) = (D - wL)^{-1} (wU - wL - (w-2) D) \tag{67}$$

and

$$(I - \mathcal{L}_w) = (D - wL)^{-1} (-wU - wL + wD)$$

$$= w(D - wL)^{-1} A \tag{68}$$

from which it follows that

$$(I + \mathcal{L}_w)(I - \mathcal{L}_w)^{-1} = (I - \mathcal{L}_w)^{-1} (I + \mathcal{L}_w)$$

$$=(wA)^{-1} (wU - wL - (w-2) D). \tag{69}$$

Hence the quantity

$$Q = \underline{x}^{(k)T} A \underline{x}^{(k)} - \underline{x}^{(k+1)T} A \underline{x}^{(k+1)}$$

$$= \tfrac{1}{2} \underline{x}^{(k)} (A + A^T) \underline{x}^{(k)} - \tfrac{1}{2} \underline{x}^{(k+1)T} (A + A^T) \underline{x}^{(k+1)}$$

$$= \tfrac{1}{2} (\underline{x}^{(k)T} - \underline{x}^{(k+1)T})(A + A^T)(\underline{x}^{(k)} + \underline{x}^{(k+1)})$$

$$= \tfrac{1}{2} \underline{\Delta}^{(k)T} (A + A^T)(wA)^{-1} (wU - wL - (w-2)D) \underline{\Delta}^{(k)} . \tag{70}$$

If A is symmetric this quadratic form reduces to

$$Q = w^{-1}(w-2) \underline{\Delta}^{(k)T} D \underline{\Delta}^{(k)} \tag{71}$$

and is positive if A is positive definite. This provides an alternative proof of the convergence of SOR for $0 < w < 2$ for positive-definite matrices. If A is near-symmetric so that

$$\| A - A^T \| = \varepsilon, \tag{72}$$

then it is easily seen that

$$Q = w^{-1}[(\frac{2}{w} - 1) \underline{\Delta}^{(k)T} D \underline{\Delta}^{(k)} + O(\varepsilon)]. \tag{73}$$

If ε is small and the elements of D are positive, Q will be positive if

$$0 < w < \alpha < 2, \tag{74}$$

where α is a constant depending on A. With w in this range we will have convergence if and only if $(A + A^T)$ is positive-definite. The argument runs in exact parallel with that for a symmetric A.

This result was established by Stein (1951) for the case w = 1. He states explicitly which matrix must be positive definite in order that $Q > 0$ for any $\underline{\Delta}^{(k)}$. We feel, however, that this will be of little practical value. The result is useful in a practical situation in that if we know that $\| A - A^T \|$ is smaller and that $(A + A^T)$ is positive definite, then it is worth trying the iteration with w not too close to 2. Conversely, if $\| A - A^T \|$ is small and $(A + A^T)$ is not positive definite, there is no point in trying SOR.

Comparison Theorem of Stein and Rosenberg

If the Jacobi iteration matrix is non-negative, the Stein-Rosenberg theorem provides a direct comparison between Jacobi and Gauss-Seidel iterations. If their spectral radii are ρ_J and ρ_G respectively, then one and only one of the following relations holds:

$$
\begin{aligned}
&1. \quad \rho_J = \rho_G = 0 \\
&2. \quad 0 < \rho_G < \rho_J < 1 \\
&3. \quad 1 = \rho_G = \rho_J \\
&4. \quad 1 < \rho_J < \rho_G .
\end{aligned}
\tag{75}
$$

This is established by Varga (1962, Page 70). Varga also shows that the result of page (24), that the Jacobi process is convergent for matrices that are diagonally dominant by rows, can be extended to Gauss-Seidel also.

Kaczmarz iteration

Kaczmarz (1937) proposed an iteration that is convergent for any non-singular matrix. If A is partitioned by rows and b by elements in the form

$$A = \begin{bmatrix} \underline{a}_1^T \\ \vdots \\ \underline{a}_n^T \end{bmatrix}, \qquad \underline{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \tag{76}$$

then the individual equations can be written

$$\underline{a}_i^T \underline{x} = b_i . \tag{77}$$

Each of these equations can be thought to represent a hyperplane in n-dimensional space and their meet corresponds to the required solution. Let the iterate $\underline{x}^{(k,i)}$ corresponds to the point $P^{(k,i)}$ in this space. We obtain $P^{(k,i+1)}$ as the foot of the perpendicular from $P^{(k,i)}$ to the i^{th} plane. The normal to this plane that passes through $P^{(k,i)}$ is given by

$$\underline{z} = \underline{x}^{(k,i)} + \underline{a}_i \lambda, \tag{78}$$

and a point on this line lies in the plane if

$$\underline{a}_i^T (\underline{x}^{(k,i)} + \underline{a}_i \lambda) = b_i, \tag{79}$$

that is if

$$\lambda = (\underline{a}_i^T \underline{a}_i)^{-1} (b_i - \underline{a}_i^T \underline{x}^{(k,i)}). \tag{80}$$

It follows that

$$\underline{x}^{(k,i+1)} = \underline{x}^{(k,i)} + (\underline{a}_i^T \underline{a}_i)^{-1} (b_i - \underline{a}_i^T \underline{x}^{(k,i)}) \underline{a}_i . \tag{81}$$

The cycle is completed by setting

$$\underline{x}^{(k+1,1)} = \underline{x}^{(k,m+1)} . \tag{82}$$

Since A is non-singular there is a unique solution and we can define a distance function

$$d^{(k,i)} = [\,(\underline{x} - \underline{x}^{(k,i)})^T\,(\underline{x} - \underline{x}^{(k,i)})\,]^{\frac{1}{2}}\,, \tag{83}$$

which is just the distance from $p^{(k,i)}$ to the solution. Of course $d^{(k,i)}$ is positive and by Pythagoras' theorem

$$d^{(k,i+1)} \leq d^{(k,i)}\,. \tag{84}$$

Hence the sequence $\left[d^{(k,i)}\right]$ is monotonic decreasing and bounded by zero and therefore must converge to a limit. Given any positive ϵ there must exist a k_0 such that

$$\left|d^{(k,i)} - d^{(k,i+1)}\right| < \epsilon \tag{85}$$

for $k \geq k_0$, $i = 1, 2 \ldots n$. If $\delta^{(k,i)}$ is the distance between $p^{(k,i)}$ and $p^{(k,i+1)}$ then

$$\begin{aligned}\delta^{(k,i)^2} &= d^{(k,i)^2} - d^{(k,i+1)^2} \\ &< (d^{(k,i+1)} + \epsilon)^2 - d^{(k,i+1)^2} \\ &= \epsilon(2d^{(k,i+1)} + \epsilon) \\ &\leq \epsilon(2d^{(1,1)} + \epsilon)\,.\end{aligned} \tag{86}$$

Since any plane is reached in not more than (n-1) steps, the distance to any plane cannot be greater than $(n-1)\sqrt{\epsilon(2d^{(1,1)} + \epsilon)}$. We have therefore proved convergence.

It can be shown (Darwent, 1963) that the rate of convergence is identical with that of Gauss-Seidel applied to the equations

$$B\,\underline{x} = \underline{g}, \quad B = AA^T\,. \tag{87}$$

In the Kaczmarz procedure, the residuals are given by

$$r_j^{(k,i)} = b_j - \underline{a}_j^T\,\underline{x}^{(k,i)}\,. \tag{88}$$

Writing a similar equation for the next step and using (31) gives

$$r_j^{(k,i+1)} = b_j - \underline{a}_j^T\,(\underline{x}^{(k,i)} + [\underline{a}_i^T\,\underline{a}_i]^{-1}[b_i - \underline{a}_i^T\,\underline{x}^{(k,i)}]\,\underline{a}_i)\,. \tag{89}$$

or

$$r_j^{(k,i+1)} = r_j^{(k,i)} - [a_i^T a_i]^{-1} \underline{a}_j^T \underline{a}_i \, r_i^{(k,i)} \, . \tag{90}$$

In the Gauss-Seidel iteration applied to (87) the residuals satisfy the equation

$$r_j^{(k,i+1)} = r_j^{(k,i)} - b_{ii}^{-1} \, b_{ji} \, r_i^{(k,i)} \, . \tag{91}$$

But

$$b_{ji} = \underline{a}_j^T \underline{a}_i \, , \tag{92}$$

so that (90) and (91) are identical, proving our result.

To find the vector $\underline{x}^{(k,i+1)}$ as many changes have to be made to the vector $\underline{x}^{(k,i)}$ as and there are non-zero elements in $\underline{a}_i$. The work per step is therefore about double that of Gauss-Seidel. Furthermore, in the case of a positive-definite matrix A the convergence is likely to be slower since AA^T is worse-conditioned than A. However it does provide a method whose convergence is guaranteed for a general matrix. The Gauss-Seidel method applied to the equations

$$A^T A \, \underline{x} = A^T \underline{b} \tag{93}$$

will converge ~~at the same rate~~ and we may use over-relaxation but the system is less well-conditioned and will require more storage. Kaczmarz iteration does not suffer this disadvantage since it uses the residuals of the original set of equations.

Kaczmarz iteration may be extended to an over-relaxed version, given by the equation

$$\underline{x}^{(k,i+1)} = \underline{x}^{(k,i)} + w[\underline{a}_i^T \underline{a}_i]^{-1} \, [b_i - \underline{a}_i^T \underline{x}^{(k,i)}] \, \underline{a}_i \, . \tag{94}$$

Our proof of convergence can be extended to cover this iteration for $0 < w < 2$. Also there is a correspondence with SOR on the matrix

A^TA of exactly the same form as that for w=1. We do not know of any method other than experiment to find the best value of w.

There also exists a symmetric Kaczmarz iteration. Equation (94) may be written

$$\underline{x}^{(k,i+1)} = S_i \, \underline{x}^{(k,i)} + w \, b_i [\underline{a}_i^T \, \underline{a}_i]^{-1} \, \underline{a}_i \tag{95}$$

where

$$S_i = I - w \, \frac{\underline{a}_i \, \underline{a}_i^T}{\underline{a}_i^T \, \underline{a}_i} \tag{96}$$

If we iterate in symmetric cycles, that is replace

$$\underline{x}^{(k+1,1)} = \underline{x}^{(k,n+1)} \tag{97}$$

by

$$\underline{x}^{(k+1,1)} = \underline{x}^{(k,2n+1)} \tag{98}$$

and use

$$\underline{x}^{(k,i+1)} = S_{2n-i+1} \, \underline{x}^{(k,i)} + w b_j [\underline{a}_j^T \, \underline{a}_j] \underline{a}_j , \tag{99}$$

$j = 2n - i + 1,$

for i = n+1, ... 2n then the iteration matrix is

$$T = (S_1 \, S_2 \cdots S_n)(S_n \cdots S_2 \, S_1). \tag{100}$$

Now S_i is symmetric so that

$$\begin{aligned} T &= (S_1 \, S_2 \cdots S_n)(S_n^T \cdots S_2^T \, S_1^T) \\ &= (S_1 \, S_2 \cdots S_n)(S_1 \cdots S_{n-1} \, S_n)^T \end{aligned} \tag{101}$$

which is symmetric and positive definite. We shall see that this has considerable advantages, as has SOR.

CHAPTER 3

THE ESTIMATION OF RELAXATION PARAMETERS

Matrices with Property A

The matrix A is often of such a form that there exists a permutation matrix P such that

$$P^{-1} A P = \begin{bmatrix} D_1 & -A_1 \\ -A_2 & D_2 \end{bmatrix} \qquad (1)$$

where D_1 and D_2 are diagonal matrices. The SOR method applied to the matrix $P^{-1}AP$ has iteration matrix

$$L_w = \begin{bmatrix} D_1 & \\ -wA_2 & D_2 \end{bmatrix}^{-1} \begin{bmatrix} (1-w)\, D_1 & wA_1 \\ & (1-w)\, D_2 \end{bmatrix} . \qquad (2)$$

Suppose it has an eigenvalue λ and corresponding eigenvector $\underline{Y}$, so that

$$\begin{bmatrix} (1-w)\, D_1 & wA_1 \\ & (1-w)\, D_2 \end{bmatrix} \underline{Y} = \lambda \begin{bmatrix} D_1 & \\ -wA_2 & D_2 \end{bmatrix} \underline{Y} , \qquad (3)$$

that is

$$w \begin{bmatrix} & A_1 \\ \lambda A_2 & \end{bmatrix} \underline{Y} = (\lambda + w - 1) \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} \underline{Y} . \qquad (4)$$

Now suppose that

$$G = \begin{bmatrix} I & \\ & \lambda^{\frac{1}{2}} I \end{bmatrix} \qquad (5)$$

so that (4) can be written as

$$\lambda^{\frac{1}{2}} wG \begin{bmatrix} & A_1 \\ A_2 & \end{bmatrix} G^{-1}\underline{y} = (\lambda+w-1) \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} \underline{y} \quad . \tag{6}$$

But the Jacobi iteration matrix is

$$B = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}^{-1} \begin{bmatrix} & A_1 \\ A_2 & \end{bmatrix} \tag{7}$$

Thus we have shown that $G^{-1}\underline{y}$ is an eigenvector of the Jacobi matrix and that the eigenvalues μ of B are related to the eigenvalues λ of L_w by the equation

$$(\lambda + w - 1)^2 = \lambda w^2 \mu^2 , \tag{8}$$

provided $w \neq 0$ and $\lambda \neq 0$. The case $w = 0$ is of no interest to us. If $\lambda = 0$, then $w = 1$ from equation (3) since the elements of D_1 and D_2 must be non-zero for SOR iteration to be applicable. Equation (8) is therefore valid for $\lambda = 0$. We have shown, incidentally, that for a matrix of the form (1) that is symmetric and positive-definite, the Jacobi iteration converges. This follows from (8) and the fact that SOR converges for $0 < w < 2$.

We are now led to ask whether the result holds for any other ordering, that is for any permutation matrix other than P. As in (19) of Chapter 2 we write

$$A = D - L - U, \tag{9}$$

where D is diagonal and L and U are strictly lower and upper triangular. It is easily seen that the result remains valid provided, for any non-zero λ, there exists a non-singular diagonal matrix G such that

$$U + \lambda L = \lambda^{\frac{1}{2}} G (L + U) G^{-1} . \tag{10}$$

Young (1954) defined the matrix A as having "property A" if there exists a vector $\underline{q} = (q_1, q_2, \ldots q_n)^T$, with integral coefficients, such that if $a_{ij} \neq 0$ and $i \neq j$ then $|q_i - q_j| = 1$.

The ordering is consistent if, whenever $a_{ij} \neq 0$ and $q_i > q_j$, the i^{th} row follows the j^{th} row in the ordering, and whenever $a_{ij} \neq 0$ and $q_i < q_j$, the j^{th} row follows the i^{th} row in the ordering.

If we set $G = \mathrm{diag}(\lambda^{q_i})$, it is easily seen that such a matrix satisfies (10).

Young (1962) and Fox (1962) have each rephrased this definition in an attempt to make it more understandable. Young says that an ordering is consistent if, starting from this ordering, we can permute the rows and corresponding columns in such a way that if $a_{ij} \neq 0$ then the ordering relation between the i^{th} and j^{th} rows is unchanged and so that the permutated matrix is block tridiagonal. Fox says that the ordering is consistent if the arithmetic performed in the iteration is identical with that performed if the iteration is applied to the matrix $P^{-1}AP$, where P is a permutation matrix and $P^{-1}AP$ is block tridiagonal.

Unfortunately, these simplified definitions are insufficiently general. We will see at the end of this chapter that elliptic differential equations in three dimensions can give rise to matrices with orderings that satisfy (10) without satisfying them. Varga (1962) uses a definition that is equivalent to (10), namely that for any λ

$$\det \left[D + U + L \right] = \det \left[D + \lambda^{-1} U + \lambda L \right]. \qquad (11)$$

Varga (1962) has generalized these results to cases where the Jacobi matrix is cyclic of order p. Property A is the special case of $p = 2$. However, in the writer's experience, cases with

$p > 2$ are very uncommon, and they will not be considered here.

Dependence of the Asymptotic Convergence Rate on the Iteration Parameter

If the matrix A has property A and is consistently ordered and the Jacobi matrix has real roots, it is well-known that an analysis of (8) shows that the asymptotic convergence rate of the SOR iteration is optimized with relaxation parameter

$$w_{opt} = \frac{2}{1 + \left[1 - \mu^2\right]^{1/2}}, \qquad (12)$$

where μ is the greatest root of the Jacobi matrix. This is verified by Varga (1962, page 110). As w varies, the eigenvalues of the SOR matrix behave as follows:

$0 < w < 1$	All eigenvalues real and positive.
$w = 1$	One of the roots of (8) vanishes, so that the eigenvalues are either zero or equal to μ_i^2.
$1 < w < w_{opt}$	At least one eigenvalue is real. The complex eigenvalues have modulus (1-w).
$w_{opt} < w < 2$	All eigenvalues have modulus (1-w) and are complex.

These properties are easily established by examination of equation (8). If the chosen w is greater than w_{opt}, the spectral radius of the iteration will be (1-w) so that $\frac{d|\lambda|}{dw} = 1$. A little analysis of of equation (8) shows that as w tends to w_{opt} from below, $\frac{d|\lambda|}{dw}$ tends to infinity. It follows that it is very important to estimate w_{opt} accurately, particularly if it is near 2, and that it is better to over-estimate than under-estimate. We illustrate this with some numerical examples in table 1.

$\mu = .9$			$\mu = .99$			$\mu = .999$		
w	λ	-log λ	w	λ	-log λ	w	λ	-log λ
1.300	.625	.470	1.730	.836	.180	1.890	.959	.042
1.350	.556	.587	1.740	.818	.201	1.900	.951	.050
1.393	.393	.934	1.750	.785	.242	1.910	.938	.065
1.400	.400	.916	1.753	.753	.284	1.914	.914	.089
1.450	.450	.799	1.760	.760	.274	1.920	.920	.083
1.500	.500	.693	1.770	.770	.261	1.930	.930	.073

TABLE 1

The dependence of the convergence on the choice of w near $w = w_{opt}$.

Unless we have an a prieri estimate of w_{opt}, we shall need an algorithm that finds it without increasing unduly the total amount of work. If $w \geq w_{opt}$ all the roots of the iteration matrix have equal moduli and as w is decreased from w_{opt}, the dominant root increases while the modulus of the remaining roots decrease. In fact if the second root of the Jacobi matrix is μ', the ratio of dominant to subdominant root of the SOR iteration matrix will have a peak at

$$w'_{opt} = \frac{2}{1 + \left[1 - \mu'^2\right]^{1/2}} \tag{13}$$

Unfortunately it seems difficult to exploit this fact, but both Kulsrud (1961) and Carré (1961) describe useful algorithms for calculating w_{opt} and depend on the use of a relaxation factor slightly less than w_{opt}.

Carré's technique is to iterate a few times with a w that is a deliberate underestimate of the optimum and to estimate the dominant eigenvalue by the ratio of the norms of the last two changes in the vector. From this [redacted] he obtains an estimate, w', of w_{opt}, using (8), and continues the iteration with relaxation factor

$$w_m = w' - \tfrac{1}{4}(2 - w'). \tag{14}$$

He states that in his experience this is likely to converge to a value of w not far from w'_{opt}. When the changes in w' are small, he continues the iteration with $w = w'$. He claims that the total number of iterations required by this technique rarely exceeds by more than 35 % the number required if w_{opt} is used throughout. Kulsrud's method is essentially the same, except that he uses the latest estimate of w_{opt} rather than Carré's w_m. These estimates will steadily rise, for if we use w_0 and find an approximation λ_0 to the dominant root of L_w then the new estimate will be

$$w_1 = \frac{2}{1 + \left[1 - \dfrac{(w_0 - 1 + \lambda_0)^2}{w_0^2 \lambda_0}\right]^{1/2}} \tag{15}$$

It is easily shown that $w_1 \geq w_0$, for

$$w_1 = \frac{2}{1 + \left[1 - \dfrac{(w_0 - 1 - \lambda_0)^2}{w_0^2 \lambda_0} - \dfrac{4(w_0 - 1)}{w_0^2}\right]^{1/2}}$$

$$\geq \frac{2}{1 + \left[\dfrac{w_0^2 - 4w_0 + 4}{w_0^2}\right]^{1/2}}$$

$$= \frac{2}{1 + \dfrac{2 - w_0}{w_0}} = w_0 . \tag{16}$$

Thus the sequence of estimates will steadily increase. It is clear that an underestimate must be used to start the procedure and that there is some danger that a gross overestimate will be found. The estimation procedure is stopped when w_0 and w_1 agree to a prescribed accuracy. Kularud does not make clear how he gets an accurate

value of λ_o when w_o is near w_{opt}. Since there will be a number of complex eigenvalues all with modulus (w-1) any inspection of the norms of successive changes may give a very poor value of λ_o. If, a ~~large number of iterations are used we have~~

$$\lambda_o \approx \left\{ \frac{\| \underline{\Delta}^{(k+p)} \|}{\| \underline{\Delta}^{(k)} \|} \right\}^{1/p} , \tag{17}$$

with p of moderate size, we will be less likely to get a very poor result. Kulsrud states that in three test problems he required no more iterations than if the optimum parameter were used throughout.

Both the ~~techniques~~ of Carré and that of Kulsrud require that the Jacobi matrix should have real roots. The most usual case is where the matrix A is symmetric. Here, in the notation of (9), the Jacobi matrix is $D^{-1}(L + U)$ and if we have an approximate eigenvector $\underline{z}$ with error $O(\epsilon)$, its Rayleigh quotient

$$\mu_R = \frac{\underline{z}^T (L + U)\underline{z}}{\underline{z}^T D \underline{z}} \tag{18}$$

will be an approximate eigenvalue with error $O(\epsilon^2)$. Moreover

$$| \mu_R | \leq \rho[D^{-1}(L + U)]. \tag{19}$$

Neither Carré nor Kulsrud exploit these properties. We will now describe a technique based on (18) and (19).

In a typical step we iterate a few times with a relaxation parameter w and form the vector

$$\underline{z}^{(k)} = G^{-1} \underline{\Delta}^{(k)} , \tag{20}$$

where $\underline{\Delta}^{(k)}$ is the last displacement vector and G is the matrix of equation (10) with λ approximated by (w-1). We form the Rayleigh quotient

$$\mu_R^{(k)} = \frac{\underline{z}^{(k)T} (L + U) \underline{z}^{(k)}}{\underline{z}^{(k)T} D \underline{z}^{(k)}} \qquad (21)$$

and hence a new relaxation parameter

$$w^{(k)} = \frac{2}{1 + \left[1 - \mu_R^{(k)2}\right]^{1/2}} . \qquad (22)$$

The iteration is continued with the greater of W, $w^{(k)}$. We start by forming the Rayleigh quotient of an arbitrary vector $\underline{z}$, continue until successive values of w show little change and then revert to straightforward SOR.

This technique has several advantages. In common with Kulsrud we are always using the best value of w currently available and in common with Carré we always use an under-estimate, so that our estimate will continue to improve until it is contaminated with rounding errors. The current value of w is likely to be good since we are using the Rayleigh quotient of a symmetric matrix and no prior knowledge of w_{opt} is needed.

The only disadvantage is that some additional work and storage are involved in the formation of the Rayleigh quotient. For a general matrix we need a whole vector of additional storage but for a matrix of band-width r only r storage locations are required. The additional work in finding each Rayleigh quotient is comparable with one SOR iteration. We can hope for adequate compensation in the improved convergence.

SOR without property A

If the matrix does not possess property A (or Varga's generalisation), then we have to rely on trial and error to estimate the best value of w. A theorem of Kahan (1958) helps a little in this. He shows that if SOR iteration with parameter w is used, then the spectral radius of the iteration matrix is not less

than $|w-1|$ and equality is possible only if all the eigenvalues of the iteration matrix have modulus $|w-1|$.

To prove this theorem we consider the iteration matrix

$$L_w = (I - wD^{-1}L)^{-1}\,(wD^{-1}U + (1-w)\,I), \tag{23}$$

in the usual notation. Since these two matrices are triangular we find

$$\det(I - wD^{-1}L) = 1, \quad \det(wD^{-1}U + (1-w)\,I) = (1-w)^n \tag{24}$$

and hence that

$$\det(L_w) = (1-w)^n . \tag{25}$$

But $\det(L_w)$ is the product of its eigenvalues and the result follows at once.

This theorem is useful in that if we have found the spectral radius ρ_1 of L_{w_1} we need search only in the range $1-\rho_1 < w < 1+\rho_1$. Any search of this nature can be a time consuming operation, particularly if we should encounter a complex dominant root. As the example of matrices with property A shows, the converge may be quite critically dependent on a good choice of w. For these reasons we prefer other acceleration techniques, to be described below, but these all require more storage than the one vector needed by SOR.

SSOR Parameter

If the matrix A is symmetric we can find the best value of the relaxation parameter for SSOR by the technique described by Habetler and Wachpress (1961). We first manipulate the iteration matrix to the form

$$M_w = (D-wU)^{-1}\,(wL - (w-1)D)\,(D - wL)^{-1}\,(wU - (w-1)\,D)$$
$$= [I - w(D-wU)^{-1}\,A]\,[I - w\,(D-wL)^{-1}\,A]$$
$$= I - w\,[D-wU]^{-1}\,[D-wL - wA + D - wU]\,(D-wL)^{-1}\,A$$
$$= I - w\,(2-w)(D - wU)^{-1}D\,(D - wL)^{-1}\,A\,. \qquad (26)$$

If $\underline{z}$ is the eigenvector of M_w corresponding to its dominant eigenvalue, then this eigenvalue is given by the Rayleigh quotient

$$\lambda_R = 1 - \frac{w(2-w)\,\underline{z}^T A\,\underline{z}}{\underline{z}^T(D-wL)D^{-1}(D-wU)\,\underline{z}}\,. \qquad (27)$$

Now a small perturbation of $\underline{z}$ will cause a second-order perturbation in λ_R. Hence $\frac{d\lambda_R}{dw}$ will not involve any terms in $\frac{d\underline{z}}{dw}$. Now writing

$$k = \underline{z}^T L\,D^{-1}U\,\underline{z}$$
$$\tau = \underline{z}^T (L+U)\,\underline{z} \qquad (28)$$
$$\delta = \underline{z}^T D\,\underline{z}$$

it follows from (27) that

$$\lambda_R = 1 - \frac{w(2-w)(\delta - \tau)}{\delta - w\tau + w^2 k}\,, \qquad (29)$$

and $\frac{d\lambda_R}{dw} = 0$ if

$$(\delta - w\tau + w^2k)(2w - 2) + w\,(2-w)(-\tau + 2kw) = 0$$

that is
$$2kw^2 - \tau w^2 + 2w\delta - 2\delta = 0. \qquad (30)$$

Hence the optimum relaxation parameter is given by

$$\frac{1}{w_{opt}} = \frac{-2\delta - \left[4\delta^2 + 8\delta\,(2k - \tau)\right]^{1/2}}{-4\delta} \qquad (31)$$

or

$$w_{opt} = \frac{2}{1 + \left[1 + \frac{4k}{\delta} - \frac{2\tau}{\delta}\right]^{1/2}}\,. \qquad (32)$$

We take the positive radical since we know that for convergence $0 < w < 2$. This must give the minimum of λ_R since $\lambda_R > 1$ for w outside the range [0, 2] and $\lambda_R < 1$ inside it. It should be noticed that

$$\begin{aligned} \beta &= 1 + \frac{\Delta k}{\delta} - \frac{2\tau}{\delta} \\ &= \frac{\underline{z}^T(D - 2L)D^{-1}(D - 2L)\underline{z}}{\underline{z}^T D \underline{z}} \\ &\geq 0 \end{aligned} \tag{33}$$

and if A and consequently D is positive definite.

Evans and Forrington (1963) have described an iterative process for finding w_{opt}. After each iteration they calculate β by (33) and then replace w by

$$\frac{2}{1 + \beta^{\frac{1}{2}}} \tag{34}$$

for the next iteration and continue until w has settled down. They point out that the spectral radius of the iteration matrix is not very critically dependent on small changes in w near w_{opt}. This is because w_{opt} corresponds to a minimum in a continuous function. It is therefore unnecessary to estimate w_{opt} very accurately. But they fail to point out that if we wish to use Chebyshev extrapolation, to be described in Chapter 5, an accurate estimate of the spectral radius of the iteration is necessary. Furthermore it is better to over-estimate than under-estimate. Their procedure is to take the spectral radius as the last value of the ratio

$$\frac{\| \underline{\Delta}^{(k+1)} \|}{\| \underline{\Delta}^{(k)} \|} \tag{35}$$

where the Euclidean norm is used. This, however, is not the Rayleigh quotient and in general will give a poorer estimate than that obtained from (27).

An Example

We conclude this chapter with an example to show just how powerful an iterative method can be. We consider the solution of the elliptic partial differential equation

$$\nabla^2\phi = 0 \tag{36}$$

over the unit cube with ϕ given on the boundary. If we use the simple seven-point finite-difference approximation with step-width $\frac{1}{s+1}$ we find the matrix equation

$$A\underline{x} = \underline{b}, \tag{37}$$

where A has elements given by

$$\left.\begin{aligned} a_{ij} &= -\frac{1}{6} \text{ if } |i-j| = 1 \text{ or } s \text{ or } s^2 \\ a_{ii} &= +1 \\ a_{ij} &= 0 \text{ otherwise.} \end{aligned}\right\} \tag{38}$$

The order of the matrix A is s^3 and the band-width $2s^2 + 1$. It is positive definite. From table 2 of Chapter 1 we note that the number of arithmetic operations in the direct solution of equation (37) is $\frac{1}{2}s^7(\mu + \alpha) + O(s^5)$.

The smallest eigenvalue is

$$\lambda_1 = 1 - \cos\frac{\pi}{s}, \tag{39}$$

and hence the dominant root of the Jacobi matrix is

$$\mu_1 = \cos\frac{\pi}{s}. \tag{40}$$

The matrix is consistently ordered with property A. This follows by taking Young's ordering vector as $(q_1, q_2, \ldots, q_n)^T$ where

$$q_{is^2+js+k} = i + j + k, \tag{41}$$

i, j, k all less than s. We set $G = \text{diag}(\lambda^{\frac{1}{2}q_i})$ for equation (10).

Note that the later definition of Young and Fox's definition are not satisfied in this instance. On account of the property A the optimum SOR parameter is

$$w_{opt} = \frac{2}{1 + [1 - \mu_1^2]^{1/2}} = \frac{2}{1 + \sin \frac{\pi}{s}} \tag{42}$$

and the radius of convergence of the iteration is

$$w_{opt} - 1 = \frac{1 - \sin \frac{\pi}{s}}{1 + \sin \frac{\pi}{s}} \tag{43}$$

The number of arithmetic operations per cycle of SOR is $2s^3(\mu + 4\alpha)$. We can deduce from (43) the number of iterations required for a given accuracy and hence the number of arithmetic operations. These we show in the Table for s = 10, 15, 20. Only the dominant terms are given. In all cases the iteration method is very much more economical.

s	10	15	20
Number of operations for direct method	$5x10^6(\mu+\alpha)$	$8.5x10^7(\mu+\alpha)$	$6.4x10^8(\mu+\alpha)$
Radius of convergence of SOR	.53	.66	.73
Number of iterations for 4 decimals	15	23	30
Number of iterations for 7 decimals	26	39	52
Number of operations for 4 decimals	$3x10^4(\mu+4\alpha)$	$1.6x10^5(\mu+4\alpha)$	$4.8x10^5(\mu+4\alpha)$
Number of operations for 7 decimals	$5.2x10^4(\mu+4\alpha)$	$2.6x10^5(\mu+4\alpha)$	$8.3x10^5(\mu+4\alpha)$

CHAPTER 4

BLOCK ITERATIVE METHODS

It often happens that the matrix A can be partitioned into blocks in a natural way, in the form

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & \\ \cdots & \cdots & \cdots & \cdots \\ A_{N1} & A_{N2} & & A_{NN} \end{bmatrix} \qquad (1)$$

In all the block iterative methods we treat the blocks in exactly the same way as we previously treated the elements. As each block is treated we will need to solve a set of equations of the form

$$A_{ii} x_i = y_i . \qquad (2)$$

If the method is to be used efficiently we should calculate the triangular decomposition of A_{ii} once and for all so that to solve (2) we need only perform a process of forward and backward substitution. If all the A_{ii} are full matrices or band matrices, full within the band, then the total amount of arithmetic needed per complete iteration is identical with that of point iteration. If the matrix A is stored explicitly, then the storage is identical, but if it is stored implicitly, that is in the form of a programme, then more storage is needed to keep the triangular decompositions of the diagonal submatrices.

The Jacobi, Von Mises, Gauss-Seidel, SOR and symmetric SOR iterations and the results that we have proved about them all generalize to block iteration in a perfectly straightforward way. We will not consider the details here except to mention how we generalize $D^{\frac{1}{2}}$. We replace D by the block diagonal matrix

$$D_b = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{nn} \end{bmatrix} . \qquad (3)$$

If the matrix A is symmetric, then so is D_b and hence there exists an orthogonal matrix O such that

$$D_b = O R O^T \qquad (4)$$

and R is diagonal. We now define

$$D_b^{\pm\frac{1}{2}} = O R^{\pm\frac{1}{2}} O^T . \qquad (5)$$

The generalization of Kaczmarz iteration is perhaps not quite so obvious. We now drop a perpendicular onto the meet of all the planes comprising one block. In equations (77) onwards of Chapter 2 b_i and λ become vectors and a_i becomes a rectangular matrix.

Block property A is defined exactly as point property A but with blocks replacing elements and in (10) L and U are block triangular. Some matrices possess block property A without possessing point property A. In such cases block SOR is preferable to point SOR.

Comparisons of Different Splittings

In a practical problem there are often several natural ways of choosing the blocks and we are led to ask which gives the best convergence. An answer to this problem for a special class of matrices is given by Varga (1962, Chapter 3). He first defines the splitting $A = M - N$ to be regular if M^{-1} and N are non-negative, that is have all their elements non-negative. He then proves that if

$$A = M_1 - N_1 = M_2 - N_2 \qquad (6)$$

are two regular splittings of the matrix A, if A^{-1} is positive

(has all elements positive) and if $(N_2 - N_1)$ and N_1 are non-negative and non-null then

$$1 > \rho(M_2^{-1} N_2) > \rho(M_1^{-1} N_1), \tag{7}$$

where ρ denotes spectral radius.

Two important classes of matrices for which A^{-1} is positive are

1. $A = (a_{ij})$ irreducibly* diagonally dominant by rows with $a_{ij} \leq 0$ for $i \neq j$ and $a_{ii} > 0$

and

2. $A = (a_{ij})$ symmetric and positive definite, irreducible* and with $a_{ij} \leq 0$ for $i \neq j$.

If A is irreducible, we should really be treating the problem in terms of its irreducible parts, but we may also argue that, by continuity over small perturbations of A, the results (7) are still valid if equality is allowed.

If A belongs to class 2 above and also possesses point property A, then the previous theory tells us that the more non-zero elements of A that are included in M the better the convergence of the Jacobi iteration. In particular block Jacobi will converge faster

---

* The matrix A is reducible if there exists a permutation matrix P such that

$$P A P^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}.$$

It is irreducible if no such matrix P exists.

than point Jacobi. Applying the property A theory we have that optimum block SOR converges faster than optimum point SOR. If no more work per iteration is required block iteration is therefore superior, but experience seems to indicate that rarely is the improvement sufficient to justify the use of block SOR when more work per iteration is required.

CHAPTER 5

GRADIENT OR SEMI-ITERATIVE METHODS

All the iteration methods considered so far are of the form

$$\underline{x}^{(i+1)} = T\,\underline{x}^{(i)} + \underline{g}\,. \tag{1}$$

In a semi-iterative method we consider a linear combination of the iterates

$$\underline{y}^{(i)} = \sum_{j=0}^{i} v_j(i)\,\underline{x}^{(j)}\,. \tag{2}$$

The coefficients $v_j(i)$ are chosen so that $\underline{y}^{(i)}$ is rapidly convergent, in some sense, to the solution of the problem. We require the condition

$$\sum_{j=0}^{i} v_j(i) = 1 \tag{3}$$

in order that $\underline{y}^{(r)}$ should be the solution $\underline{x}$, if $\underline{x}^{(0)} = \underline{x}$.

It is easily seen that if $\underline{e}^{(i)} = \underline{x} - \underline{x}^{(i)}$ and $\underline{f}^{(i)} = \underline{x} - \underline{y}^{(i)}$ are the error vectors corresponding to $\underline{x}^{(i)}$ and $\underline{y}^{(i)}$ then

$$\underline{f}^{(i)} = \sum_{j=0}^{i} v_j(i)\,T^j\,\underline{e}^{(0)}. \tag{4}$$

If we define the polynomial

$$P_i(\mu) = \sum_{j=0}^{i} v_j(i)\,\mu^j, \tag{5}$$

then

$$\underline{f}^{(i)} = P_i(T)\,\underline{e}^{(0)}, \tag{6}$$

and condition (3) can be written

$$P_i(1) = 1. \tag{7}$$

We will now show that the semi-iterative method (2) is only another way of writing the gradient method. Suppose

$$P_i(\mu) = \prod_{j=1}^{i} \left(\frac{\mu - q_j}{1 - q_j} \right), \tag{8}$$

then from (6) we have

$$\underline{f}^{(i)} = \prod_{j=1}^{i} \frac{T - q_j I}{1 - q_j} \underline{e}^{(0)}$$

$$= \frac{T - q_i I}{1 - q_i} \underline{r}^{(i-1)}$$

$$= \underline{f}^{(i-1)} + \frac{1}{1-q_i} \left[T \underline{f}^{(i-1)} - \underline{f}^{(i-1)} \right] \tag{9}$$

and hence

$$\underline{x}^{(i)} = \underline{x}^{(i-1)} + \frac{1}{1 - q_i} \left[T \underline{x}^{(i-1)} - \underline{x}^{(i-1)} + \underline{k} \right]. \tag{10}$$

Now we define a residual vector for the iterate $\underline{y}^{(i)}$ as

$$\underline{r}^{(i)} = T \underline{y}^{(i)} - \underline{y}^{(i)} + \underline{k}. \tag{11}$$

Note that in the case of the Jacobi iteration applied to a matrix A with units down its diagonal, this reduces to the usual definition

$$\underline{r}^{(i)} = \underline{b} - A \underline{x}^{(i)}. \tag{12}$$

Using the definition (11) the iteration may be written

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \frac{1}{1-q_{i+1}} \underline{r}^{(i)}. \tag{13}$$

Now suppose T is symmetric and consider the quadratic form

$$Q_e(\underline{y}) = (\underline{y} - C^{-1}\underline{k})^T C (\underline{y} - C^{-1}\underline{k}). \tag{14}$$

where $C = I - T$. If the iteration is convergent, $\rho(T) < 1$ and so C is positive definite. Thus $Q_e(\underline{y})$ is non-negative and takes its

minimum value of zero at the required solution, $\underline{y} = \underline{x} = C^{-1}\underline{g}$. Now

$$\text{grad } Q_o(\underline{y}^{(i)}) = 2(C\underline{y}^{(i)} - \underline{g}) = 2\underline{r}^{(i)} \tag{15}$$

and so $\underline{r}^{(i)}$ gives the direction from $\underline{y}^{(i)}$ in which $Q_o(\underline{y})$ diminishes the most rapidly, a fact leading to the name "gradient method".

The argument will be simplified if we consider the matrix C of (14) rather than T. This will bring our notation into line with that of Stiefel (1958) whose results we shall be quoting.

We define a new polynomial

$$R_i(\mu) = P_i(1 - \mu) \tag{16}$$

and (6) and (7) become

$$\underline{f}^{(i)} = R_i(C)\,\underline{e}^{(0)} \tag{17}$$

and

$$R_i(0) = 1. \tag{18}$$

Chebyshev Semi-iteration

Suppose that C has all its eigenvalues real and positive and that its Jordan canonical form is diagonal. Then there exists a complete set of eigenvectors, and the original error vector can be expanded in the form

$$\underline{e}^{(0)} = \sum_{j=1}^{n} \underline{C}_j\ , \tag{19}$$

where $\underline{C}_j$ is an eigenvector of C. Equation (17) gives

$$\underline{f}^{(i)} = \sum_{j=1}^{n} R_i(\lambda_j)\,\underline{C}_j\ , \tag{20}$$

where the λ_j are eigenvalues of C. Our problem is to minimize $|R_i(\lambda_j)|$ subject to $R_i(0) = 1$. Suppose that the eigenvalues lie in the range

$$0 < a \leq \lambda_j \leq b, \tag{21}$$

and that we replace the problem by that of minimizing the maximum value of $|R_i(\lambda)|$ over the range (a, b), still subject to the condition $R_i(0) = 1$. The well-known solution is given by the ith Chebyshev polynomial over (a, b),

$$R_i(\lambda) = \frac{T_i\left(\frac{-2\lambda + a + b}{b - a}\right)}{T_i\left(\frac{b + a}{b - a}\right)} . \tag{22}$$

Using the recurrence relation for the Chebyshev polynomials we obtain for the polynomials $R_i(\lambda)$ the recurrence relation

$$R_{i+1}(\lambda) = R_{i-1}(\lambda) + \frac{T_i(\gamma)}{T_{i+1}(\gamma)} \left\{ \frac{-4\lambda R_i(\lambda)}{b - a} + 2\gamma (R_i - R_{i-1}) \right\} , \tag{23}$$

where $\gamma = \frac{b + a}{b - a}$. This leads to the corresponding recurrence relation satisfied by $\underline{x}^{(i)}$, given by

$$\underline{x}^{(i+1)} = \underline{x}^{(i-1)} + \frac{T_i(\gamma)}{T_{i+1}(\gamma)} \left\{ \frac{4}{b-a} \underline{r}^{(i)} + 2\gamma (\underline{x}^{(i)} - \underline{x}^{(i-1)}) \right\} \tag{24}$$

where $\underline{r}^{(i)}$ is defined by

$$\underline{r}^{(i)} = \underline{g} - C \underline{x}^{(i)}. \tag{25}$$

Formulae (24) and (25) provide a convenient scheme for actual computation. Probably the computer programme will be designed to compute the vector

$$\underline{x}^{(i+1)} = T \underline{x}^{(i)} + \underline{g} , \tag{26}$$

in analogy with equation (1). From this $\underline{r}^{(i)}$ can be calculated conveniently as

$$\underline{r}^{(i)} = \underline{x}^{(i+1)} - \underline{y}^{(i)} \quad . \tag{27}$$

We shall also need a convenient method of calculating

$$\alpha_i = \frac{T_i(\gamma)}{T_{i+1}(\gamma)} \tag{28}$$

From the recurrence relation for the Chebyshev polynomials it follows that

$$\alpha_i = \frac{1}{2\gamma - \alpha_{i-1}} \tag{29}$$

This formula has the advantage of stability, for if an error ϵ is present in α_i, we will have a consequent error in α_{i+1} of amount

$$\frac{\epsilon}{(2\gamma - \alpha_i)^2} + O(\epsilon^2) \quad . \tag{30}$$

Now $\gamma > 1$ and $\alpha_i < 1$, so the result is established.

Provided there are no eigenvalues outside the range (a, b) it is easily seen that the convergence rate ultimately behaves as

$$R_i = \frac{1}{i} \log [T_i(\gamma)] \tag{31}$$

and in the limit

$$R_\infty = -\log (\gamma - \sqrt{\gamma^2 - 1}). \tag{32}$$

Chebyshev acceleration is very successful provided we can find good estimates for a and b. The lower bound, a, is often rather critical since the overall convergence is governed by the size of γ. We illustrate the improvement by considering a case that is very slowly convergent. Suppose that $\rho(T) = 1-\epsilon$ and that C is positive definite. Then $a = \epsilon$, $b = 1$, $\gamma = (1+\epsilon)/(1-\epsilon)$ and

$$R_\infty = -\log\left\{\frac{1+\varepsilon-\sqrt{4\varepsilon}}{1-\varepsilon}\right\}$$

$$= -\log\left[1-2\varepsilon^{\frac{1}{2}}+O(\varepsilon)\right] \qquad (33)$$

$$= 2\varepsilon^{\frac{1}{2}}+O(\varepsilon)\ .$$

For the unaccelerated iteration $R_\infty = \varepsilon$. We thus have a very considerable gain.

If there is an eigenvalue μ outside the estimated range (a, b) then the convergence will not be as good as possible, and the residuals will ultimately behave according to the equation

$$\underline{r}^{(i+1)} = \frac{\gamma-\sqrt{\gamma^2-1}}{\gamma_1-\sqrt{\gamma_1^2-1}}\ \underline{r}^{(i)}\ , \qquad (34)$$

where

$$\gamma_1 = \frac{-2\mu+a+b}{b-a}\ . \qquad (35)$$

Behaviour in accordance with (34) is easily recognised and tells us immediately that the range (a, b) does not include all the eigenvalues. If this pattern is clearly recognisable then we may estimate μ from $\|\underline{r}^{(i+1)}\| \ / \ \|\underline{r}^{(i)}\|$ using equations (34) and (35), or we may obtain a more accurate estimate from the Rayleigh quotient

$$\frac{\underline{r}^{(i)T}C\,\underline{r}^{(i)}}{\underline{r}^{(i)T}\underline{r}^{(i)}}\ , \qquad (36)$$

since $\underline{r}^{(i)}$ will be dominated by the eigenvector corresponding to μ. The iteration is now restarted with altered range (a, b) and with the most recent estimate of the solution vector. The process may be used to find estimates of a and b if no a priori estimates are available. We simply commence with values of a and b that are likely to be too large and too small respectively.

Two important examples of iterations to which this may be applied are SOR where A is positive definite and symmetric Kaczmarz

for a general matrix. It is also used to accelerate Jacobi iteration where A is positive definite. Here C = A and we should note that the accelerated iteration with properly chosen bounds a and b will converge even if Jacobi does not.

If we relax the condition that the eigenvalues of C be real and positive and require only that they be real, we may apply the acceleration to the iteration with C^2 as its matrix. This has the disadvantage that each iteration will require about twice the work, since we shall need two successive applications of our algorithm for multiplying a given vector by the matrix C. Also the bounds for the eigenvalues will be less favourable.

Comparison of Chebyshev Acceleration with SOR

Varga (1962) considers the special case of T symmetric, convergent and possessing property A. By a suitable reordering of the vector elements we have

$$T = \begin{bmatrix} 0 & F \\ F^T & 0 \end{bmatrix} . \tag{37}$$

If we partition the vectors $\underline{x}^{(i)}$ and $\underline{g}$ in the corresponding way, the SOR iteration will be given by

$$\begin{aligned} \underline{x}_1^{(i+1)} &= \underline{x}_1^{(i)} + w \left[F \underline{x}_2^{(i)} + \underline{g}_1 - \underline{x}_1^{(i)} \right] \\ \underline{x}_2^{(i+1)} &= \underline{x}_2^{(i)} + w \left[F^T \underline{x}_1^{(i)} + \underline{g}_2 - \underline{x}_2^{(i)} \right] . \end{aligned} \tag{38}$$

If the only information about the eigenvalues of T that we use is its spectral radius, ρ, then equation (24) simplifies to

$$\underline{y}^{(i+1)} = \underline{y}^{(i-1)} + w_i \left[T \underline{y}^{(i)} + \underline{g} - \underline{y}^{(i-1)} \right] , \tag{39}$$

where

$$w_1 = \frac{2T_1(1/\rho)}{\rho\, T_{i+1}(1/\rho)} \quad . \tag{40}$$

If the vectors $y^{(i)}$ are partitioned to correspond with (37), then equation (39) may be written as

$$\begin{aligned} x_1^{(i+1)} &= x_1^{(i-1)} + w_i \left[F\, x_2^{(i)} + g_1 - x_1^{(i-1)} \right] \\ x_2^{(i+1)} &= x_2^{(i+1)} + w_i \left[F^T x_1^{(i)} + g_2 - x_2^{(i-1)} \right] \quad . \end{aligned} \tag{41}$$

It is clear that the sequence $\{x_1^{(0)}, x_2^{(1)}, x_1^{(2)}, x_2^{(3)}, \ldots\}$ may be generated on its own, thus halving both computing time and storage. If we compare this sequence with the sequence $\{x_1^{(0)}, x_2^{(0)}, x_1^{(1)}, x_2^{(1)}, \ldots\}$ generated by (38), we see an exact correspondence except that the relaxation factor is held constant in (38). Also it is quite easy to show that

$$\lim_{i\to\infty} [w_i] = \frac{2}{1+\sqrt{1-\rho^2}} \quad , \tag{42}$$

so that the Chebyshev iteration degenerates into optimal SOR, and the asymptotic convergence rate is identical. However the well-known minimal property of the Chebyshev polynomials shows us that the average rate of convergence, as defined in chapter 2, is better for the Chebyshev iteration. The proof is somewhat complicated and is given in Golub and Varga (1961). It is clear from equations (38) and (41) that there is no difference between the two iterations as far as storage or convenience is concerned. It must also be mentioned that the result of Golub and Varga only applies to the ordering of (37), the so-called " γ ordering", and it is a common experience that other consistent orderings may give quite different average rates of convergence over a small number of iterations. However we have not seen a case where the Chebyshev method showed a poorer average convergence rate.

We may feel tempted to use Chebyshev acceleration for the Seidel iteration since, for the matrix (37), this has real roots in the range $0 \leq \lambda \leq \rho^2$. It has been shown by Tee (1963) that the Seidel iteration matrix has a complete set of eigenvectors in this case, so that Chebyshev acceleration is applicable. A little elementary algebra shows that this gives exactly the same asymptotic convergence rate as the iteration of Golub and Varga. However it suffers from the disadvantage of requiring storage for two vectors instead of one. Tee gives an example of a matrix that has property A and is consistently ordered, but is not of the form (37); he finds that it does not have a complete set of eigenvectors, so that the theoretical basis for Chebyshev acceleration is lacking. He therefore recommends Chebyshev acceleration of the Seidel process only when the matrix has σ_1 ordering, and gives examples showing that slow convergence is obtained when pagewise ordering is used. For the case where the matrix T does not possess property A, but is convergent with real eigenvalues, Varga (1962) considers the problem

$$\begin{aligned} \underline{x} &= T\underline{y} + \underline{f} \\ \underline{y} &= T\underline{x} + \underline{f} \quad , \end{aligned} \tag{43}$$

and applies SOR. The Jacobi matrix is

$$J = \begin{bmatrix} 0 & T \\ T & 0 \end{bmatrix}, \tag{44}$$

and

$$J^2 = \begin{bmatrix} T^2 & 0 \\ 0 & T^2 \end{bmatrix}. \tag{45}$$

It follows that J has real roots and $\rho(J) = \rho(T)$, and that the optimum relaxation factor is

$$w_b = \frac{2}{1 + \sqrt{1 - \rho(T)^2}} \qquad (46)$$

Both the vectors $\underline{x}$, $\underline{y}$ will converge to the required solution. At first sight it would appear that we have made the same gain in convergence rate as in the case where T has property A. However each iteration requires twice the amount of work, since the order of J is twice the order of T. It is still true that a very considerable improvement in convergence rate will have been obtained if $\rho(T)$ is near unity.

Varga now considers the case of T symmetric and shows that the Chebyshev acceleration of (1) where the eigenvalues of T are taken in the range $(-\rho(T), \rho(T))$, is similar to SOR applied to (43), except that the relaxation factor is not constant. The asymptotic convergence rate is identical since the Chebyshev factors converge to the optimum SOR factor. He shows that the average rate of convergence is better for the Chebyshev method. The case for using Chebyshev is thus quite strong. This is all the more true if we can place the eigenvalues of T in a range that is not symmetric about the origin. In fact if (I - T) is definite, the Chebyshev method yields a convergent process even if T is not convergent.

Chebyshev Acceleration of a General Iterative Process

The case where the matrix T has complex roots has been considered by Wrigley (1963). If the roots of C = I - T are λ_j, then we define constants γ_j as

$$\gamma_j = \frac{-2\lambda_j + a + b}{b - a} \quad . \qquad (47)$$

It follows from (22) that the component of the j^{th} eigenvector is reduced during the (i+1)th iteration by the factor

$$\mu_{i,j} = \frac{T_{i+1}(\gamma_j)}{T_{i+1}(\gamma)} \times \frac{T_i(\gamma)}{T_i(\gamma_j)} \quad . \tag{48}$$

As $i \to \infty$, $\mu_{i,j}$ converges to μ_j, given by

$$\mu_j = \frac{\gamma_j + \sqrt{\gamma_j^2 - 1}}{\gamma + \sqrt{\gamma^2 - 1}} \quad . \tag{49}$$

Writing $c = \gamma + \sqrt{\gamma^2 - 1}$, we have, after a little algebraic manipulation,

$$\gamma_j = \tfrac{1}{2}\left[c\mu_j + c\frac{1}{\mu_j} \right] . \tag{50}$$

It follows that if μ_j lies on the circle

$$|\mu_j| = k, \tag{51}$$

then λ_j lies on the ellipse

$$\frac{(2x-a-b)^2}{\left[\frac{1}{2}(b-a)(ck+\frac{1}{ck})\right]^2} + \frac{(2y)^2}{\left[\frac{1}{2}(b-a)(ck-\frac{1}{ck})\right]^2} = 1. \tag{52}$$

Thus the problem at hand is to find which ellipse of this form includes all the eigenvalues λ_j and has the minimum value of k. For fixed a, b the system (52) consists of confocal ellipses. Wrigley's suggestion is that the iteration be performed with trial values of a and b and continued until the vector of changes is dominated by the contribution of the eigenvector corresponding to the dominant root, if this is real, or the contributions of the two complex conjugate eigenvectors if the dominant root is complex. In either case the dominant root may be estimated and the contribution of the corresponding eigenvector or eigenvectors eliminated by using a suitable linear combination of the iterates. The same process is applied to find the subdominant root, and values of a, b are now

chosen so that the ellipse given by equation (52) passes through both of the known roots. He makes light, however, of the difficulties in finding reliable values for the roots of the iteration matrix, particularly should they be both complex. In the example he quotes, the subdominant root is real and is the limit of the sequence

$$.937,\ .870,\ .907,\ .919,\ .828\ \ldots \tag{53}$$

He applies δ^2 extrapolation to the middle three numbers and obtains .921. These numbers are not nearly well enough behaved to use extrapolation and in fact if δ^2 extrapolation is applied to the middle three the result is .925.

A further disadvantage of Wrigley's technique is that we have no guarantee that the process will converge, for we cannot be sure that his ellipse will enclose all the eigenvalues. The process should therefore be used with great caution. We recommend it only where it is known that the imaginary parts of all the eigenvalues are small. In such cases the chance of divergence is much reduced and the ellipse will have greater eccentricity so that the improvement in convergence compared with the unaccelerated method will be more marked.

Method of Conjugate Gradients for a Positive-definite Matrix

Suppose that the matrix C is symmetric and positive definite and that we choose $R_i(\lambda)$ so as to minimise the quadratic form

$$Q_\mu = (\underline{r}^{(i)},\ C^{\mu-1}\ \underline{r}^{(i)}), \tag{54}$$

where μ is a non-negative integer.

If

$$R_i(\lambda) = 1 + \sum_{j=1}^{i} \rho_j\ \lambda^j\ , \tag{55}$$

then

$$Q_\mu = (\underline{r}^{(0)} + \sum_{j=1}^{i} \rho_j C^j \underline{r}^{(0)},\ C^{\mu-1} \underline{r}_0 + \sum_{j=1}^{i} \rho_j C^{j+\mu-1} \underline{r}^{(0)}), \tag{56}$$

and

$$\frac{\partial Q_\mu}{\partial \rho_j} = 2(\underline{r}^{(i)},\ C^\mu C^{j-1} \underline{r}^{(0)}). \tag{57}$$

At a minimum of Q this will be zero for all j. Hence $\underline{r}^{(i)}$ will be orthogonal to $\underline{r}^{(j)}$ for $j < i$, with respect to the weight function C^μ. This is all we need to build a recurrence relation. Suppose

$$\underline{r}^{(i+1)} = \underline{r}^{(i)} + \frac{1}{q_i}(-C \underline{r}^{(i)} + \beta_i (\underline{r}^{(i)} - \underline{r}^{(i-1)}), \tag{58}$$

then $\underline{r}^{(i+1)}$ is orthogonal to $\underline{r}^{(i)}$ and $\underline{r}^{(i-1)}$ with respect to C^μ if

$$q_i = \frac{(C \underline{r}^{(i)},\ C^\mu \underline{r}^{(i)})}{(\underline{r}^{(i)},\ C^\mu \underline{r}^{(i)})} - \beta_i, \tag{59}$$

where

$$\beta_i = \frac{-(C \underline{r}^{(i)},\ C^\mu \underline{r}^{(i-1)})}{(\underline{r}^{(i-1)},\ C^\mu \underline{r}^{(i-1)})}, \qquad \text{for } i > 0 \text{ and } \beta_0 = 0. \tag{60}$$

It is easily seen by induction that we have achieved our aim. For suppose $\{\underline{r}^{(0)}, \underline{r}^{(1)}, \ldots \underline{r}^{(i)}\}$ form an orthogonal set with respect to C^μ, then for $j \leq i-2$,

$$\begin{aligned}
(\underline{r}^{(i+1)},\ C^\mu \underline{r}^{(j)}) &= -\frac{1}{q_i}(C \underline{r}^{(i)},\ C^\mu C \underline{r}^{(j)}) \\
&= -\frac{1}{q_i}(\underline{r}^{(i)},\ C^\mu \underline{r}^{(j)}) \\
&= -\frac{1}{q_i}(\underline{r}^{(i)},\ C^\mu \sum_{k=0}^{j+1} t_k \underline{r}^{(k)}) \\
&= 0.
\end{aligned} \tag{61}$$

The iteration itself takes the form

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \frac{1}{q_i} [\underline{r}^{(i)} + \beta_i (\underline{x}^{(i)} - \underline{x}^{(i-1)}] . \tag{62}$$

Sometimes β_i is calculated from the relation

$$\beta_i = \frac{(\underline{r}^{(i)}, C^\mu \underline{r}^{(i)})}{(\underline{r}^{(i-1)}, C^\mu \underline{r}^{(i-1)})} \quad q_{i-1} . \tag{63}$$

This is found from (60) using (58) with i replaced by (i-1), to eliminate $(C \underline{r}^{(i)}, C^\mu \underline{r}^{(i-1)})$. This formula (63) is to be preferred since one less inner product need be calculated. For practical computation we are concerned only with the cases $\mu = 0, 1$. These require the computation of only $C \underline{r}^{(i)}$ to obtain β_i and q_i, whereas for higher μ we need $C^2 \underline{r}^{(i)}$, etc. We can use $C \underline{r}^{(i)}$ to compute recursively the next residual with the help of (58).

It is important to note that since the set $\{\underline{f}^{(i)}\}$ is orthogonal with respect to the weight function $C^{\mu+2}$, there cannot be more than m vectors $\underline{f}^{(i)}$ if m is the dimension of the vector space spanned by the vectors $C^j \underline{e}^{(0)}$. Of course $m \leq n$ where n is the order of C. Thus after m steps the iteration should be complete. Unfortunately round-off errors destroy this property. The case where C is symmetric but non-definite should be mentioned. If we take $\mu = 1$ the quadratic form $Q_1 = (\underline{r}^k, \underline{g}^k)$ is still positive-definite. The minimum value of Q_1 corresponds to the true answer and in theory we shall at each stage have chosen that gradient iteration which produces the smallest value of Q_1. It should theoretically converge after m steps, where m is the dimension of the vector space spanned by $C^i \underline{r}^{(0)}$. The case $\mu = 0$ should also show convergence after m steps, but there is no corresponding minimum property. There is in theory a possibility of failure. Suppose we have chosen $\underline{r}^{(0)}$ such that

$$(\underline{r}^{(i)},\ C\,\underline{r}^{(i)}) = 0 \quad \text{and } i < m, \tag{64}$$

then from (62) we will have $\underline{y}^{(i+1)} = \underline{y}^{(i)}$ and the iteration will make no further progress. This situation can be remedied by performing two iterations concurrently. We replace (58) by the equation

$$\underline{r}^{(i+1)} = \underline{r}^{(i)} - \alpha_i C\,\underline{r}^{(i)} - \beta_i C^2\,\underline{r}^{(i)} + \gamma_i(\underline{r}^{(i)} - \underline{r}^{(i-1)}) + \delta_i C(\underline{r}^{(i)} - \underline{r}^{(i-1)}) \tag{65}$$

and the iteration (62) by

$$\underline{y}^{(i+1)} = \underline{y}^{(i)} + \alpha_i \underline{r}^{(i)} + \beta_i C\,\underline{r}^{(i)} + \gamma_i(\underline{y}^{(i)} - \underline{y}^{(i-1)}) + \delta_i C(\underline{y}^{(i)} - \underline{y}^{(i-1)}). \tag{66}$$

The coefficients are so chosen that

$$(\underline{r}^{(i)},\ C\,\underline{r}^{(j)}) = (\underline{r}^{(i)},\ C^2\,\underline{r}^{(j)}) = 0 \text{ for } i \neq j. \tag{67}$$

This involves solving the set of equations

$$\alpha_i(\underline{r}^{(i)}, C^2\underline{r}^{(i)}) + \beta_i(\underline{r}^{(i)}, C^3\underline{r}^{(i)}) - \gamma_i(\underline{r}^{(i)}, C\underline{r}^{(i)}) - \delta_i(\underline{r}^{(i)}, C^2\underline{r}^{(i)}) = (\underline{r}^{(i)}, C\,\underline{r}^{(i)})$$

$$\alpha_i(\underline{r}^{(i)}, C^3\underline{r}^{(i)}) + \beta_i(\underline{r}^{(i)}, C^4\underline{r}^{(i)}) - \gamma_i(\underline{r}^{(i)}, C^2\underline{r}^{(i)}) - \delta_i(\underline{r}^{(i)} C^3\underline{r}^{(i)}) = (\underline{r}^{(i)}, C^2\underline{r}^{(i)})$$

$$\beta_i(\underline{r}^{(i-1)}, C^3\underline{r}^{(i)}) + \gamma_i(\underline{r}^{(i-1)}, C\underline{r}^{(i-1)}) + \delta_i(\underline{r}^{(i-1)} C^2\underline{r}^{(i-1)}) = 0$$

$$\alpha_i(\underline{r}^{(i-1)}, C^3\underline{r}^{(i)}) + \beta_i(\underline{r}^{(i-1)}, C^4\underline{r}^{(i)}) + \gamma_i(\underline{r}^{(i-1)}, C^2\underline{r}^{(i-1)}) + \delta_i(\underline{r}^{(i-1)} C^3\underline{r}^{(i-1)}) - \delta_i(\underline{r}^{(i-1)} C^3\underline{r}^{(i)}) = 0. \tag{68}$$

The matrix C^2 is positive-definite and so $(\underline{r}^{(i)},\ C^2\,\underline{r}^{(i)})$ is non-zero unless $\underline{r}^{(i)}$ is zero. Hence we cannot have the solution $\alpha_i = \beta_i = \gamma_i = \delta_i = 0$ unless $\underline{r}^{(i)}$ is zero, and $\underline{y}^{(i+1)} = \underline{y}^{(i)}$ only if

$\underline{r}^{(i)} = 0$. As might be expected, this iteration requires about twice the amount of work, since we must compute $C\,\underline{r}^{(i)}$ and $C^2\underline{r}^{(i)}$. The storage is increased from three vectors to five.

We have no experience of any conjugate gradient iteration with a non-definite symmetric matrix. Using the double iteration and $\mu = 1$ seems the more attractive on account of the minimum property.

Method of Conjugate Gradients for a General Matrix

We can extend the method to deal with a general matrix. Unfortunately we need to carry out an iteration on the matrix equation

$$C^T X = E \tag{69}$$

in parallel with the iteration we require. The error vector for (69) satisfies the equation

$$\underline{e}^{(i)} = R_i\,(C^T)\underline{e}^{(0)} \tag{70}$$

where $\underline{e}^{(0)}$ is the error in the first iterate, and the residual vector is

$$\underline{g}^{(i)} = C^T\,\underline{e}^{(i)}. \tag{71}$$

The analysis is exactly parallel with that given for the symmetric case. In the first part of an inner product we replace C, $\underline{e}^{(0)}$, $\underline{r}^{(i)}$, $\underline{r}^{(i)}$ by C^T, $\underline{g}^{(0)}$, $\underline{q}^{(i)}$, $\underline{s}^{(i)}$ respectively. The iterates for equation (69) are given by

$$\underline{z}^{(i+1)} = \underline{z}^{(i)} + \frac{1}{q_i}\,(\underline{s}^{(i)} + \beta_i\,(\underline{z}^{(i)} - \underline{z}^{(i-1)}) \quad . \tag{72}$$

There appears to be one serious drawback to this technique. Suppose that $(\underline{s}^{(i)}, C^\mu\,\underline{r}^{(i)})$ is zero before we have reached the point of theoretical convergence. The iteration will then break down. This may happen for several reasons. If the vector space spanned by $C^i\underline{e}^{(0)}$ has a greater dimension than that of the space spanned by

$C^{T^i}\ \underline{g}^{(0)}$, then $\underline{s}^{(i)}$ may be null without $\underline{r}^{(i)}$ being null. The vectors $\underline{s}^{(i)}$ and $C^{\mu}\ \underline{r}^{(i)}$ may be orthogonal without either being zero. Round-off errors will prevent either of these situations occurring in practice, but the effect of their "nearly" happening is not clear.

Numerical Stability

It is a well-known experience that the Chebyshev semi-iterative method is remarkably stable in the sense that provided the bounds a, b have been properly chosen the convergence rate shows good agreement with the theory. This is in marked contrast with the conjugate gradient method. If C is positive definite and the dimension of the space $C^j\ \underline{g}^{(0)}$ is m, the error is often found to be quite large after m steps; indeed if the algorithm is continued the error often drops quite markedly after a few more iterations. Engeli et al (1959) give examples of these phenomena.

We will not attempt a detailed analysis here, but will indicate why the Chebyshev method is more stable. In the range (a, b) the Chebyshev polynomial oscillates between its two extreme values, $\pm \frac{1}{T_i(\gamma)}$. We would expect that the polynomial that is actually taken in the computation would not differ markedly from this and that its modulus would nowhere greatly exceed $\frac{1}{T_i(\gamma)}$. Now it often happens that a large number of the eigenvalues, λ_j, of C, are clustered near the lower end of the range (a, b). The residual polynomial is

$$R_i(\mu) = \left(1 - \frac{\mu}{\lambda_1}\right)\left(1 - \frac{\mu}{\lambda_2}\right) \cdots \left(1 - \frac{\mu}{\lambda_n}\right), \qquad (73)$$

so that it is clear that this clustering will result in wild oscillations at the higher end of the range. Since rounding errors will result in the zeros of $R_i(\mu)$ being slightly different from the eigenvalues, it is possible that large errors may remain in the

components of the eigenvectors corresponding to the larger λ_j.

Methods of Steepest Descent

In a method of steepest descent the iteration takes the form

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - t_k \underline{g}^{(k)}, \tag{74}$$

where $\underline{g}^{(k)}$ is the gradient of the quadratic form Q_μ defined in equation (54) and t_k is a scalar chosen so as to minimise $Q_\mu(\underline{x}^{(k+1)})$ in the direction $\underline{g}^{(k)}$. The choice of $\underline{g}^{(k)}$ as the gradient of Q_μ ensures that at each step the change is in the direction of steepest descent of Q_μ.

As with conjugate gradients, only the cases $\mu = 0, 1$ are of interest to us, since higher values of μ involve more matrix by vector multiplications. If $\mu = 0$ it is easily verified that $\underline{g}^{(k)} = \underline{r}^{(k)}$ and hence the iteration is the same as the first step of the conjugate gradient method with $\mu = 0$. It is hardly surprising that Engeli et al (1959) report very slow convergence with this method.

If $\mu = 1$ then $\underline{g}^{(k)} = C^T \underline{r}^{(k)}$ and the iteration is equivalent to the first step of the conjugate gradient method with $\mu = 0$ applied to the system

$$C^T C \underline{x} = C^T \underline{g} . \tag{75}$$

We again recommend the use of conjugate gradients rather than steepest descent, since this minimizes Q_1 over all steps so far performed, rather than just over one step. Note that the matrix $C^T C$ need not be calculated explicitly. The only requirements are routines for multiplying an arbitrary vector by C and C^T. This method is not suitable for a positive-definite matrix, since in this case the ratio of largest to smallest eigenvalues of $C^T C$ will be the square of the corresponding ratio for C, and the troubles caused by rounding errors

will be magnified. For a general matrix the condition of the system (75) will again be much worse than the original system, but we now have a practical method with none of the previous possibilities of failure.

Khabaza (1963) has suggested a natural extension of the method of steepest descent. He considers the iteration

$$\underline{y}^{(k+1)} = \underline{y}^{(k)} - \sum_{j=0}^{m-1} t_j C^j \underline{r}^{(k)}, \tag{76}$$

where the coefficients are chosen to minimise the function

$$Q_1 = (\underline{r}^{(k)}, \underline{r}^{(k)}). \tag{77}$$

Finding the coefficients t_j involves solving an $m \times m$ set of equations. This iteration for positive-definite C is exactly equivalent to m steps of the conjugate gradient method with $\mu = 1$, and is less convenient to apply. Similarly for C symmetric but non-definite there is a correspondence if m is even. However, in the case of a general matrix we cannot generate the coefficients t_j recursively since we needed the symmetry of C in (61). It is a pity that Khabaza does not mention any experiments with non-symmetric matrices. He reports very good results in some test cases with symmetric matrices. This suggests that when using the conjugate gradient method we should not continue the algorithm until convergence is obtained, but rather restart the iteration after a moderate number of steps. Perhaps a suitable test would be to see how nearly orthogonal the current residual is to the original residual.

Combined Methods

We return to consider further improvements in the convergence where the eigenvalues of C are real and positive. We saw that the convergence of the Chebyshev accelerated iteration was critically dependent on the lower bound of the range (a, b). Suppose now that we choose a so that a few of the smaller eigenvalues lie outside

the range. The components of the eigenvectors corresponding to eigenvalues within the range will be reduced much more rapidly than before, and we should be able to eliminate the remaining error by a very small number of conjugate gradient steps. Engeli et al (1959) experimented with this idea. They applied the conjugate gradient method to an initial residual that practically spanned the space of the first nine eigenvectors. They report that after nine steps of the conjugate gradient method the residual was still quite large. This is because the components of the last few eigenvectors will be much increased during the conjugate gradient steps. We will consider in more detail the matrix that they were using. An upper bound for the eigenvalues is b = 64 and the first three eigenvalues are

$$\lambda_1 = .042$$

$$\lambda_2 = .057$$

$$\text{and} \quad \lambda_3 = .188 .$$

Possible choices for the lower bound of the range for Chebyshev acceleration are any of these three. The analysis of page 55 shows that the asymptotic convergence rates are approximately in the ratio

$$(42)^{\frac{1}{2}} : (57)^{\frac{1}{2}} : (188)^{\frac{1}{2}}$$

$$\text{that is} \qquad 6.5 : 7.6 : 13.7 .$$

If we take $a = \lambda_2$, then the conjugate gradient residual polynomial after just one step should be

$$R_1(\lambda) = \left(1 - \frac{\lambda}{.042}\right)$$

$$\text{and} \quad R_1(42) = -999.$$

If there is an eigenvalue greater than 42, which is likely since b = 64, then the component of the corresponding eigenvector will therefore be increased by a factor of at least 999. This can hardly be compensated for in the increased convergence rate of the

Chebyshev iteration. If we take $a = \lambda_3$, then we have

$$R_2(\lambda) = \left(1 - \frac{\lambda}{.042}\right)\left(1 - \frac{\lambda}{.188}\right).$$

and $R_2(42) = 2.2 \times 10^5$.

For an upper bound to the growth we must look at $R_2(64)$ which is 5.2×10^5. We do now have a much improved convergence rate, but we are paying dearly for it.

This example illustrates well the difficulties of this technique for an ill-conditioned problem. It is clear that it should only be used when the smaller eigenvalues are very well separated. For example, if Engeli's matrix had not possessed the eigenvalue $\lambda_2 = .06$, there would have been a strong case for using $(a, b) = (.18, 64)$ and keeping three additional decimals in the Chebyshev iteration.

A process that can be used with great success if the two smallest roots of the matrix C are well separated is Chebyshev acceleration over the range $[\lambda_2, \lambda_n]$, with Aitken's δ^2 method to eliminate the component of the dominant eigenvector of the iteration matrix. The components of the other eigenvectors will be changed by the δ^2 process so guarding figures must be kept or some additional Chebyshev steps performed afterwards.

Rutishauser (1959) proposed a modification of this technique for use where the eigenvalues outside the range (a, b) are known accurately. Suppose λ_i is one such eigenvalue and that instead of using the conjugate gradient iteration we use a Chebyshev iteration, so chosen that the residual polynomial has a zero at λ_i. We can do this by p Chebyshev steps over the range (a^*, b) where

$$a^* = \lambda_i - (b - \lambda_i) \tan^2\left(\frac{\pi}{4p}\right) . \qquad (78)$$

If p is chosen so large that $a^* > 0$, then the residual polynomials will not exceed unity in modulus anywhere in the range (0, b). We

have then completely avoided the instability of the conjugate gradient method. The procedure is applied in turn to all the eigenvalues outside (a, b). Engeli et al report favourably on this technique. Again, its usefulness will depend on how well the smallest eigenvalues are separated.

Rutishauser (1959) also discusses the idea of a spectral transformation. He replaces the equation

$$C \underline{x} = \underline{g} \tag{79}$$

by

$$C^* \underline{x} = \underline{g}^* , \tag{80}$$

where

$$C^* = I - R_m(C), \tag{81}$$

and

$$\underline{g}^* = C^{-1}[I - R_m(C)]\underline{g} , \tag{82}$$

where $R_m(\lambda)$ is a polynomial of degree m.

The eigenvalues of C^* are related to the eigenvalues of C by the equation

$$\lambda_i^* = 1 - R_m(\lambda_i). \tag{83}$$

He takes as $R_m(\lambda)$ the residual polynomial of m steps of a gradient method and calls this the inner iteration. He solves the set (80) by another gradient method, which he calls the outer iteration. If this outer iteration is started with the vector $\underline{y}^*$, then we will need the corresponding residual

$$\underline{r}^* = \underline{g}^* - C^* \underline{y}^* . \tag{84}$$

This is found by performing m inner iterations on the set

$C\underline{x} = \underline{g}$ with initial approximant $\underline{y}^*$ Now

$$\underline{r}^* = \underline{y}^* - \underline{y}^{(m)}, \tag{85}$$

for

$$\underline{r}^{(m)} = R_m(C)\ \underline{r}^{(0)},$$

so that

$$\underline{r}^{(0)} - \underline{r}^{(m)} = [I - R_m(C)]\ \underline{r}^{(0)}$$

and

$$\underline{y}^* - \underline{y}^{(m)} = C^{-1}[I - R_m(C)]\ [\underline{g} - C\ \underline{y}^*] \tag{86}$$

$$= \underline{g}^*\ C^*\ \underline{y}^* .$$

Also, in the course of the outer iteration we will need to find $C^*\ \underline{r}^{*(i)}$, given $\underline{r}^{*(i)}$. This is found by performing m iterations of the inner method applied to $A\underline{x} = \underline{r}^{*(i)}$, with starting vector $\underline{x}^{(0)} = \underline{0}$. This gives us

$$C^*\ \underline{r}^{*(i)} = \underline{r}^{(0)} - \underline{r}^{(m)}, \tag{87}$$

since

$$\underline{r}^{(0)} = \underline{r}^{*(i)},\ \underline{r}^{(m)} = R_m(C)\ \underline{r}^{(0)}. \tag{88}$$

The solution to the problem is therefore found by a gradient method applied to (80), with each step involving m multiplications by C. It is important to ensure that very few steps of the outer met od are required. Ginsburg (1959) proposed that the outer method should be the conjugate gradient method and that one of two devices should be used to ensure that few outer iterations are needed. The first uses Chebyshev iteration over (a, b) where this range is chosen to include all but a few of the smaller eigenvalues of C. This means that the residual of the outer method will be composed of only this number of eigenvectors. If the inner method is also properly chosen, then the condition of the matrix C* will be much better than the condition of C, and hence we will not have the kind of trouble noted in the method of page 70. Ginsburg's other suggestion is

to choose the inner method so that most of the eigenvalues of C^* are clustered about one point. Then the contributions from the eigenvectors corresponding to these will be nearly eliminated in one conjugate gradient step.

The most suitable method for the inner iteration would seem to be Chebyshev iteration. We cannot use conjugate gradients since this is dependent on the initial residual. Now we should not use Chebyshev iteration for the outer method, since the combination of inner and outer iterations is itself a gradient method and we will clearly obtain the best convergence in the Chebyshev sense with an ordinary Chebyshev iteration. We should get a better result with conjugate gradient method as outer iteration since this iteration minimises $(\underline{r}^{*(i)}, C^{*\mu-1}\,\underline{r}^{*(i)})$. Of course we should in theory get a better result with the ordinary conjugate gradient iteration, but we have already seen that rounding errors prevent this. Engeli et al (1959) report favourably on this combination of Chebyshev and conjugate gradients. They ensured the good conditioning of the matrix C^* by using a reasonably large number of inner Chebyshev iterations, actually in the range five to twenty.

CHAPTER 6

SOLUTION OF THE EIGENVALUE PROBLEM USING LANCZOS' METHOD

We consider here the problem of finding one or more solutions of the eigenvalue equation

$$A \underline{x} = \lambda \underline{x} , \tag{1}$$

where A is a sparse matrix. The transformation methods of Givens, Householder and Jacobi are unsuitable since they require the storage of all, or nearly all, the matrix elements. Lanczos' method of minimized iterations (Lanczos, 1950) does not suffer from these disadvantages and will be considered here.

The Symmetric Case

If A is symmetric we build up a sequence of vectors, $\underline{y}^{(i)}$, given by $\underline{y}^{(0)} = 0$, $\underline{y}^{(1)}$ arbitrary and

$$\gamma_{i+1} \underline{y}^{(i+1)} = A \underline{y}^{(i)} - \alpha_i \underline{y}^{(i)} - \beta_{i-1} \underline{y}^{(i-1)}. \tag{2}$$

The constants α_i, $\beta_{i-1}, \gamma_{i+1}$ are chosen so that $\underline{y}^{(i+1)}$ is normalised and orthogonal to both $\underline{y}^{(i)}$ and $\underline{y}^{(i-1)}$. The constants are therefore given by the equations

$$\alpha_i = \underline{y}^{(i)T} A \underline{y}^{(i)} , \tag{3}$$

$$\beta_{i-1} = \underline{y}^{(i-1)T} A \underline{y}^{(i)} , \tag{4}$$

and γ_{i+1} is chosen so that $\underline{y}^{(i+1)T} \underline{y}^{(i+1)} = 1$.

This choice of the constants α_i, β_{i-1}, γ_{i+1} implies that the sequence $\underline{y}^{(1)}$, $\underline{y}^{(2)}$ ... is orthonormal. This follows from a simple induction argument applied to equation (2). It follows that if the dimension of the space spanned by the set of vectors $\{A^i \underline{y}^{(1)}\}$ is k, then there will not be more than k non-zero vectors $\underline{y}^{(i)}$, and $\underline{y}^{(k+1)}$ will be zero. If Y is the matrix whose columns are $\underline{y}^{(i)}$ for

$i = 1, 2 \ldots k$, then equation (2) can be written in the matrix form

$$AY_k = Y_k C_k , \tag{5}$$

where

$$C_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_2 & \alpha_2 & \beta_2 & & \\ & \gamma_3 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \gamma_k & \alpha_k \end{bmatrix} \tag{6}$$

The tridiagonal matrix C_k is, furthermore, symmetric since

$$\begin{aligned} \beta_{i-1} &= \underline{y}^{(i-1)T} A \underline{y}^{(i)} \\ &= \underline{y}^{(i)T} A \underline{y}^{(i-1)} \\ &= \underline{y}^{(i)T} \left[\alpha_{i-1} \underline{y}^{(i-1)} + \beta_{i-2} \underline{y}^{(i-2)} + \gamma_i \underline{y}^{(i)} \right] \\ &= \gamma_i . \end{aligned} \tag{7}$$

Methods for finding the roots and vectors of a symmetric tridiagonal matrix are well-established. We use the method of bisections, possibly with a Newton iteration to accelerate the convergence. For details we refer the reader to Fox (1964), for example.

Unfortunately the process will not terminate owing to the occurrence of rounding errors, except in some very special cases where it is possible to use rational numbers throughout. With a matrix of quite moderate size it may be quite impossible to recognise the point at which the process should have terminated. It has been suggested that this difficulty be overcome by reorthogonalisation, that is subtracting out from each vector $\underline{y}^{(i+1)}$ the components of

$\underline{y}^{(1)}, \underline{y}^{(2)} \ldots \underline{y}^{(i-2)}$. For the problems we have in mind, however, this would involve a prohibitive amount of work ~~since the number of arithmetic operations would be $n^3 + O(n^2)$~~ and time would probably also be wasted in transfers to and from an auxiliary store.

Accurate results, however, may be obtained without any reorthogonalisation. Some examples are given by Ginsburg (1958). He calls it the method of conjugate gradients and also considers the case of orthogonalisation with respect to A, so that $\underline{y}^{(i)T} A \underline{y}^{(j)} = 0$ for $i \neq j$. As in Chapter 5 for the solution of linear equations, there is no difference in principle between the two versions and it seems unlikely that either would give significantly better results than the other. Ginsburg stops the iteration after k steps, where k is no longer necessarily equal to the dimension of the space $\left\{ A^i \underline{y}^{(1)} \right\}$, and finds the eigenvalues of the matrix C_k. Some of these are likely to be good approximations to the eigenvalues of A, but he does not give a reliable procedure for determining which. In fact we can show that an eigenvalue λ of C_k will be a good approximation to an eigenvalue of A if the last element in the corresponding vector is small. If the eigenvector is $\underline{x} = (x_1 \ldots x_k)^T$ then since

$$AY_k = Y_k C_k + (\underline{0}, \underline{0}, \ldots \underline{0}, \gamma_{k+1} \underline{y}^{(k+1)}) \tag{8}$$

it follows that

$$\begin{aligned} AY_k \underline{x} &= Y_k C_k \underline{x} + \gamma_{k+1} x_k \underline{y}^{(k+1)} \\ &= \lambda Y_k \underline{x} + \beta_k x_k \underline{y}^{(k+1)} \end{aligned} \tag{9}$$

and $Y_k \underline{x}$ is a good approximation to an eigenvector of A if $\| \beta_k x_k \underline{y}^{(k+1)} \| = |\beta_k x_k|$ is small and the error in the eigenvalue λ is bounded by $|\beta_k x_k|$. If a really accurate eigenvalue is required we suggest the use of the Rayleigh quotient with the result of Wilkinson (1961) to give an error bound. If the approximate eigenfunction has Euclidean norm unity and the norm of the residual

is ε then the error in the eigenvalue is not greater than $\frac{\varepsilon^2}{a}/(1 - \varepsilon^2/a^2)$ where a is the distance to the nearest distinct eigenvalue of A.

If k is taken to be larger than the dimension of the space $\left\{A^i \underline{y}^{(1)}\right\}$, which will in general be necessary, then it is likely that the matrix C_k will represent some of the eigenvalues of A more than once. It may happen that none of the eigenvectors of C_k corresponding to a single eigenvalue of A has a small last element, but we may find a linear combination of two eigenvectors which is an approximate eigenvector of C_k and which has a small last element.

There seems no reason to suppose that we should not obtain all the eigenvalues of A in this way provided k is taken sufficiently large. We will, however, obtain only one eigenvector for each degenerate eigenvalue. To obtain the others we restart the iteration with a vector which is orthogonal to all the eigenvectors that have been found so far.

An Example

We illustrate with the matrix

$$A = \begin{bmatrix} P & I & & & & & \\ I & P & I & & & & \\ & I & P & I & & & \\ & & I & P & I & & \\ & & & I & P & I & \\ & & & & I & P & I \\ & & & & & I & P \end{bmatrix} \qquad (10)$$

where I is the unit matrix of order 7 and

$$P = \begin{bmatrix} -4 & 1 & & & & & & \\ 1 & -4 & 1 & & & & & \\ & 1 & -4 & 1 & & & & \\ & & 1 & -4 & 1 & & & \\ & & & 1 & -4 & 1 & & \\ & & & & 1 & -4 & 1 & \\ & & & & & 1 & -4 & \end{bmatrix} . \qquad (11)$$

For starting vector we take $\underline{y}^{(1)} = (1, 0, 0, \ldots 0)^T$ and use the arithmetic of the Mercury computer, which works with 29 significant binary figures. We show in Table 1 the values of α_k and β_k obtained for k=1, 2, ... 33.

k	α_k	β_k
1	-4.000000	1.414214
2	-4.000000	1.732051
3	-4.000000	1.825742
4	-4.000000	1.888562
5	-4.000000	1.915749
6	-4.000000	1.938857
7	-4.000000	1.949789
8	-4.000000	1.949885
9	-4.000000	1.894084
10	-4.000000	1.777367
11	-4.000000	1.705817
12	-4.000000	1.641283
13	-4.000000	0.889256
14	-4.000000	2.004822
15	-4.000000	1.208326
16	-4.000000	1.547263
17	-4.000000	1.052819
18	-4.000000	1.148055
19	-4.000000	0.752624
20	-4.000000	0.972239
21	-4.000000	0.718450
22	-3.999986	1.015782
23	-3.999844	0.432507
24	-3.988897	0.727068
25	-3.735736	1.889646
26	-3.544555	3.085471
27	-4.730320	0.059328
28	-3.743282	3.252928
29	-4.257260	0.013666
30	-5.371088	2.445771
31	-2.702467	0.170410
32	-4.275053	2.315253
33	-2.986031	0.466423

TABLE 1

In this case the dimension of the space $\{A^i \underline{y}(1)\}$ is 25, but β_{25} is not small and it is impossible to recognise this dimension by inspection of the β_k. We show in Table 2 the eigenvalues of C_k and the last elements of the corresponding normalised vectors for k = 33, 30, 27, 24.

k = 33		k = 30		k = 27		k = 24	
-0.304482	1, -7	-0.304482	7, -7	-0.304482	4, -6	-0.304482	8, -8
-0.304483	1, -5	-0.304483	9, -5	-0.3057	5, -1		
-0.738028	4, -5	-0.738028	2, -6	-0.738028	8, -7	-0.738028	2, -6
-0.738029	1, -3	-0.738065	2, -3				
-1.148080	6, -1						
-1.171573	8, -7	-1.171573	1, -8	-1.171573	1, -5	-1.171573	5, -5
-1.326586	5, -1						
-1.386875	1, -7	-1.386875	1, -8	-1.386875	5, -5	-1.386875	2, -4
-1.820420	2, -8	-1.820420	3, -8	-1.820420	2, -4	-1.820420	1, -3
-2.152242	1, -7	-2.152242	4, -7	-2.152242	3, -3	-2.152273	1, -2
-2.469267	3, -7	-2.469267	2, -6	-2.469267	2, -2	-2.4707	7, -2
-2.585787	4, -7	-2.585787	3, -6	-2.585786	3, -2	-2.5883	9, -2
-2.917609	7, -7	-2.917609	9, -6	-2.917605	8, -2	-2.932	2, -1
-3.234634	8, -7	-3.234634	2, -5	-3.234628	1, -1	-3.254	1, -1
-3.351154	1, -6	-3.351154	3, -5	-3.351137	2, -1	-3.449	3, -1
-3.566455	1, -6	-3.566455	3, -5	-3.566442	2, -1		
-4.000000	1, -7	-4.000000	4, -6	-4.000000	2, -2	-3.971	7, -1
						-4.021	6, -1
-4.433547	1, -6	-4.433547	6, -5	-4.433550	2, -1	-4.550	3, -1
-4.648848	2, -6	-4.648847	8, -5	-4.648855	2, -1		
-4.765367	1, -6	-4.765367	6, -5	-4.765370	1, -1	-4.746	1, -1
-5.082393	9, -7	-5.082393	8, -5	-5.082395	8, -2	-5.067	2, -1
		-5.371055	1, 0				
-5.414214	6, -7	-5.414214	2, -4	-5.414214	3, -2	-5.4115	9, -2
-5.530734	6, -7	-5.530734	4, -5	-5.530735	2, -2	-5.5292	7, -2
-5.847760	3, -7	-5.847760	2, -6	-5.847760	3, -3	-5.84772	1, -2
-6.030625	6, -1						
-6.179581	7, -8	-6.179581	1, -7	-6.179581	2, -4	-6.179581	1, -3
-6.613127	1, -8	-6.613127	2, -8	-6.613127	5, -5	-6.613127	2, -4
-6.828428	1, -6	-6.828428	2, -8	-6.828428	1, -5	-6.828428	5, -5
-6.829235	5, -2						
-7.261973	3, -5	-7.261852	5, -3	-7.261973	8, -7	-7.261973	2, -6
-7.261974	5, -4	-7.261973	1, -6				
-7.695518	6, -7	-7.695518	3, -5	-7.693653	7, -1	-7.695518	9, -8
-7.695519	9, -6	-7.695519	3, -4	-7.695518	9, -6		

TABLE 2

Note that we need to take k well above 25 to be able to guarantee all the eigenvalues and that the spurious eigenvalues are clearly recognisable by the size of the last elements of the corresponding vectors.

The General Matrix

The method of Lanczos is also applicable to unsymmetric matrices if used in a modified form. We now generate two sets of biorthogonal vectors, $\underline{y}^{(i)}$ and $\underline{z}^{(i)}$ by the recurrences

$$\begin{aligned} \gamma_{i+1}\,\underline{y}^{(i+1)} &= A\,\underline{y}^{(i)} - \alpha_i\,\underline{y}^{(i)} - \beta_{i-1}\,\underline{y}^{(i-1)} \\ \gamma_{i+1}\,\underline{z}^{(i+1)} &= A^T\underline{z}^{(i)} - \alpha_i\,\underline{y}^{(i)} - \beta_{i-1}\,\underline{z}^{(i-1)} \end{aligned} \qquad (12)$$

where $\underline{y}^{(0)} = \underline{z}^{(0)} = 0$, $\underline{y}^{(1)}$ and $\underline{z}^{(1)}$ are arbitrary and

$$\alpha_i = \underline{z}^{(i)T} A\,\underline{y}^{(i)}, \quad \beta_{i-1} = \underline{z}^{(i-1)T} A\,\underline{y}^{(i)}$$

and γ_{i+1} is so chosen that $\underline{z}^{(i+1)T}\,\underline{y}^{(i+1)} = 1$.

The matrix C_k is again defined by (6) and we now have the relations

$$AY_k = Y_k C_k, \quad A^T Z_k = Z_k C_k . \qquad (13)$$

The matrix C_k is not symmetric and in general we cannot use the method of bisections. Fox (1964, page 256) suggests using Muller's method for finding the roots of the characteristic polynomial. It is a simple matter to calculate $\det(C_k - \mu I)$ for any μ. However, if $\beta_i \geq 0$ for all i, then C_k is similar to the symmetric matrix C_k' given by

$$C_k' = \begin{bmatrix} \alpha_1 & \sqrt{\beta_1\gamma_2} & & \\ \sqrt{\beta_1\gamma_2} & \alpha_2 & \sqrt{\beta_2\gamma_3} & \\ & \sqrt{\beta_2\gamma_3} & \alpha_3 & \sqrt{\beta_3\gamma_4} \\ & & \ddots & \ddots \end{bmatrix} \qquad (14)$$

In this case we may use the method of bisections on C_k', and this is exactly the same as performing the method on C_k itself.

As in the symmetric case, there are difficulties over the termination of the sequence. Once again we should compute the eigenvectors of C_k and accept only those with small last elements. This will give us approximate eigenvalues and vectors of A with small residuals, but with an unsymmetric matrix a small residual does not necessarily imply an accurate eigenvalue and eigenvector.

CHAPTER 7

SOLUTION OF THE EIGENVALUE PROBLEM USING TRIANGULAR DECOMPOSITION

We again consider the eigenvalue equation

$$A\underline{x} = \lambda\underline{x}, \tag{1}$$

where A is a sparse matrix. We now assume that A is a band matrix and that we have sufficient storage for triangular decomposition to be possible. The methods of this chapter will be most valuable where there are few zero elements inside the band.

Inverse Iteration

If a single eigenvalue and corresponding eigenvector are required, then inverse iteration, given by the equation

$$(A - pI)\underline{x}^{(n+1)} = \underline{x}^{(n)}, \tag{2}$$

is satisfactory provided all the elementary divisors corresponding to the required eigenvalue are linear and that the matrix $(A - pI)$ does not have more than one distinct eigenvalue of smallest modulus. The vector $\underline{x}^{(n)}$ converges to an eigenvector of A and $\dfrac{\|\underline{x}^{(n)}\|}{\|\underline{x}^{(n+1)}\|}$ converges to $(\lambda - p)$ where λ is the corresponding eigenvalue. Any vector norm may be used. The iteration (2) is invariant with respect to a transformation of coordinates, so there is no loss in generality if it is written

$$\underline{Y}^{(n+1)} = J_p \underline{Y}^{(n)} \tag{3}$$

where J_p is the Jordan canonical form of $(A - pI)^{-1}$. By consideration of J_p it is easily seen that the convergence of the iteration is eventually governed by the equation

$$\underline{x}^{(n)} = \frac{1}{(\lambda - p)^n}\ \underline{x} + O\left\{\frac{1}{|\mu - p|^n}\right\} \tag{4}$$

where μ is the second nearest eigenvalue to p. Thus the convergence

is geometric and is very dependent on the closeness of p to the required eigenvalue.

The matrix $(A - pI)$ is factorised once only into the product of two triangular matrices, and during each iteration equation (2) is solved by forward and backward substitution. ~~In view of the relation (4) we may use Aitken's δ^2 process to accelerate the convergence.~~

Where $(A - pI)$ has more than one eigenvalue of smallest modulus or where not all the elementary divisors corresponding to the required root are linear, the process may not converge, or the convergence may be very slow. Suppose there are r eigenvalues of minimum modulus, then provided sufficient iterations (2) have been performed there exists a vector $\underline{\varepsilon}^{(l)}$, whose norm is small compared with the norm of $\underline{x}^{(l)}$, such that the dimension of the space spanned by the vectors

$$(A - pI)^{-k} (\underline{x}^{(l)} + \underline{\varepsilon}^{(l)}) \qquad (5)$$

for $k = 0, 1, 2, \ldots$ is r. We should therefore be able to obtain the r eigenvalues by r steps of Lanczos method of minimized iterations with $\underline{x}^{(l)}$ as starting vector. The problem of the effect of rounding errors is not nearly as severe as in the general case considered in Chapter 6. For suppose that γ is the modulus of the largest eigenvalue of $(A - pI)^{-1}$, then α_i, β_{i-1} and γ_{i+1} of equation (2) or (12) of Chapter 6 will be approximately equal to γ and so the component in $\underline{y}^{(i+1)}$ which is outside the space (5) cannot increase very rapidly. Its norm will increase by a factor that cannot exceed a constant depending on the size of $\underline{\varepsilon}^{(l)}$ and approximately equal to 3. We should therefore work to an accuracy of not less than 1 in 3^r. Under this condition the Lanczos vector will show a sharp drop in norm after the r^{th} step and the process should then be terminated. If no such sharp drop occurs, we may

assume that 1 was not taken sufficiently large. This technique may also be applied when several roots of A are nearly equal. The unwanted roots will be eliminated eventually by the inverse iteration, but the Lanczos iteration will save much work.

The Rayleigh Quotient Iteration

We see from (4) that the convergence of inverse iteration is geometric. We may obtain quadratic and sometimes cubic convergence at the expense of more work per iteration by the Rayleigh quotient iteration. This has been considered in some detail by Ostrowski (1957-9). For a symmetric matrix the iteration is given by the equations

$$\underline{x}^{(k+1)} = (A - \lambda_k I)^{-1} \underline{x}^{(k)} , \tag{6}$$

$$\lambda_{k+1} = \frac{\underline{x}^{(k+1)T} A \underline{x}^{(k+1)}}{\underline{x}^{(k+1)T} \underline{x}^{(k+1)}} . \tag{7}$$

Ostrowski shows that this iteration has cubic convergence. However, we must now perform a triangular decomposition at each iteration. From table 2 of chapter 1 we may calculate that the work is increased by factor of approximately $\frac{2r}{3}$, when the matrix has band-width (2r-1). Thus if r is not very large and the root and vector are required accurately, the Rayleigh quotient iteration will probably involve less work in all. It also provides an alternative method for a tridiagonal symmetric matrix and is likely to compare favourably with the method of bisections.

For non-symmetric matrices Ostrowski first considers the iteration given by the equations

$$\underline{\xi}^{(k+1)} = (A - \lambda_k I) \underline{\xi}^{(k)} , \tag{8}$$

$$\underline{\eta}^{(k+1)} = (A^T - \lambda_k I) \underline{\eta}^{(k)} , \tag{9}$$

$$\lambda_{k+1} = \frac{\underline{\eta}^{(k+1)T} A \underline{\xi}^{(k+1)}}{\underline{\eta}^{(k+1)T} \underline{\xi}^{(k+1)}} \tag{10}$$

He shows that, provided the required eigenvalue does not correspond to a non-linear elementary divisor, then in general the convergence is better than quadratic in the sense that

$$\lambda_{k+1} - \sigma = o\left[(\lambda_k - \sigma)^2\right] \quad \text{as } k \to \infty . \tag{11}$$

For an eigenvalue corresponding to a non-linear elementary divisor he forms the rational function $\phi(\lambda)$, defined by the equations

$$(A - \lambda I)\underline{\xi} = \underline{\alpha} \tag{12}$$

$$(A^T - \lambda I)\underline{\eta} = \underline{\beta} \tag{13}$$

$$\phi(\lambda) = \frac{\underline{\eta}^T A \underline{\xi}}{\underline{\eta}^T \underline{\xi}} , \tag{14}$$

where $\underline{\alpha}$, $\underline{\beta}$, are fixed and arbitrary vectors. He shows that, in general, the derivative of ϕ at the wanted eigenvalue σ is given by

$$\phi'(\sigma) = 1 - \frac{1}{L} , \tag{15}$$

so that

$$\phi(\lambda) = \sigma + (1 - \tfrac{1}{L})(\lambda - \sigma) + \sum_{\nu=2} a_\nu (\lambda - \sigma)^\nu , \tag{16}$$

where L is the greatest exponent of a non-linear elementary divisor corresponding to σ. It follows that iteration with $\phi(\lambda)$ is geometrically convergent to σ. A quadratically convergent process may be obtained by iterating with the function $\phi_L(\lambda)$ defined by

$$\phi_L(\lambda) = L\,\phi(\lambda) - (L-1)\lambda, \tag{17}$$

since, from (16), we have

$$\phi_L(\lambda) = \sigma + \sum_{\nu=2}^{\infty} L\, a_\nu (\lambda - \sigma)^\nu . \tag{18}$$

In general, of course, we will not know in advance the value of L. Ostrowski suggests that in such a case we should iterate with $\phi_2(\lambda)$.

If $L \neq 2$, then we may deduce from the iterates the true value of L since

$$\phi_2'(\lambda) = 1 - \frac{2}{L} . \tag{19}$$

He also considers a similar process applied to the ordinary Rayleigh quotient. The rational function $\phi(\lambda)$ is now defined by

$$(A - \lambda I)\, \underline{\xi} = \underline{\alpha} , \tag{20}$$

$$\phi(\lambda) = \frac{\underline{\xi}^T A\, \underline{\xi}}{\underline{\xi}^T \underline{\xi}} , \tag{21}$$

and he shows that in general

$$\phi(\lambda) = \lambda + (\lambda - \sigma)^L [\gamma + O(\lambda - \sigma)], \tag{22}$$

as $\lambda \to \sigma$. If a new rational function is defined by

$$\underline{\Phi}(\lambda) = \frac{\lambda\, \phi(\phi(\lambda)) - \phi(\lambda)^2}{\lambda - 2\,\phi(\lambda) + \phi(\phi(\lambda))} \tag{23}$$

then

$$\underline{\Phi}'(\sigma) = 1 - \frac{1}{L} , \tag{24}$$

where L is again the greatest exponent of an elementary divisor corresponding to σ. A quadratically convergent process may be obtained as in equation (17), but this would require the solution of a total of four sets of linear equations, whereas only three are needed to form Householder's function

$$\psi(\lambda) = \frac{\lambda\, \phi(\underline{\Phi}(\lambda)) - \phi(\lambda)\, \underline{\Phi}(\lambda)}{\phi(\underline{\Phi}(\lambda)) - \phi(\lambda) - \underline{\Phi}(\lambda) + \lambda} \tag{25}$$

and iteration with this shows quadratic convergence.

Ostrowski also shows that if the ordinary Rayleigh quotient iteration given by equations (6) (7) is applied to a non-symmetric matrix, then, in general, the convergence is quadratic provided that the roots are real and correspond to linear elementary divisors.

He concludes that this is an advisable procedure to adopt since the work per iteration is about half that of the generalised Rayleigh quotient iteration defined by equations (12) (13) and (14). However it is quite rare to be able to guarantee these conditions and equation (25) involves approximately 1½ times as much work per iteration as equation (17). We feel bound, therefore, to recommend (17) for use in the general case. It is, of course, still valid for L = 1, and in the absence of any more precise information this value of L should be used to start the iteration.

Rutishauser's L-R Transformation

If several eigenvalues are required a powerful technique is the L-R transformation of Rutishauser (1958). Each step of the iteration is given by the equations

$$A_k = L_k R_k , \tag{26}$$

$$A_{k+1} = R_k L_k \tag{27}$$

$$A_1 = A, \tag{28}$$

where (26) represents the usual triangular decomposition. Each step is a similarity transformation since

$$\begin{aligned} A_{k+1} &= R_k L_k \\ &= L_k^{-1} L_k R_k L_k \\ &= L_k^{-1} A_k L_k , \end{aligned} \tag{29}$$

and, in fact,

$$A_k = B_k^{-1} A B_k = P_k A P_k^{-1} \tag{30}$$

where $B_k = L_1 L_2 \cdots L_k$

and $P_k = R_k R_{k-1} \cdots R_1$.

It is easily verified that

$$A^k = B_k P_k , \tag{31}$$

so that B_k and P_k give the triangular decomposition of A^k. Note that the matrices A_k are all band matrices with the same band width.

Rutishauser proves the following theorem.

Theorem If the matrices B_k as defined above converge for $k \to \infty$, then $\lim_{k\to\infty} A_k$ exists and is an upper triangular matrix.

Proof If $B_\infty = \lim_{k\to\infty} B_k$ exists,

then $\lim_{k\to\infty} L_k = \lim_{k\to\infty} B_{k-1}^{-1} B_k$ is the unit matrix.

Further since $R_k = B_k^{-1} A B_{k-1}$, this converges and so

$$A_k = L_k R_k \text{ converges and is upper triangular.}$$

Rutishauser goes on to prove another theorem.

Theorem If A has latent roots with distinct moduli,

$$|\lambda_1| > |\lambda_2| \cdots > |\lambda_n| \tag{32}$$

then there exists a matrix X such that

$$A = X \operatorname{diag}(\lambda_1 \cdots \lambda_n) X^{-1} . \tag{33}$$

Suppose that no leading principle submatrix of X or X^{-1} is singular, then A_k converges to an upper triangular matrix with the latent roots of A on its diagonal in descending order of modulus.

Proof We give here a new proof, which permits a generalisation to be applied more easily.

The conditions on X and X^{-1} imply that the triangular decompositions

$$X = CD \quad \text{and} \quad X^{-1} = EF \tag{34}$$

exist. Thus A^k is given by

$$A^k = CD \text{ diag } (\lambda_1^k \dots \lambda_n^k) EF. \tag{35}$$

Now $D \text{ diag } (\lambda_1^k \dots \lambda_n^k) E =$

$$\begin{bmatrix} d_{11}\lambda_1^k e_{11} + O(\lambda_2^k) & d_{12}\lambda_2^k e_{22} + O(\lambda_3^k) & d_{13}\lambda_3^k e_{33} + O(\lambda_4^k) & \dots & d_{1n}\lambda_n^k e_{nn} \\ d_{22}\lambda_2^k e_{21} + O(\lambda_3^k) & d_{22}\lambda_2^k e_{22} + O(\lambda_3^k) & d_{23}\lambda_3^k e_{33} + O(\lambda_4^k) & \dots & d_{2n}\lambda_n^k e_{nn} \\ d_{33}\lambda_3^k e_{31} + O(\lambda_4^k) & d_{33}\lambda_3^k e_{32} + O(\lambda_4^k) & d_{33}\lambda_3^k e_{33} + O(\lambda_4^k) & \dots & d_{3n}\lambda_n^k e_{nn} \\ \dots & \dots & \dots & \dots & \dots \\ d_{nn}\lambda_n^k e_{n1} & d_{nn}\lambda_n^k e_{n2} & d_{nn}\lambda_n^k e_{n3} & \dots & d_{nn}\lambda_n^k e_{nn} \end{bmatrix} \tag{36}$$

Now for $r = 1, 2 \dots n$, $d_{rr} \neq 0$ and $e_{rr} \neq 0$ since neither X nor X^{-1} have a singular leading principle submatrix. Hence if the matrix in (36) is split into triangles then in the limit the lower triangular part will be just the unit matrix. It follows that when A^k is split into triangles,

$$A^k = B_k P_k , \tag{37}$$

then B_k converges to C. Hence $\lim A_k$ exists and is given by

$$\begin{aligned} A_\infty &= B_\infty^{-1} A B_\infty \\ &= B_\infty^{-1} X \text{ diag } (\lambda_1 \dots \lambda_n) X^{-1} B_\infty \\ &= D \text{ diag } (\lambda_1 \dots \lambda_n) D^{-1} \end{aligned} \tag{38}$$

and the latter matrix is upper triangular with diagonal elements $\lambda_1 \dots \lambda_n$.

For the generalisation of this result we consider A's Jordan ~~canonised~~ canonical form J so ordered that the moduli of the diagonal elements

form a non-increasing sequence. Further let J be partitioned so that

$$J = \text{diag}\,(\lambda_1 \ \ldots \ \lambda_n), \tag{39}$$

where each λ_i is a principle submatrix of J having diagonal elements all of equal modulus. Then the analysis proceeds as before, where now the letters corresponding to submatrices rather than elements, but the triangular decompositions are still point decompositions. We justify the expressions $O(\lambda_i^k)$ in (36) by the result of Varga (1962, page 64), that if λ_i is of order p_i then

$$\| \lambda_i^k \| \sim \binom{k}{p_i - 1} \left[\rho(\lambda_i) \right]^{k-p_i+1} \quad \text{as } k \to \infty , \tag{40}$$

where $\rho(\lambda_i)$ is the spectral radius of λ_i. In the triangular decomposition of (36), the lower triangular part is now block diagonal. It follows that the off-diagonal blocks of B_k converge to the corresponding blocks of C and hence that in the limit L_k is block diagonal. Unfortunately we cannot deduce at once that A_k is block upper triangular, since this requires that R_k be bounded. However if block triangular decomposition is used, the proof carries through in exact correspondence with the earlier case and convergence is assured. We suggest therefore that in a practical case point decomposition should be used until the block matrix becomes apparent and thereafter block iteration be used.

It should be noted that the convergence of the iteration is dependent on the ratios $\left|\frac{\lambda_{k+1}}{\lambda_k}\right|$. We shall later use the property to accelerate the converge for the iteration when applied to a positive-definite matrix. If this ratio is very near unity then we may group together the two eigenvalues into a block and use the above result in its block form. We therefore have convergence to a block upper triangular matrix much sooner than convergence to

an upper triangular matrix. There is no need to iterate beyond convergence to this block form, since the eigenvalues of the blocks may then be found by any of the standard techniques.

Fox (1964) considers a slight modification of the process in which interchanges are included in the triangular decompositions. These are just as necessary here as in the solution of linear equations. The iteration is now given by the equations

$$A_k = I_k^{-1} L_k R_k , \tag{41}$$

and

$$A_{k+1} = R_k I_k^{-1} L_k , \tag{42}$$

where I_k is a permutation matrix, so chosen that no element of L_k is greater than unity in modulus. We know of no proof of the convergence of this iteration. Some numerical experiments would be interesting, and we suspect that if the eigenvalues are real and distinct no further interchanges will be required after a finite number of steps. Thereafter, Rutishauser's theorem gives us sufficient conditions for convergence. Perhaps the safest course would be to return to Rutishauser's original iteration after a number of iterations given by (41) and (42). A check should be kept on the size of the elements of L_k and additional guarding figures kept if necessary. In any case we cannot allow an indefinite number of interchanges since we are here considering band matrices and we wish to keep the band-width as small as possible.

When A is positive definite the decomposition (26) is replaced by the symmetric decomposition of Cholesky, given by

$$A_k = L_k L_k^T , \tag{43}$$

and A_{k+1} is defined by

$$A_{k+1} = L_k^T L_k \,. \tag{44}$$

This iteration preserves the symmetry, so that our storage requirement is approximately halved. Rutishauser shows that under this iteration A_k is bound to converge to a diagonal matrix, although the elements down the diagonal may not necessarily be ordered. If A is symmetric and non-definite we work with $(A+pI)$, choosing p so that we obtain a diagonally dominant and hence positive-definite matrix. In this way we avoid interchanges and an increase in band-width.

Rutishauser (1958) has proposed a very useful accelerating device for the positive-definite case, which is particularly powerful where we require several of the smallest latent roots. We have seen that the convergence depends on the ratios $\frac{\lambda_{k+1}}{\lambda_k}$. If we have a good lower bound for λ_n, say $\lambda_n > \mu$, and we iterate on the matrix $(A - \mu I)$, then convergence in the last row and column will be much improved. Now we may evaluate $\det(A - \mu I)$ as a by-product of the triangular decomposition. If we evaluate this for two different values of μ, each known to be less than λ_n, then we may obtain as a lower bound for λ_n, the quantity

$$\mu = \frac{\mu_1 \det(A-\mu_2) - \mu_2 \det(A-\mu_1)}{\det(A-\mu_2) - \det(A-\mu_1)} \,, \tag{45}$$

which comes from fitting a straight line through the two known points. It must be a lower bound on account of the convexity of the characteristic polynomial beyond its smallest root. The convergence can, indeed, be still further improved by the use of a different value μ_k of μ for each L-R step. We find μ_k by (45) with μ_1, μ_2 replaced by the greatest two of $\mu_1, \mu_2 \cdots \mu_{k-1}$. This process is quadratically convergent. Once convergence in the last row and column has been obtained, these may be deleted and the same procedure applied to the

remaining (n-1) x (n-1) matrix, and so on.

A similar acceleration may be applied to the iteration for a general matrix if its smallest eigenvalue is real. If the off-diagonal elements in the last row and column of A_i are small, then this last diagonal element, $a_{nn}^{(i)}$ will approximate to an eigenvalue of A_i and the convergence will be accelerated if A_i is replaced by

$$A_i' = A_i - a_{nn}^{(i)} I. \tag{46}$$

We may even use this procedure for a complex root if we are prepared to use complex arithmetic. Rutishauser remarks that when the symmetric iteration, as given by equations (43) and (44), is applied to a positive-definite matrix, then the answers obtained have been found to be very accurate. We will now demonstrate this property. Suppose that the exact value of $L_k L_k^T$ is given by the equation

$$L_k L_k^T = A_k + E_k, \tag{47}$$

then Wilkinson (1961, page 305) shows that if we use binary-place arithmetic, with double-length accumulation of scalar products, and if

$$(n+1)2^{-t} \| A_k^{-1} \| < 1 \quad \text{and} \quad | a_{ij} | < 1 - 2^{-t}, \tag{48}$$

then

$$|E_{ij}| \leq \begin{cases} 2^{-t-1} & |l_{ii}| \quad \text{for } j > i \\ 2^{-t-1} & |l_{jj}| \quad i > j \\ 2^{-t} & |l_{ii}| \quad i = j \end{cases}. \tag{49}$$

Wilkinson shows further that $|l_{ii}| < 1$. It follows that

$$\| E_k \| < 2^{-t} r, \tag{50}$$

where $(2r-1)$ is the band-width of A_k. Let us assume that

$$\| A_k \| < 1 - p_k 2^{-t}. \tag{51}$$

where p_k is a positive integer. This ensures that the second condition of (48) holds. We also have that

$$\begin{aligned} |l_{ij}| \leq \| L_k \| &= \| A_k + E_k \|^{\frac{1}{2}} \\ &\leq \left[\| A_k \| + \| E_k \| \right]^{\frac{1}{2}} \\ &< 1 \end{aligned} \tag{52}$$

provided that $p_k > r$.

It follows that in the computation of $L_k^T L_k$, provided double-length arithmetic is used for scalar products, the only significant rounding error will be in the final truncation. Suppose that

$$A_{k+1} = L_k^T L_k + F_k , \tag{53}$$

then

$$\| F_k \| \leq 2^{-t} (2r-1) \| A_{k+1} \| , \tag{54}$$

and we deduce that

$$\| A_{k+1} \| < \frac{1 - (p_k - r)2^{-t}}{1 - (2r-1)2^{-t}} < 1 - (p_k - 3r)2^{-t} \tag{55}$$

provided $2r(2r-1)2^{-t} \leq 1$. Thus if $p_1 > 3rk$, the required conditions hold for all iterations up to the kth. The matrices A_k, E_k and $L_k L_k^T$ are all symmetric and it is a trivial consequence of the minimax characterisation of the eigenvalues that corresponding eigenvalues of A_k and $L_k L_k^T$ differ by not more than $\| E_k \|$. Similarly the eigenvalues of $L_k^T L_k$ and A_{k+1} differ by not more than $\| F_k \|$. Thus in each step the loss of accuracy in each eigenvalue is not greater than $3r\, 2^{-t}$. In fact this will be an over-estimate

during the later stages when the matrices A_k are near diagonal. We may obtain a more accurate result if we are prepared to store the matrices A_k to double-length precision. In this case $\|F_k\|$ is negligible and the loss of accuracy at each step is bounded by $r.2^{-t}$.

CHAPTER 8

ITERATIVE METHODS FOR THE EIGENVALUE PROBLEM

In this chapter we consider the same problem as in Chapters 6 and 7, namely the determination of one or more of the solutions of the equation

$$A \underline{x} = \lambda \underline{x} . \tag{1}$$

We assume now that either A is too large to be stored inside our computer or that is is a band matrix with so many zeros inside its band that the methods of Chapter 7 are uneconomic.

Direct Iteration

For a symmetric matrix we can use the "direct" iteration given by

$$\underline{x}^{(i+1)} = (A-pI)\underline{x}^{(i)} \tag{2}$$

which converges to the eigenvector corresponding to the smallest or largest root of A, according to the choice of p. The situation here is in exact parallel with that in Chapter 5. With equal ease we can form the vector

$$\underline{y}^{(i)} = p_i (A)\, \underline{x}^{(0)}, \tag{3}$$

where $p_i(\mu)$ is any polynomial of degree i. To obtain the vectors corresponding to either the smallest or largest root a good choice for $p_i(\mu)$ is the ith Chebyshev polynomial over the range of all the other roots of A. If all the other roots lie in the range (a, b) then the polynomial $p_i(\mu)$ will be given by the equation

$$p_i (\mu) = T_i \left(\frac{-2\mu+a+b}{b-a} \right) , \tag{4}$$

and the iteration takes the form

$$\underline{y}^{(i+1)} = \frac{2}{b-a}\left[(a+b)\,\underline{y}^{(i)} - 2A\,\underline{y}^{(i)}\right] - \underline{y}^{(i-1)}. \qquad (5)$$

We may judge the progress of the iteration by the ratio $\|\underline{y}^{(i+1)}\| / \|\underline{y}^{(i)}\|$ since the vector $\underline{y}^{(i)}$ will eventually behave according to the relation

$$\underline{y}^{(i+1)} = (\gamma + \sqrt{\gamma^2 - 1})\underline{y}^{(i)}, \qquad (6)$$

where

$$\gamma = \frac{-2\lambda + a + b}{b - a}, \qquad (7)$$

and λ is the required root. The root itself is best found from the Rayleigh quotient

$$\lambda \simeq \frac{\underline{y}^{(i)T} A \underline{y}^{(i)}}{\underline{y}^{(i)T} \underline{y}^{(i)}}. \qquad (8)$$

This iteration is very similar to that given by equation (24) of Chapter 5; the general properties of the convergence are also similar, and means of obtaining estimates of the two end eigenvalues and corresponding eigenvectors were indicated there. However we do not now need the condition $p_i(0) = 1$. We may use a technique similar to that of Chapter 5 for improving the values of a and b, for if λ is the only eigenvalue outside the range (a, b) the residual

$$\underline{r}^{(i)} = \lambda\,\underline{y}^{(i)} - A\,\underline{y}^{(i)} \qquad (9)$$

will not show any regular behaviour. If, however, there is a second eigenvalue λ_1 outside the range then $\underline{r}^{(i)}$ will behave as

$$\underline{r}^{(i+1)} \simeq (\gamma_1 + \sqrt{\gamma_1^2 - 1})\,\underline{r}^{(i)}, \qquad (10)$$

where

$$\gamma_1 = \frac{-2\lambda_1 + a + b}{b - a}. \qquad (11)$$

As in Chapter 5 we may either use $\| \underline{r}^{(i+1)} \| / \| \underline{r}^{(i)} \|$ to estimate λ_1 from (10) and (11), or, we may use the Rayleigh quotient

$$\lambda_1 \simeq \frac{\underline{r}^{(i)T} A \underline{r}^{(i)}}{\underline{r}^{(i)T} \underline{r}^{(i)}} \quad . \tag{12}$$

In principle this method can be extended to obtain successive eigenvalues and corresponding vectors, starting from one end of the range of eigenvalues. The components of the vectors already found must be subtracted out at regular intervals. Rutishauser (1959) proposed a modification of this method that allows several of the smallest (or largest) eigenvalues to be computed simultaneously. We choose the range (a,b) to include all but the required eigenvalues and iterate until the current vector lies in the space of the corresponding eigenvectors. Suppose that the eigenvalues and corresponding vectors are λ_j and $\underline{\xi}_j$ and that the starting vector can be expanded in the form

$$\underline{y}^{(0)} = \sum_{j=1}^{n} a_j \underline{\xi}_j \, , \tag{13}$$

so that

$$\underline{y}^{(i)} = \sum_{j=1}^{n} T_i(\gamma_j) \underline{\xi}_j \, , \tag{14}$$

where

$$\gamma_j = \frac{-2\lambda_j + a + b}{b - a} \quad . \tag{15}$$

If λ_j lies outside the range (a,b) then it is easily seen that as $i \to \infty$, $T_i(\gamma_j)$ behaves as

$$T_i(\gamma_j) = \tfrac{1}{2}\left(\gamma_j + \sqrt{\gamma_j^2 - 1}\right)^i + o(1), \tag{16}$$

and for γ_j inside the range we have

$$| T_i(\gamma_j) | \leqslant 1 \tag{17}$$

It is therefore reasonable to suppose that by the time the current vector is dominated by the eigenvector $\underline{\xi}_j$ where λ_j lies outside (a,b), then the error term in (16) may be neglected. We now apply Lanczos' method of minimized iterations to the vectors $\underline{y}^{(i)}, \underline{y}^{(i+1)} \ldots \underline{y}^{(i+s)}$. The Lanczos vectors are not obtained by the usual process but instead an ordinary Schmidt orthogonalization process is used to find an orthogonal set $\underline{z}^{(i)}, \ldots \underline{z}^{(i+s)}$. In this way we obtain eigenvectors of A and the numbers $(\gamma_j + \sqrt{\gamma_j^2 - 1})$ from which the corresponding eigenvalues may be found. Again we may use the Rayleigh quotients to obtain better estimates of the values. The limitation of this technique was recognised by Rutishauser in a comment that it would be difficult, in actual computing, to find more than two or three eigenvalues in this way. The trouble is that the Chebyshev polynomials increase very rapidly outside the range (a,b) and the vectors $\underline{y}^{(i)}$ will have larger components of eigenvectors corresponding to the extreme values. This will present no grave difficulty as long as we can keep a few additional guarding figures, and provided s is not large.

An alternative suggested by Ginsburg (1959) is to use the Chebyshev iteration as in Rutishauser's method above and then apply an ordinary Lanczos iteration with the resulting vector as $\underline{y}^{(1)}$. We can expect a good result for the required eigenvalues with far fewer iterations than with the straightforward use of Lanczos, and this is confirmed by Ginsburg in practical applications.

Use of SOR Iteration

If we require a single eigenvalue and eigenvector or wish to improve an approximation we may be tempted to use the inverse iteration of Chapter 7, given by the equation

$$(A - pI)\, \underline{x}^{(i+1)} = \underline{x}^{(i)}, \qquad (18)$$

and an iterative method to solve the linear equations. However if

p is close to an eigenvalue, the matrix $(A - pI)$ is nearly singular and any iterative method can, at best, converge very slowly. If A is symmetric and p is exactly equal to its smallest eigenvalue, then it is easy to confirm that the successive over-relaxation iteration matrix, L_w, with any relaxation factor w in the range $0 < w < 2$, has dominant eigenvalue unity.

For suppose

$$(A - pI) = D - L - U, \tag{19}$$

where D is diagonal and L and U are respectively strictly lower and upper triangular, then we have the relation

$$(A - pI)\underline{y} = 0,$$

that is

$$(D - L - U)\underline{y} = 0, \tag{20}$$

where $\underline{y}$ is an eigenvector of A corresponding to eigenvalue p. It follows that

$$\begin{aligned} L_\omega \underline{y} &= (D - wL)^{-1} (wU - (w-1)D)\underline{y} \\ &= (D - wL)^{-1} (D - wL)\underline{y} \\ &= \underline{y} \end{aligned} \tag{21}$$

Let us assume that every row of the matrix A has at least one non-zero off-diagonal element, since otherwise the problem is trivially reducible. Then it follows that all the elements of D are positive. No element of D can be negative since $(D - L - U)$ is positive semi-definite. Suppose $d_{ii} = 0$ that $a_{ij} = a_{ji} \neq 0$ and that we perform a permutation similarity transformation on $(D - L - U)$ to bring the ith and jth rows and columns into the 1st and 2nd rows and columns. Then it is easily seen that the leading 2 x 2 submatrix

is not positive definite. We conclude that $d_{ii} > 0$ for all i. Hence the argument of Chapter 2, equations (61) and (63), is applicable, and the error vector will ultimately be an eigenvector of A corresponding to the eigenvalue p. Furthermore from consideration of the Jordan canonical form of L_w it may be seen that convergence is impossible unless L_ω has no non-linear elementary divisors corresponding to the eigenvalue unity. We shall find this property useful below (page 103).

Using subscripts to denote SOR iteration, we have from equation (18) the relation

$$\underline{x}_j^{(i+1)} = \underline{x}^{(i+1)} + L_w^j(\underline{x}_0^{(i+1)} - \underline{x}^{(i+1)}). \tag{22}$$

For large j the vector $L_w^j(\underline{x}_0^{(i+1)} - \underline{x}^{(i+1)})$ will change little if p is a good approximation to the smallest eigenvalue of A and will not be small compared with $\underline{x}^{(i+1)}$. Thus there is little gain in solving (18) over solving the homogenous set

$$(A - pI)\underline{x} = \underline{0}, \tag{23}$$

which will involve a simpler iteration and require less storage. In this case equation (22) takes the form

$$\underline{x}_j = L_w^j \underline{x}_0 . \tag{24}$$

The rate of convergence will be governed by the second eigenvalue of L_w. When A has "property A" we may use the techniques of Chapter 3, with a slight modification, to estimate the optimum value of w. We require an estimate of the current rate of convergence from successive values of the displacement vector, $\underline{x}_{j+1} - \underline{x}_j$. However this vector may be polluted by a component of the dominant eigenvector of L_w if the dominant eigenvalue is not exactly unity. The current vector may be assumed to be an approximation to this dominant eigenvector, and we overcome the difficulty by applying the procedure

to the vector of changes less its component in the direction of the current vector. The iteration should be continued until the vector of changes lies substantially in the same direction as the current vector. We then find an improved estimate for the eigenvalue from the Rayleigh quotient of the current vector and use this new value in (23). We will show that in virtue of the stationary property of the Rayleigh quotient this iteration ~~involving equations (23) and (25)~~ converges quadratically in the limit. For if $L_w(\epsilon)$ is the SOR iteration matrix for $p = \lambda + \epsilon$, then

$$L_w(\epsilon) = (D + \epsilon I - wL)^{-1} (wU - (w-1)(D + \epsilon I)) \tag{25}$$

$$= L_w(0) + \epsilon E_w + O(\epsilon^2), \tag{26}$$

where E_w is a matrix that depends on w, D, L, U. Suppose that the Jordan canonical of $L_w(0)$ is J and that

$$J = X^{-1} L_w X, \tag{27}$$

so that

$$X^{-1} L_w(\epsilon) X = J + \epsilon X^{-1} E_w X + O(\epsilon^2). \tag{28}$$

Now we know that the part of J corresponding to the unit root is diagonal. It follows that for sufficiently small ϵ the dominant root of $L_w(\epsilon)$ differs from unity by a term of order ϵ, and the corresponding vectors also differ by a term of order ϵ. It follows from the stationary property of the Rayleigh quotient that the ~~in~~ error ~~in p'~~ is of order $O(\epsilon^2)$, and the convergence is therefore quadratic in the limit.

This procedure has been described briefly by Forsythe and Wasow (1960, page 375). As they point out, it is not applicable to other than the smallest or largest eigenvalues, but we have found it very powerful in this case and feel that it deserves attention.

The second eigenvalue may be found by a slight modification of the procedure. We iterate until we have found the dominant eigenvector of L_w and then iterate again, subtracting out the component of this vector at regular intervals. In this way we find the required eigenvector of L_w. However it is difficult to see how to choose the best value of w. In principle the technique may be extended indefinitely to produce any eigenvalue and corresponding vector, but it is clearly too clumsy to be of practical use for more than a few of the lower eigenvalues. An obvious alternative for a middle eigenvalue is to use Kaczmarz iteration on equation (23). If p is exactly equal to an eigenvalue of A, we are applying Kaczmarz iteration to a singular matrix, and the hyperplanes will have in common the space spanned by the eigenvectors corresponding to the eigenvalue zero. The distance from the current point to this space will decrease steadily and, by the same argument as in Chapter 2, the iteration will converge. In this case we find an eigenvector corresponding to the value zero. It also follows that the iteration matrix has spectral radius unity and the eigenvalue unity corresponds to no non-linear elementary divisors, and hence that if p differs from an eigenvalue of A by ε, then the dominant eigenvector of the iteration matrix will differ from the corresponding eigenvector of A by a term of order $O(\varepsilon)$. The overall convergence of the method will therefore remain quadratic.

Use of Kaczmarz Iteration

For a general unsymmetric matrix we need approximations to both the right and left eigenvectors of A in order to calculate a Rayleigh quotient. Thus we must apply Kaczmarz iteration to both the equations

$$(A - pI)\underline{x} = \underline{0} \tag{29}$$

and

$$(A^T - pI)\underline{y} = \underline{0}, \tag{30}$$

and take as the new approximate eigenvalue the quantity

$$p' = \frac{\underline{y}^T A \underline{x}}{\underline{y}^T \underline{x}} . \tag{31}$$

The convergence of these iterations may be improved by the use of symmetric Kaczmarz with Chebyshev acceleration. If the iteration matrix is T, then the accelerated iterates will be given by

$$\underline{y}^{(i)} = T_i(T)\underline{y}^{(0)}, \tag{32}$$

when $T_i(\mu)$ is the ith Chebyshev polynomial over the range (0,b) and $b < 1$. As with other Chebyshev iterations, we may correct an under-estimate of b. We consider the vector

$$\frac{\underline{y}^{(i)}}{T_i(1)} - \frac{\underline{y}^{(i-1)}}{T_{i-1}(1)} , \tag{33}$$

less its components in the direction $\underline{y}^{(i)}$. If b is an over-estimate this will behave in a random fashion, but if it is an under-estimate it will eventually lie substantially in the direction of the subdominant eigenvector of the iteration matrix T.

The Kaczmarz iteration will in general show rather slow convergence since applied to the matrix A it converges at the same rate as Gauss-Seidel applied to AA^T. If a number of eigenvalues are needed this is unlikely to be an economical method, and Lanczos' method is to be preferred. The usefulness of the methods of this chapter lies in finding an isolated eigenvalue or in improving an approximate eigenvalue and corresponding eigenvector.

CHAPTER 9

THE NUMERICAL RANK OF A MATRIX AND THE SOLUTION OF THE ILL-CONDITIONED PROBLEM

In this chapter we shall use an order relation between matrices. If the matrices A and B have elements (a_{ij}) and (b_{ij}) and the same number of rows and the same number of columns, then we say $A \leqslant B$ or $A < B$ if $a_{ij} \leqslant b_{ij}$ or $a_{ij} < b_{ij}$ for all i and j. We shall ~~also~~ denote by A^+ the matrix whose elements are $(|a_{ij}|)$.

Numerical Rank

If a matrix is given numerically the classical definition of rank has no significance unless the elements can be expressed exactly inside the computer. However it should be possible to state error bounds for each element. If there is a singular matrix which differs from the given matrix, element by element, by less than the appropriate bound then it seems reasonable to say that the matrix is "numerically singular". Extending this idea leads us to define "numerical rank". We say that the numerical rank of the matrix A with respect to the matrix E, where $E > 0$ and A, E have the same number of rows and columns, is the minimum rank of the set of matrices [B] where

$$A - E \leqslant B \leqslant A + E. \tag{1}$$

This definition reduces to the ordinary definition if $E = 0$. Unfortunately it may be difficult to calculate the value of the numerical rank in a practical case, but we give below methods for finding upper and lower bounds. These may coincide or we may be satisfied with one of them.

We show first that if the matrix A is non-singular then it is numerically non-singular with respect to the non-negative matrix E if

$$\| (A^{-1})^{+} E \| < 1, \tag{2}$$

where the norm is the row norm. To prove this we suppose that $B = A + F$ where $F^{+} \leq E$, ~~Then~~ so that

$$A^{-1}B = I + A^{-1}F. \tag{3}$$

But $$\| A^{-1}F \| \leqslant \| A^{-1+}F^{+} \| \leqslant \| A^{-1+}E \| < 1, \tag{4}$$

and it follows from Gershgorin's theorem that B is non-singular, which is the required result.

Suppose that the elements of E are all of the same order of magnitude or that the matrices A and E can be so scaled, by rows and columns, that the new E is of this form. We can then find a lower bound for the numerical rank by performing Gaussian elimination with full interchanges. If A_i is a leading principal submatrix of the permuted matrix A and E_i is the corresponding submatrix of E, then the numerical rank is not less than the maximum value of i for which $\| A_i^{-1+} E_i \| < 1$.

If the numerical rank is completely unknown this method will involve a great deal of computation. However it is true in practical cases that we generally have a good idea of its approximate value. In a linear system arising from an integral equation, for example, the numerical rank may be very small, say four or five, when the order of the matrix is quite large. Very little work is required to check $\| A_i^{-1+} E_i \|$ for i = 1, 2 ... 5. In a common case we wish to confirm that the matrix is non-singular. If the inverse is required in any case, little extra work is necessary, and some economy is in fact possible. We have the triangular decomposition

$$A_i = L_i U_i, \tag{5}$$

so that

$$\| (A_i^{-1+}) E_i \| \leqslant \| U_i^{-1+} L_i^{-1+} E_i \|$$
$$\leqslant \| U_i^{-1} \| \| L_i^{-1} \| \| E_i \| , \qquad (6)$$

and the latter expression may be calculated comparatively easily. If we find it is greater than unity we must revert to the calculation of $\| A_i^{-1+} E_i \|$.

To find an upper bound for the numerical rank we again assume that the elements of E are all of the same order of magnitude and perform Gaussian elimination with full interchanges. For simplicity we assume that A is correctly ordered at the start. If at any stage the last r rows of the reduced matrix are smaller in modulus than the corresponding elements of the error matrix E, then the numerical rank of A is not greater than (n-r). This test is unfortunately rather clumsy in the sense that we have not considered the effect of perturbations in the first (n-r) rows. There is a better test for the case r = 1. If the decomposition of A is given by

$$A = LU, \qquad (7)$$

and if F is a matrix of small changes in the elements of A then we have the equation

$$L^{-1}(A + F) = U + L^{-1}F. \qquad (8)$$

If $\underline{e}_n$ is the last column of E, $\underline{l}_n^T$ is the last row of L^{-1} and u_{nn} is the last element of U, and if

$$| u_{nn} | \leq \underline{l}_n^{T+} \underline{e}_n \qquad (9)$$

it is clear that A is numerically singular with respect to E.

The Solution of Ill-conditioned Sets of Equations

If a matrix has rank (n-r) it has r linearly independent

eigenvectors corresponding to the eigenvalue zero. This will be true of the matrix B of the definition of numerical rank. It follows that the solution vector $\underline{x}$ of the set of equations

$$(A + F)\underline{x} = \underline{b} \tag{10}$$

has an arbitrary component in this space of dimension r. In practice, the solution required is probably that with zero component in this space, but any other solution may be found with the help of approximations to the relevant eigenvectors, which we will need in any case. Suppose that we can order the rows and columns of A so that the leading principle submatrix of order (n-r) is non-singular, or equivalently that the interchanges are such that this would have been the form of A if no interchanges had been used. We take as zeros the last r elements of the solution and solve for the remaining elements. However this solution will in general contain a component in the space of eigenvectors corresponding to the value zero. Suppose an independent set of these eigenvectors is $\underline{x}_i$ and a corresponding biorthogonal set for A^T is $\underline{y}_i$. Then we obtain the required solution

$$\underline{x}' = \underline{x} - \sum_{r=1}^{r} \frac{\underline{y}_i^T \underline{x}}{\underline{y}_i^T \underline{x}_i} \underline{x}_i . \tag{11}$$

The case of symmetric A is simpler since $\underline{x}_i = \underline{y}_i$.

Matrices sometimes occur having a numerical rank that is small compared with the order. In such a case it is more economical to find the (n-r) eigenvectors corresponding to eigenvalues that are not small. We find the components of $\underline{b}$ along each eigenvector and hence the components of the solution $\underline{x}$.

CHAPTER 10

PROBLEMS IN COMPLEX LINEAR ALGEBRA

We have assumed so far that all the coefficients in our matrices have been real. The majority of practical matrix problems are of this nature, but complex matrices do occur and we consider briefly here how the techniques we have considered may be extended to this case.

The methods of Chapter 1 all generalise to complex matrices with no difficulty. We must replace transposed matrices by conjugate transposed matrices and symmetric matrices by Hermitian matrices. We should note in particular that interchanges are not required ~~from~~ for positive-definite Hermitian matrices. On most computers the amount of work will be about four times that for the real case. The most useful comparison is in the time taken to accumulate a scalar product. On Mercury, using machine language, the ratio is 7: 25.

An alternative which avoids complex arithmetic is to replace the equation

$$(A + iB)(\underline{x} + i\,\underline{y}) = \underline{b} + i\,\underline{c}, \tag{1}$$

where A, B, $\underline{x}$, $\underline{y}$, $\underline{b}$, $\underline{c}$ are real, by the equation

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \begin{bmatrix} \underline{b} \\ \underline{c} \end{bmatrix}. \tag{2}$$

The equation (2) has real coefficients and is solved by one of the standard techniques. If (A + iB) is Hermitian, then $\begin{pmatrix} A & -B \\ B & A \end{pmatrix}$ is symmetric and if (A + iB) is positive definite then so is $\begin{pmatrix} A & -B \\ B & A \end{pmatrix}$.

The form (2) is unsuitable for band matrices since the band ~~matrix~~ structure is destroyed. We may use instead the equation

$$\begin{bmatrix} a_{11} & -b_{11} & a_{12} & -b_{12} & \cdot & \cdot \\ b_{11} & a_{11} & b_{12} & a_{12} & \cdot & \cdot \\ a_{21} & -b_{21} & a_{22} & -b_{22} & \cdot & \cdot \\ b_{21} & a_{21} & b_{22} & +a_{22} & \cdot & \cdot \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} b_1 \\ c_1 \\ b_2 \\ c_2 \\ \cdot \\ \cdot \end{bmatrix} . \qquad (3)$$

The use of equations (2) or (3) is convenient since we may use programmes already in existence and there is no need to use any complex arithmetic. However the order and band-width have been doubled so that the storage is four times as great as for a real matrix and the work about eight times as great. Compared with solving (1) with complex arithmetic we need twice the storage and the work is approximately doubled.

Similar considerations apply to the generalisations of the methods of the other chapters. We may replace the eigenvalue problem

$$(A + iB)(\underline{x} + i\,\underline{y}) = \lambda(\underline{x} + i\,\underline{y}) \qquad (4)$$

by the equation

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \lambda \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} . \qquad (5)$$

All eigenvalues will now be repeated since if $\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix}$ is an eigenvector, then so is $\begin{bmatrix} -\underline{y} \\ \underline{x} \end{bmatrix}$. It would seem to be advisable, in all cases, to use complex arithmetic.

CHAPTER 11

SOLUTION OF ELLIPTIC DIFFERENTIAL EQUATIONS BY FINITE DIFFERENCES

Methods of setting up finite-difference approximations to differential equations are well-known and we refer the reader to Fox (1962, Chapter 21), for example. It is probably true to say that a majority of those sets of linear equations which are solved by iterative methods arise from differential equations and we have already mentioned one or two examples. We will not attempt here any comprehensive account of possible difficulties, but consider the case where the boundary is curved in a problem of "Dirichlet" type and the case where the boundary contains one or more reentrant corners.

Curved Boundaries

We consider the differential equation

$$\nabla^2\phi = f(x,y), \tag{1}$$

with boundary values specified on a given closed curve. The standard procedure is to cover the area with a uniform square mesh with interval h and to represent the differential equation at an internal point by the set of equations

$$\frac{1}{h^2}\left\{4\,\phi(x,y) - S_1[\phi(x,y)]\right\} + f(x,y) = \frac{h^2}{4!}\,E_4(x,y,h), \tag{2}$$

where

$$S_1[\phi(x,y)] = \phi(x+h,y) + \phi(x-h,y) + \phi(x,y+h) + \phi(x,y-h), \tag{3}$$

$$E_4(x,y,h) = \frac{\partial^4\phi}{\partial x^4}(x+\theta_1 h,y) + \frac{\partial^4\phi}{\partial x^4}(x-\theta_2 h,y) + \frac{\partial^4\phi}{\partial y^4}(x,\,y+\theta_3 h) + \frac{\partial^4\phi}{\partial y^4}(x,\,y-\theta_4 h), \tag{4}$$

and $0 < \theta_1 < 1$. Sufficient conditions for the validity of (4) are that $\dfrac{\partial^4\phi}{\partial x^4}$ be continuous on the open line $(x \pm h, y)$, that

$\frac{\partial^4 \phi}{\partial y^4}$ be continuous on the open line $(x, y \pm h)$ and that ϕ be continuous on these lines as closed intervals. At a grid point near the boundary, the finite-difference form (3) may include one or more points that do not belong to the net. We overcome this difficulty by using the known values of ϕ on the boundary to give approximations for the values of ϕ at the required points outside the net. For example, at the point 0 of Figure 1 we require

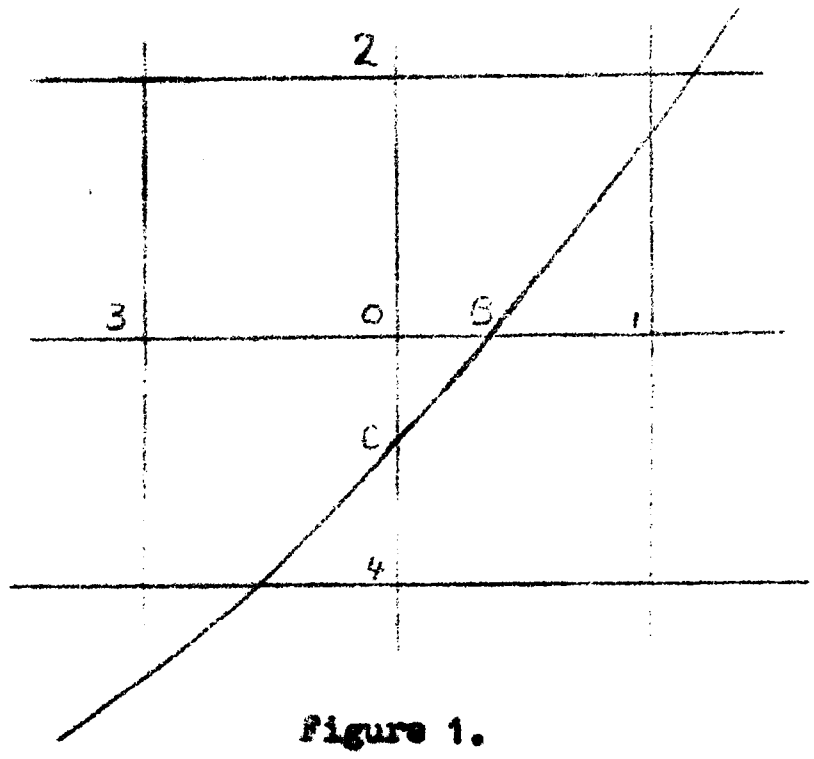


Figure 1.

approximations to the values of ϕ at points 1, 4 and these can be expressed as linear combinations of the values of ϕ at B, 0, 3 ... and C, 0, 2 ... respectively. If only two-point extrapolation is used then it is easily seen that when $\frac{h^2}{4!} E_4(x, y, h)$ is neglected the matrix of the resulting set of finite-difference equations is symmetric. It is diagonally dominant and hence positive-definite, and possesses property A. It is thus highly suitable for any of the iterative methods. If three points are used in the extrapolation, then the matrix is not symmetric although it does possess property A and is diagonally dominant. It follows that the real parts of all the eigenvalues are positive, so that SOR will converge with a sufficiently small relaxation factor. Unfortunately it is difficult to take advantage of the property A unless the Jacobi matrix has real roots. This we cannot guarantee, so that it is difficult to find an optimum relaxation factor. If we use

a direct method of solution, however, the band-width is no greater than when only two-point extrapolation is used. If four-point extrapolation is used then the band-width is increased and property A is lost, although we still have diagonal dominance.

Varga (1962) has suggested the use of a rectangular finite-difference grid, so chosen that any point where the boundary meets a grid line is also the meet of two grid lines. In general the mesh intervals will not be uniform. He considers the equation

$$-\frac{\partial}{\partial x}\left[P(x,y)\ \frac{\partial u}{\partial x}\right]-\frac{\partial}{\partial y}\left[P(x,y)\ \frac{\partial u}{\partial y}\right]+\sigma(x,y)u=f(x,y), \qquad (5)$$

with boundary condition

$$\alpha(x,y)u+\beta(x,y)\ \frac{\partial u}{\partial n}=\gamma(x,y), \qquad (6)$$

where $\frac{\partial u}{\partial n}$ is the derivative of u normal to the boundary and where $\sigma(x,y) > 0$ and $P(x,y) > 0$. He obtains a set of difference equations corresponding to a five-point approximation, the matrix of which is symmetric, diagonally dominant and possesses property A. However this result has been obtained at the expense of a local truncation error of size $O(h)$. Also it is not possible to set up this finite-difference grid for a general curved boundary and it will therefore be necessary to deform the boundary by a deformation of size $O(h)$. Varga's technique is thus clearly inferior for the differential equation (1), but for the more general equations (5) and (6) the straightforward finite-difference representation will not give a symmetric matrix. This is a disadvantage for both an iterative and direct method of solution of the algebraic equations.

The treatment of the boundary condition becomes more difficult if we use a more accurate finite-difference form. Consider, for example, the nine-point formula given by the equation

$$\frac{1}{6h^2}\,[20-4S_1-S_2][\phi(x,y)] + f(x,y) + \frac{h^2}{12}\,\nabla^2 f(x,y) = \frac{h^4}{6!}\,E_6(x,y,h) \qquad (7)$$

where $S_1\,[\phi(x,y)]$ is given by (3),

$$S_2\,[\phi(x,y)] = \phi(x+h,\ y+h) + \phi(x+h,\ y-h) + \phi(x-h,\ y+h) + \phi(x-h,\ y-h), \qquad (8)$$

and E_6 is a linear combination of sixth derivatives of ϕ at points inside the square with corners at $(x\pm h,\ y\pm h)$. Sufficient conditions for its validity are that the sixth derivatives of ϕ are continuous in the open square and that ϕ be continuous in the closed square. Since we have a local truncation error of order h^4 at each point at which (7) is applicable, we would be well advised to use four or more points in the extrapolation near the boundary. However there may not be as many as four points available in the direction required. This is likely to occur near a corner and we illustrate an example in Figure 2. For the finite-difference form at A we cannot do

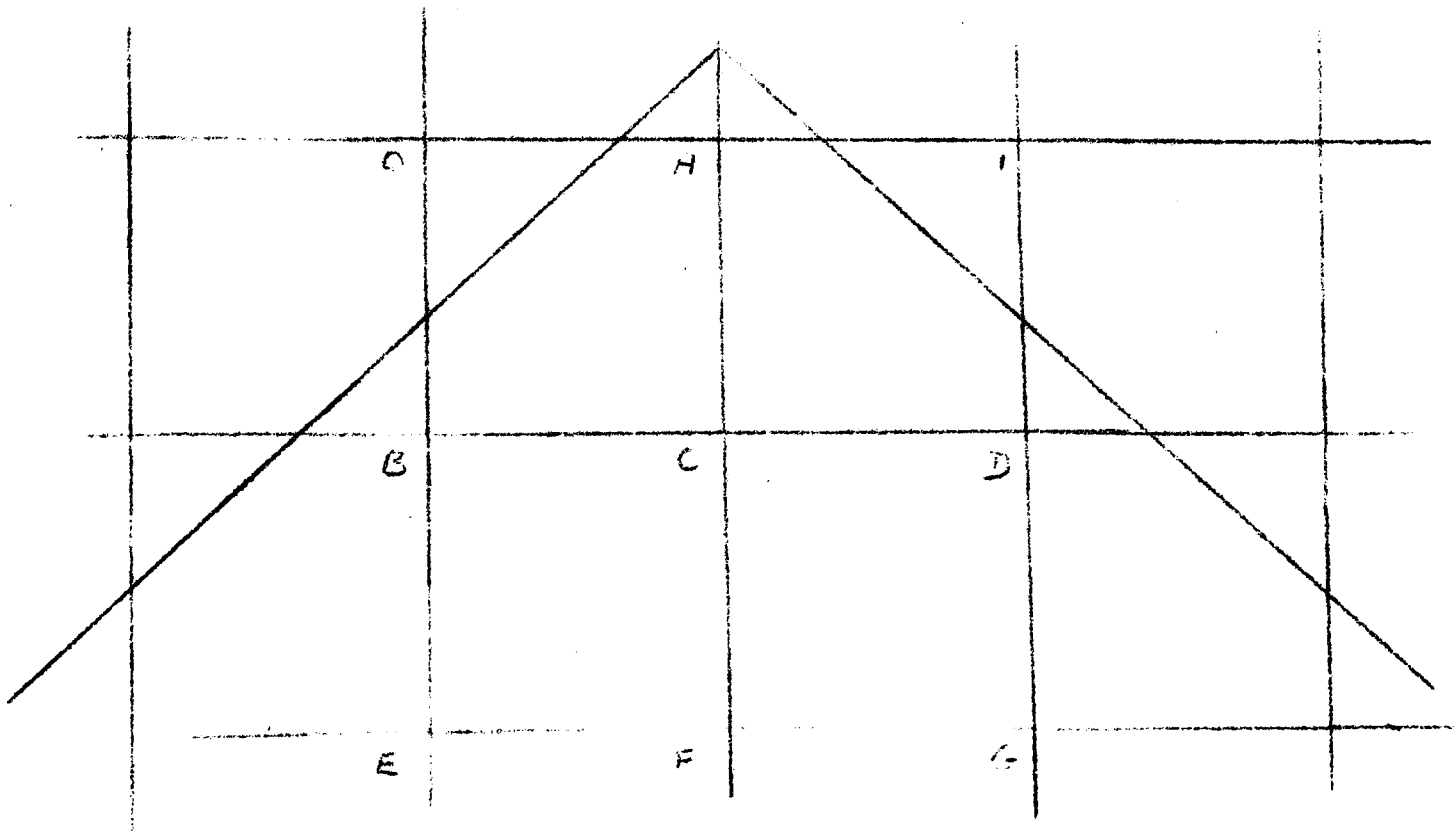


Figure 2

better than three-point extrapolation. One solution is to omit the value at A from the general equations and this procedure has much to recommend it. For a general area, provided a reasonably fine mesh is taken, it should be necessary to exclude only a few isolated points from the set. The resulting matrix will not be symmetric but it will

will be diagonally dominant provided each extrapolation is along the line of grid points which includes the point at which the differential equation is being represented. If we use a direct method the four-point extrapolation will yield a matrix whose band-width is probably about twice that for three-point extrapolation. Approximately four times as much work will be needed to obtain an error of order h^4 instead of h^3. It seems unlikely that so much extra work will be justified.

Another alternative is to replace extrapolation at the boundary by interpolation. At each point for which the finite-difference formula includes points outside the area of integration we replace this formula by one representing an interpolation along a convenient line of grid points. The matrix corresponding to this representation is unlikely to have diagonal dominance, but it does provide a convenient method of obtaining consistent orders of local truncation errors. Yet another possibility is to use the finite-difference form for those points that are at a distance greater than some specified limit, such as half the mesh length, from the boundary, and to use interpolation or extrapolation for all those other points which are involved in one or more of the finite-difference formulae. The limit on the distance of the "ordinary" points to the boundary is necessary so that we can state the order of local truncation error as a function of h.

Forsythe and Wasow (1960), Bramble and Hubbard (1962) and other authors have considered the problem of deducing the order of the error in the solution for various finite-difference representations. In general the results for particular cases suggest that for elliptic problems the order of truncation error in the solution is equal to the lesser of the orders of the truncation of the representations at the boundary and in the interior of the area of integration.

Richardson's deferred approach to the limit, described for

example by Fox (1962, page 108) is a powerful method for reducing the truncation error. However, as pointed out by Forsythe and Wasow (1960, page 307) this is valueless when the problem has a curved boundary unless the boundary condition is represented with a smaller truncation error than would otherwise be necessary. For example if the five-point representation (2) is in use for the equation (1) we must use four-point interpolation or extrapolation at the boundary if we wish the result of deferred correction to have error of order h^4. In general we must use at least as many points in the boundary extrapolation as the order we hope to have in the corrected result. This follows by an argument in exact correspondence with that of Fox (1962, page 106) for the case of the boundary value problem in an ordinary differential equation.

An Eigenvalue Problem for a Re-entrant Region

There has been some discussion about the so-called "L-membrane" problem. We consider it here since it illustrates the difficulties involved in a problem where the boundary has a re-entrant corner. The problem is given by the eigenvalue equation

$$\nabla^2 \phi + \lambda \phi = 0, \tag{9}$$

with $\phi = 0$ on the boundary, which is formed from three unit squares as shown in Figure 3. We seek the fundamental (smallest) eigenvalue

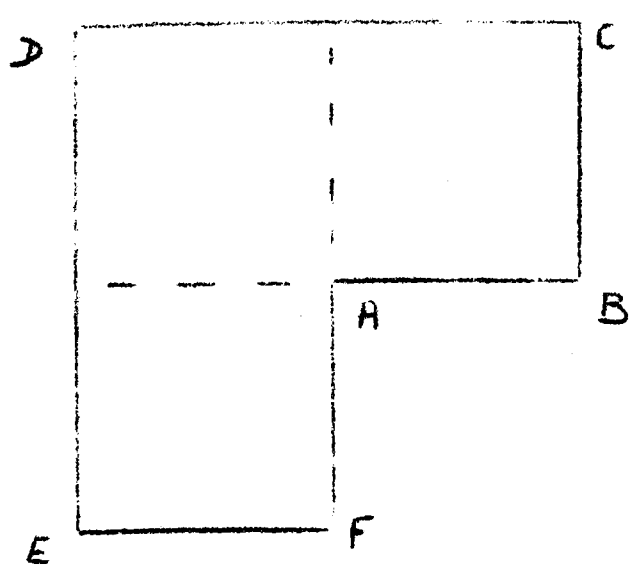


Figure 3

and corresponding eigenfunction, and also the second eigenvalue. The third presents no difficulty since the corresponding eigenfunction is an eigenfunction of the problem over the square of side 2.

The straightforward solution by finite differences involves the use of the five- or nine-point formulae corresponding to (2) and (7), but with $-\lambda\, \phi(x,y)$ and $\lambda^2\, \phi(x,y)$ replacing $f(x,y)$ and $\nabla^2 f(x,y)$ respectively. When the error term is neglected there results a matrix eigenvalue problem and we look for its two smallest latent roots. To appreciate the difficulties involved in these methods, we consider the form of the solution of the differential equation in the neighbourhood of a corner. Suppose the internal angle is π/m. We separate the variables in equation (9) expressed in polar

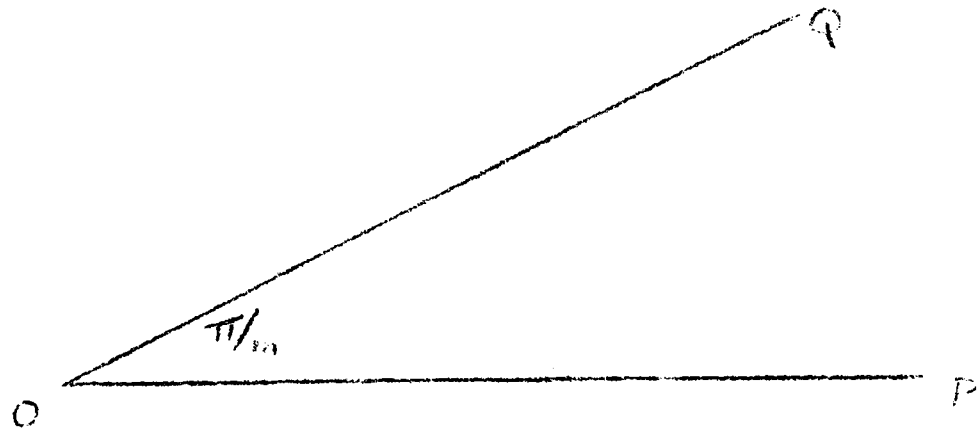


Figure 4

coordinates with O as origin and OP as initial line (see figure 4). To satisfy boundary conditions $\phi = 0$ on OP and OQ we find

$$\phi = \sum_{i=1}^{\infty} \alpha_i \sin mi\theta\, J_{mi}(r\sqrt{\lambda}), \tag{10}$$

and if we take the first term in the power expansion of the Bessel functions we have

$$\phi = a\, r^m \sin m\theta + O(r^{2m}). \tag{11}$$

If m is an integer this function is well-behaved in the sense that all its derivatives are continuous and bounded. However $m = \frac{2}{3}$

for the re-entrant corner of the L-membrane, so that unless $a = 0$ all the derivatives of ϕ in the r direction will be unbounded.

If we exclude from our finite-difference grid a small fixed area around the re-entrant corner, then the expression (4) will be bounded. Now suppose that this fixed area is sufficiently small for (11) to give an adequate representation of ϕ, so that

$$\nabla^2\phi = -\lambda\phi = O(r^{2/3}). \tag{12}$$

If we consider the grid point (h,h), then

$$(S_1 - 4)\ \phi(h,h) = \beta_{11}\, h^{2/3} + O(h^{4/3}), \tag{13}$$

where β_{11} is an absolute constant that may be found by direct computation. We deduce from (12) and (13) that the finite-difference form (2) will have truncation error $O(h^{2/3})$ at the grid point (h,h). Similarly for any grid point within the fixed area around the corner we have

$$(S_1 - 4)\ [\phi(ih,jh)] = \beta_{ij}\, h + O(h^{4/3}) \tag{14}$$

where the β_{ij} are absolute constants.

With truncation errors of this size we may expect the solution of the finite-difference equations to converge to the solution of the differential equation, although we know of no proof. Numerical experience indicates that the smallest eigenvalue of the matrix corresponding to the five-point formula behaves as

$$\lambda_5(h) = \lambda + a\, h^{4/3} + b\, h^2 + \dots, \tag{15}$$

where a and b are constants, and the eigenvector has the similar behaviour

$$\phi_5(h) = \phi + \phi_a\, h^{4/3} + \phi_b\, h^2 + \dots, \tag{16}$$

when ϕ_a and ϕ_b are constant functions. Clearly the convergence is likely to be poorer than for the same differential equation over an area whose corners are all convex.

We obtain rather better results for the second eigenvalue and corresponding eigenfunction. This is because the eigenfunction has a line of zeros along AD (in figure 3) and in the fundamental eigenfunction of the differential equation (9) with the boundary condition $\phi = 0$ on the lines AB, BC, CD, DA. This is an area with convex corners and we may apply the result of Forsythe and Wasow (1960) that the finite-difference equations (2) yield a lower bound for the eigenvalue for sufficiently small intervals. Experience indicates that this eigenvalue $\lambda_5^{(2)}(h)$ behaves as

$$\lambda_5^{(2)}(h) = \lambda + a\,h^2 + b\,h^{8/3} + \ldots \qquad (17)$$

where a and b are constants and a is negative.

We may apply similar considerations to the use of the more accurate nine-point formula (7). In the area near the corner we have, corresponding to (14), the equation

$$(4S_1 + S_2 - 20)\,[\phi(ih,jh)] = \beta'_{ij}\,h^{2/3} + O(h^{4/3}). \qquad (18)$$

Outside this area $E_6(x,y,h)$ will be bounded, so that the local truncation error will be smaller than that of the five-point formula. Results that have been obtained indicate that for the first and second eigenvalues we have

$$\lambda_9^{(1)}(h) = \lambda^{(1)} + a\,h^{4/3} + b\,h^{8/3} + \ldots , \qquad (19)$$

and

$$\lambda_9^{(2)}(h) = \lambda^{(2)} + a\,h^{8/3} + b\,h^4 + \ldots . \qquad (20)$$

We give in table 1 some results for various values of h.

We may be tempted to use deferred approach to the limit to improve on these results. If we fit the first two terms of (19) to the results at $h = \frac{1}{12}$ and $\frac{1}{16}$ we obtain $\lambda \simeq 9.6394$, a result that is in error in only the last decimal place. We feel, however, that it is dangerous to rely on such a result when we have no theoretical justification for (19).

		5 pt formula, $\lambda_5(h)$		9 pt formula, $\lambda_9(h)$	
h	No. of points	1st eigenvalue	2nd eigenvalue	1st eigenvalue	2nd eigenvalue
$\frac{1}{3}$	9	9.52514	13.73700	10.09792	15.32938
$\frac{1}{4}$	18	9.64143	14.37340	9.92500	15.22122
$\frac{1}{6}$	45	9.69083	14.83259	9.79776	15.19661
$\frac{1}{8}$	84	9.69316	14.99315	9.74593	15.19535
$\frac{1}{10}$	135	9.68829	15.06716	9.71816	15.19573
$\frac{1}{12}$	198	9.68273	15.10721	9.70106	15.19614
$\frac{1}{16}$	360	9.67351	15.14684	9.68141	15.19664

TABLE 1

These difficulties may be overcome by the use of the analytic form (10) in the neighbourhood of the corner and the finite-difference equations elsewhere. The method was first proposed by Motz (1946) and has been developed by Walsh (1960). She reports favourably on the method and obtains the result $\lambda = 9.639$ with the step $h = \frac{1}{12}$, a very great improvement on the corresponding result in Table 1. The method is easily extended to problems in higher dimensions or where there is more than one re-entrant corner, ~~but~~

but suffers the disadvantage that precise error bounds are hard to obtain.

We now consider an alternative for the L-membrane problem, namely the use of a conformal transformation. The transformation

$$w = z^{2/3}, \tag{21}$$

taken with origin at A in Figure 3, gives rise to a new area, sketched in figure 5, all of whose corners have internal angle $\frac{\pi}{2}$.

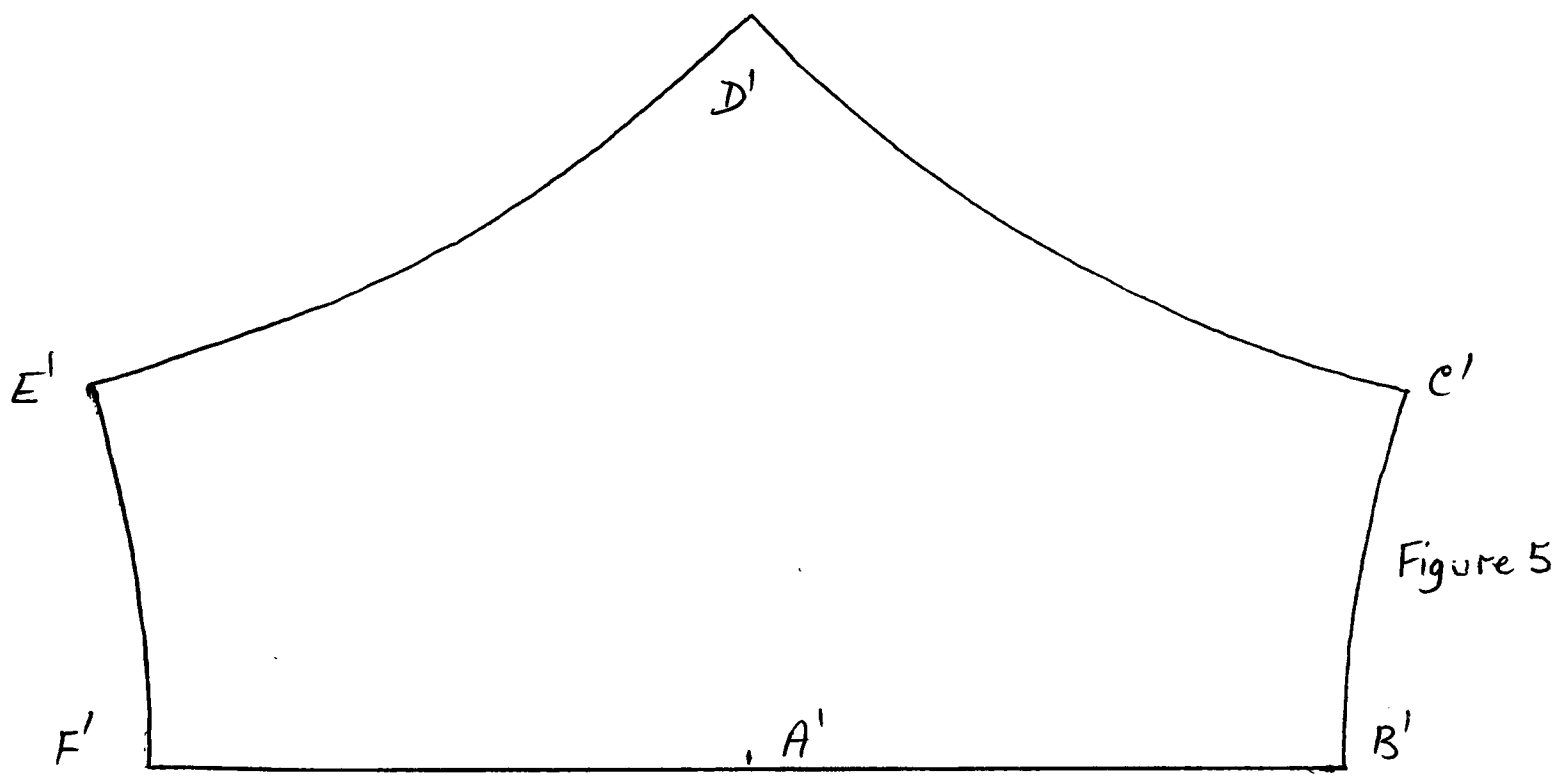


Figure 5

The differential equation (9) becomes

$$\nabla^2\phi + \lambda\rho\phi = 0, \qquad = \tfrac{2}{4}|\omega|, \tag{22}$$

with boundary condition $\phi = 0$. Since all the corners have internal angle $\frac{\pi}{2}$ equation (11) shows that all the derivatives of ϕ will be bounded. We can therefore expect good results with the use of finite differences over the whole area. There is now the complication of a curved boundary, so that this problem presents an illustration of the techniques discussed in the earlier part of this chapter.

We show in table 2 the results given by the five-point form with two-, three- and four-point extrapolation at the boundary. We used extrapolation (or interpolation) for any point that was not at least half the mesh interval away from the boundary in both the x and y coordinate directions. The mesh width h is related to

the parameter n by the formula

$$h = \frac{1}{2^{2/3} n} . \tag{23}$$

With each result is shown, in brackets, the result of deferred correction with the next. We also show the number of points in the grid in order to facilitate a comparison of the amount of work involved in this method with that for the untransformed grid.

N	6	8	10	12
Number of points	77	140	219	319
2pt boundary	9.5186 (9.6507)	9.5764 (9.6442)	9.6008 (9.6538)	9.6170
3pt boundary	9.5109 (9.6588)	9.5756 (9.6320)	9.5959 (9.6471)	9.6103
4pt boundary	9.5287 (9.6453)	9.5797 (9.6386)	9.6009 (9.6402)	9.6129

TABLE 2 $(\lambda_5(h))$

It can be seen quite clearly that the best corrected results are given with four-point extrapolation at the boundary. We know from the theory that the corrected value has error of size $O(h^4)$.

The nine-point finite-difference form must be modified for this problem. We have

$$\begin{aligned} &4S_1(\phi) + S_2(\phi) - \frac{h^6}{6!} E_6(x,y,h) \\ &= [20 + 6h^2 \nabla^2 + \tfrac{1}{2}h^4 \nabla^4]\phi \\ &= [20 - 6h^2 \lambda \rho]\phi - \tfrac{1}{2}h^4 \lambda \nabla^2(\rho\phi) \\ &= [20 - 6h^2 \rho\lambda]\phi - \tfrac{1}{2}h^2\lambda [S_1(\rho\phi) - 4\rho\phi] + O(h^6). \end{aligned}$$

Rearranging, we have

$$4S_1(\phi) + S_2(\phi) + \tfrac{1}{2} h^2\lambda\, S_1(\rho\phi) = [20 - 4h^2\lambda\rho]\phi + O(h^6). \qquad (24)$$

The results of the use of formula (24) are shown in table 3.

N	6	8	10	12
λ (h)	9.6383	9.6393	9.6396	9.6397

TABLE 3

From the approximate eigenfunction we may obtain an improved estimate of the eigenvalue by calculating the Rayleigh quotient of the matrix corresponding to a more accurate finite-difference representation. We calculate $\nabla^2\phi$ at each point by straightforward finite differences in the directions of the grid lines. The contribution to $\nabla^2\phi$ of the eighth differences for the finest grid was nowhere greater than 10^{-6} times the maximum value of $\nabla^2\phi$ and at most points was far smaller than this. Hence we considered that the use of all differences up to the eighth was sufficient. For comparison similar calculations were performed on the grids given by N = 8, 10 and the results are shown in Table 4. Unfortunately

N	8	10	12
λ	9.63969	9.63972	9.63972

TABLE 4

the matrix is not symmetric on account of the curved boundary and so we cannot apply the result of Wilkinson (1961) to obtain an error bound, unless we are prepared to find approximations for all the eigenvectors.

We should be able to find an even better approximation to the eigenvalue by calculating the Rayleigh quotient of the differential equation,

$$\lambda_R = \frac{\iint \psi \nabla^2 \psi \, dxdy}{\iint \rho \psi^2 \, dxdy} \quad . \tag{25}$$

We require that the approximate eigenfunction ψ should satisfy the boundary condition, should be continuous and should have its first and second derivatives continuous. Under these conditions the result of Courant and Hilbert (1953, Vol. 1, page 369) shows that ψ may be expanded in terms of the eigenfunctions, ϕ_i, in an absolutely and uniformly convergent series,

$$\psi = \sum_{i=1}^{\infty} a_i \phi_i \, , \tag{26}$$

where the eigenfunctions are ordered so that the corresponding eigenvalues λ_i are in ascending order of modulus. $\nabla^2 \psi$ is given by the expression

$$\nabla^2 \psi = \sum_{i=1}^{\infty} a_i \lambda_i \rho \phi_i \, . \tag{27}$$

By substituting these expansions in (25) and using the orthogonality of the functions ϕ_i we find

$$\lambda_R = \frac{\sum_{i=1}^{\infty} \lambda_i a_i^2}{\sum_{i=1}^{\infty} a_i^2} \quad . \tag{28}$$

If we form the residual function

$$\theta = \nabla^2 \psi - \lambda_R \rho \psi = \sum_{i=1}^{\infty} a_i (\lambda_i - \lambda_R) \rho \phi_i \, , \tag{29}$$

and hence evaluate the constant

$$\epsilon^2 = \iint \rho^{-1} \theta^2 \, dxdy = \sum_{i=1}^{\infty} a_i^2 (\lambda_i - \lambda_R)^2, \tag{30}$$

and if

$$|\lambda_i - \lambda_R| \geq a \tag{31}$$

for $i = 2, 3 \ldots$, then we may obtain the bound

$$|\lambda_1 - \lambda_R| \leq \frac{\epsilon^2/a}{b - \epsilon^2/a^2} \tag{32}$$

for the error in the Rayleigh quotient, where $b = \iint \rho \psi^2 \, dxdy$. To prove this result we use (30) and (31) to show that

$$\epsilon^2 \geq \sum_{i=2}^{\infty} a_i^2 (\lambda_i - \lambda_R)^2 \geq a^2 \sum_{i=2}^{\infty} a_i^2, \tag{33}$$

and that

$$\frac{\epsilon^2}{a} \geq \frac{1}{a} \sum_{i=2}^{\infty} a_i^2 (\lambda_i - \lambda_R)^2 \geq \sum_{i=2}^{\infty} |\lambda_i - \lambda_R| \, a_i^2 . \tag{34}$$

From (28) we find

$$\lambda_R \sum_{i=1}^{\infty} a_i^2 = \sum_{i=1}^{\infty} \lambda_i a_i^2 , \tag{35}$$

which may be rearranged as

$$(\lambda_R - \lambda_1) a_1^2 = \sum_{i=2}^{\infty} (\lambda_i - \lambda_R) a_i^2 , \tag{36}$$

and it follows that

$$|\lambda_R - \lambda_1|\, a_1^2 \leq \sum_{i=2}^{\infty} |\lambda_i - \lambda_R|\, a_i^2$$

$$\leq \varepsilon^2/a \,. \tag{37}$$

Now from (33) we have the relation

$$a_1^2 = \sum_{i=1}^{\infty} a_i^2 - \sum_{i=2}^{\infty} a_i^2 \geq b - \varepsilon^2/a^2 \,, \tag{38}$$

and the result (32) follows at once from (37) and (38).

If we can evaluate the relevant integrals we obtain a result with a small error and also a rigorous bound for this error. Now we have ψ given at only a finite number of isolated points and we need a rule to extend it to a continuous function with continuous first and second derivatives. One such rule is to use four-point interpolation in both the x and y directions. To find ψ at the point X in figure 6, for example, we use the values at the points A, B ... P. Near the edge we may first have to calculate some of

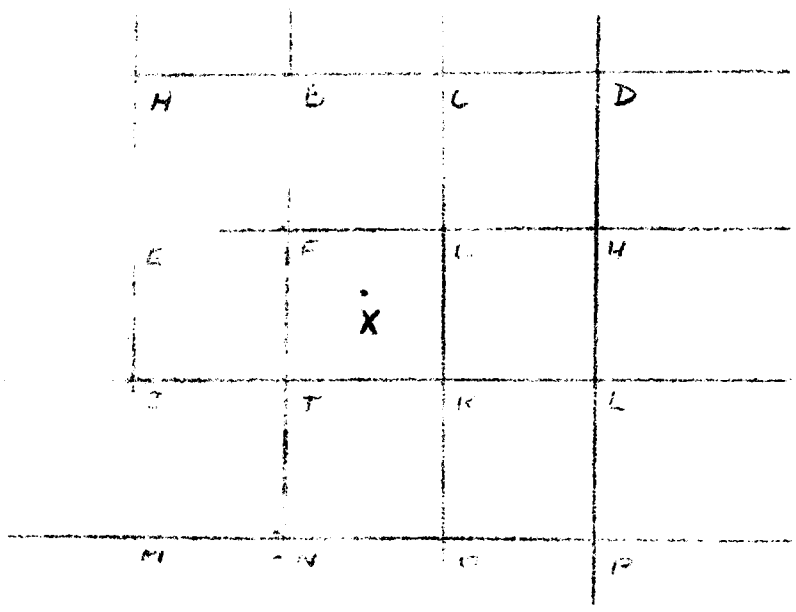


Figure 6

the values of ψ at the points A ... P by extrapolation, or possibly by taking advantage of a symmetry in the eigenfunction. The resulting function ψ will be continuous and have continuous first and second derivatives, but in general will not satisfy the boundary condition exactly. It will be zero on an approximation

to the boundary. With care, however, we should be able to arrange that this approximate boundary is either wholly inside or wholly outside the required boundary so that we may obtain upper and lower bounds for the required eigenvalue. For greater accuracy we may use a larger number of points, such as 36, 64 and so on. Over each square of the grid ψ is a polynomial and can in principle be integrated exactly. In practice it may be simpler to use quadrature over each square. Unfortunately we did not have enough time to perform this calculation for the L-membrane.

It has previously been suggested that the Rayleigh quotient (25) should be calculated using numerical differentiation and integration over the grid points only and taking into account all the significant differences. This will undoubtedly involve very much less work than the method suggested here but we would be surprised if the result were as accurate and when error bounds are not available.

For the second eigenvalue of the L-membrane we show in Table 5 results obtained from the nine-point formula with six-point extrapolation at the boundary, and also the Rayleigh quotient of the more accurate representation using up to the eighth differences.

N	6	8	12
$\lambda_9(h)$	15.205	15.199	15.198
Rayleigh Quotient		15.1975	15.1973

TABLE 5

If we have a region with more than one re-entrant corner, we may use the transformation

$$w = (z-z_1)^{m_1} + (z-z_2)^{m_2} + \ldots (z-z_n)^{m_n} , \qquad (39)$$

where the region has corners of internal angle π/m_i at the points z_i. There seems to be no reason why this should work less satisfactorily than that for the L-membrane, although the boundary in the transformed plane will be more complicated and the programme correspondingly more involved. Unfortunately we do not know of a corresponding transformation for the three dimensional problem.

Each of the Bessel functions appearing in (10) may be expanded as a power series in r^m and this at once suggests the alternative transformation

$$\bar{r} = r^{2/3} , \quad \bar{\theta} = \theta \qquad (40)$$

for the L-membrane, where polar coordinates in the untransformed plane are taken with origin at A with AB as initial line. It is possible that this may have a generalization in three dimensions. In the transformed plane the differential equation in polar coordinates is

$$\frac{4}{9} \left[\frac{\partial^2 \phi}{\partial \bar{r}^2} + \frac{1}{\bar{r}^2} \frac{\partial \phi}{\partial \bar{r}} \right] + \frac{1}{\bar{r}^3} \frac{\partial^2 \phi}{\partial \bar{\theta}^2} + \lambda \phi = 0. \qquad (41)$$

We decided that it would be best to solve this equation by finite differences, rather than the corresponding equation in Cartesian coordinates, since the latter is very complicated. The mesh steps in each direction were related to the integer n by the formulae

$$h_\theta = \frac{3\pi}{4n} , \quad h_r = \frac{1}{n} , \qquad (42)$$

and the number of points in each grid was slightly more than n^2. We show in Table 6 the result of replacing each term in (41) by its simplest central difference approximation and taking four-point extrapolation at the boundary. Improved results may be obtained by

N	8	10	12	14	16	18	20
λ	9.400	9.501	9.560	9.567	9.591	9.602	9.611

TABLE 6

h^2 extrapolation. For example, we find $\lambda = 9.646$ by extrapolation between the values given at $n = 16, 20$. We have also obtained results on the same grid using the first two terms in the central difference series corresponding to the terms in (41) and with six-point extrapolation at the boundary. The results are shown in Table 7. It should be noted, however, that solutions of accuracy

N	8	16	32
λ	9.6686	9.6408	9.6398

TABLE 7

comparable with those of the conformal transformations are obtained at the expense of grids with a far greater number of mesh points.

Federenko's Iteration

We conclude this chapter by considering a special relaxation method that has been proposed by Federenko (1961). He considers the usual five-point approximation to Poisson's equation over the rectangle and uses Seidel iteration to solve the resulting matrix equation. He observes that the dominant eigenvectors of the iteration matrix are likely to be smooth mesh functions and can therefore be represented with reasonable accuracy on a coarser grid. He therefore proposes that the iteration should be continued until the residual

is reasonably smooth and then the set of difference equations corresponding to the coarse grid be solved with these residuals as right-hand sides. The solution of this set is then extended to the original grid to provide a correction to the solution. The Seidel iteration is now restarted and the process continued, using alternately the fine and coarse grids until the required accuracy has been obtained. The same idea was used by Southwell and others to improve the convergence of relaxation methods for desk machines, but to our knowledge Federenko is the only author to propose its use for automatic machines. He tried a test case, using a 50 x 50 net, and found the method needed about a third of the work of SOR with optimum relaxation parameter.

There seems no reason for restricting ourselves to the Gauss-Seidel iteration. The idea would appear to be equally applicable for SOR iteration. We can examine analytically the case of the Poisson equation over the unit square. Suppose the step is $h = \frac{1}{n}$ where n is an integer, then the Jacobi matrix has eigenvalues

$$\lambda_{p,q} = \tfrac{1}{2}(\cos p\pi h + \cos q\pi h) \tag{43}$$

for $p = 1, 2 \ldots (n-1)$ and $q = 1, 2 \ldots (n-1)$.

The greatest of these is

$$\lambda_{1,1} = \cos \pi h. \tag{44}$$

If straightforward SOR is used, then the optimum relaxation factor is

$$w_{1,1} = \frac{2}{1 + \sqrt{1-\cos^2 \pi h}} = \frac{2}{1 + \sin \pi h}, \tag{45}$$

and the asymptotic convergence rate is

$$R_{1,1} = -\log (w_{1,1} - 1) = \pi h + O(h^2). \tag{46}$$

Suppose, however, that we choose the relaxation parameter so that all but the dominant eigenvector of the iteration matrix is eliminated as rapidly as possible and that Fedorenko's process is used to deal with this. The second eigenvalue of the Jacobi iteration matrix is

$$\lambda_{1,2} = 1 - \frac{5}{4}\pi^2 h^2 + O(h^4), \tag{47}$$

the corresponding optimum relaxation parameter is

$$w_{1,2} = 2(1 - \pi h \sqrt{2.5}) + O(h^2), \tag{48}$$

and the asymptotic convergence rate is

$$R_{1,2} = \sqrt{10}\,\pi h + O(h^2) \simeq 3.16\,\pi h + O(h^2). \tag{49}$$

This is a very considerable gain, by a factor of 1.58, in convergence rate. We have not allowed for the extra time taken in the coarse net and in interpolating back to the fine net, which will take about as long as one to two iterations on the fine net and is therefore not very significant. We can expect even better results if more of the eigenvectors of the iteration matrix are reduced on the coarse grid. The greatest disadvantage would appear to be that it is difficult to choose the relaxation parameter w and the size of the coarse grid in a practical case. Federenko used a coarse step of five times the fine step for his test case, but gives no reason for this choice.

As an example, consider Poisson's equation over the square with boundary values unity. We take fine and coarse steps of $\frac{1}{20}$ and $\frac{1}{5}$ respectively and take the null vector as starting vector. The optimum relaxation factor for the fine grid is

$$w_{1,1} = \frac{2}{1 + \sin \frac{\pi}{20}} = \frac{2}{1.1564} = 1.73 \, . \tag{50}$$

The eigenvector whose value at the grid point (ih, jh) is

$$\phi_{1,5}(i,j) = \sin \frac{i\pi}{n} \sin \frac{5j\pi}{n} \tag{51}$$

is an eigenvector of the matrix that is too oscillatory to be represented well on the coarse grid and corresponds to the smallest latent root of the matrix of all such eigenvectors. We might expect therefore that a good choice of relaxation parameter would be

$$w_{1,5} = \frac{2}{1 + \left(1 - \lambda_{1,5}^2\right)^{\frac{1}{2}}} = \frac{2}{1 + \left(1 - .8474^2\right)^{\frac{1}{2}}}$$

$$= \frac{2}{1.531} = 1.31, \tag{52}$$

and the results displayed in Table 8 bear this out. We show the result of 30 iterations with w = 1.73 and then for various values of w we alternate between 10 fine iterations and a correction on the coarse grid, which we assume takes the same time as two fine iterations. We have estimated the convergence by showing the sum of the squares of the changes in each iteration. Since it appeared that convergence was slow towards the end of the sets of ten iterations we also tried taking w = 1.3 and using cycles of only six iterations, but it would appear that the slight improvement in convergence does not compensate for the extra time required for the coarse grid. Table 8 shows clearly the gain in this instance.

1.73	1.0	1.2	1.3	1.4	1.5	1.3
2.65, 1	9.71, 0	1.16, 1	1.29, 1	1.46, 1	1.93, 1	1.29, 1
5.42, 0	2.78, 0	2.69, 0	2.78, 0	3.01, 0	4.04, 0	2.78, 0
3.08, 0	1.36, 0	1.24, 0	1.24, 0	1.32, 0	1.93, 0	1.24, 0
1.84, 0	8.35, -1	7.44, -1	7.32, -1	7.54, -1	1.07, 0	7.32, -1
1.15, 0	5.78, -1	5.09, -1	4.96, -1	5.02, -1	6.63, -1	4.96, -1
7.50, -1	4.30, -1	3.77, -1	3.65, -1	3.67, -1	4.53, -1	3.65, -1
5.10, -1	3.36, -1	2.99, -1	2.84, -1	2.84, -1	3.32, -1	
3.58, -1	2.72, -1	2.37, -1	2.30, -1	2.29, -1	2.54, -1	
2.59, -1	2.26, -1	1.97, -1	1.91, -1	1.90, -1	2.01, -1	1.96, -1
1.92, -1	1.91, -1	1.68, -1	1.63, -1	1.61, -1	1.62, -1	3.73, -2
1.44, -1						1.20, -2
1.10, -1						5.02, -3
8.39, -2	1.21, -1	6.72, -2	5.13, -2	4.07, -2	3.35, -2	2.93, -3
6.46, -2	2.56, -2	1.26, -2	1.04, -2	9.70, -3	1.39, -2	1.96, -3
4.98, -2	8.73, -3	3.19, -3	2.42, -3	2.52, -3	8.38, -3	
3.85, -2	3.41, -3	9.29, -4	6.39, -4	8.11, -4	4.94, -3	
2.97, -2	1.52, -3	4.10, -4	2.80, -4	3.88, -4	3.06, -3	8.63, -4
2.28, -2	8.04, -4	2.45, -4	1.62, -4	2.39, -4	2.07, -3	2.60, -4
1.74, -2	4.97, -4	1.73, -4	1.10, -4	1.68, -4	1.44, -3	1.15, -4
1.30, -2	3.47, -4	1.33, -4	8.10, -5	1.23, -4	1.01, -3	6.91, -5
9.31, -3	2.63, -4	1.07, -4	6.24, -5	9.25, -5	7.27, -4	5.15, -5
6.54, -3	2.10, -4	9.00, -5	4.96, -5	7.08, -5	5.28, -4	4.24, -5
4.49, -3						
3.00, -3						
1.95, -3	1.27, -4	3.63, -5	1.84, -5	2.35, -5	2.63, -4	1.42, -5
1.24, -3	3.61, -5	7.53, -6	4.05, -6	5.85, -6	1.25, -4	2.91, -6
7.66, -4	1.52, -5	2.02, -6	9.34, -7	1.52, -6	6.74, -5	8.50, -7
4.61, -4	7.20, -6	5.67, -7	1.77, -7	4.42, -7	4.24, -5	3.47, -7
2.72, -4	3.95, -6	2.52, -7	6.27, -8	2.21, -7	2.97, -5	1.82, -7
1.58, -4	2.52, -6	1.60, -7	3.15, -8	1.42, -7	2.13, -5	1.12, -7

TABLE 8

The same kind of improvement can be expected if a similar procedure is applied to a gradient method. For Chebyshev semi-iteration applied to this example we may replace the range (.0123, 1.9877) by the range (.1526, 1.9877), and if we neglect the time taken in the coarse grid the asymptotic convergence rate is improved by a factor of about four.

CHAPTER 12

OTHER PROBLEMS THAT GIVE RISE TO LARGE SPARSE MATRICES

Problems in Surveying

Surveyors frequently find the need to find a least squares solution of the matrix equation

$$A\,\underline{x} \simeq \underline{b} \qquad (1)$$

when A is a large, sparse, rectangular matrix. The set of equations actually solved is given by the matrix equation

$$A^T A\,\underline{x} = A^T\,\underline{b}\,. \qquad (2)$$

The matrix $A^T A$ is, of course, symmetric and positive-definite. We need store only the upper triangular part and we have a wide choice of iterative methods open to us. The structure of the matrix $A^T A$ in an example where A is a 244 x 84 matrix is shown in Figure 1.

Figure 1.

Each x represents a non-zero 2 x 2 matrix and only the upp r

triangular part is illustrated. Note the large number of zeros inside the band which means that much more storage will be needed if the matrix is treated as a band matrix than if it is solved by an iterative method. Our experience indicates that the major difficulties with these problems lie in the setting up of the least-squares system (2), rather than in their solution. The least-squares matrix must be stored in a condensed form and that chosen for our Mercury programme consisted in storing it by rows with indices to give the number of non-zero elements in each row and their positions in the row. An estimate of the maximum number of non-zero elements in any row is needed to fix the addresses of the starting points of the rows and a procedure is needed in case the estimated maximum number of elements in a row is exceeded.

A large case, with normal matrix of order 234, was tried on the Mercury computer and SOR was used to solve the set of equations. A guessed relaxation factor of 1.4 was used. The modulus of dominant eigenvalue of the iteration matrix was 0.80 and the solution of the normal equations took a fraction of the time taken to form them. The unaccelerated Gauss-Seidel iteration was also quite rapidly convergent, the modulus of the dominant eigenvalue of the iteration matrix being 0.88. Some experiments with other relaxation parameters and some comparisons with other iterative methods would be interesting. The method of conjugate gradients is particularly attractive since we do not need to guess a parameter.

A Problem in Statistics

We consider the problem known as linear analysis of variance. We wish to analyse an experiment whose result depends on a number of parameters i, j, k ... t and we attempt to take readings for all possible combinations of the various values that the parameters can take or are restricted to take. We attempt to fit the results to the equation

$$\phi_{i,j,k\ldots t} \simeq \quad + A_i + B_j + \ldots X_t + (AB)_{ij} \ldots + \ldots (XY)_{st} \qquad (3)$$

where each term represents an array of numbers. The term μ will always be present, but any other term may be absent, although it is assumed that if any interaction $(DE)_{kl}$ is present then so are the linear terms D_k, E_l corresponding to its component parts. The fit is essentially unchanged if a constant is added to every component of A_i and subtracted from μ. To resolve this and other similar ambiguities we add the conditions

$$\sum_i A_i = 0, \quad \sum_j B_j = 0, \ldots \quad \sum_t X_t = 0,$$

$$\sum_i (AB)_{ij} = \sum_j (AB)_{ij} = 0. \quad \text{etc.} \qquad (4)$$

We use these conditions to reduce the number of unknowns in (3) to a minimum. If we write these unknowns as the vector $\underline{x}$, then equation (3) can be written in the matrix form

$$\underline{\phi} \simeq A\,\underline{x} \qquad (5)$$

where $\underline{\phi}$ is the vector of all the readings $\phi_{ij,k\ldots t}$. We wish to solve the least-squares set of equations

$$A^T A\,\underline{x} = A^T \underline{\phi}\,. \qquad (6)$$

If an equal number of readings $\phi_{ij,k\ldots t}$ is taken for each combination of values of i, j, k ... t, then the matrix $A^T A$ will be block-diagonal, where each block corresponds to a term in (3). The solution of this system is trivial. However if a few readings are missing there will be some non-zero elements outside these blocks. The order of the matrix can be very large and an iterative method of solution is indicated.

A programme for Mercury has been written, but we do not know of the result of any large scale experiment. The form of the matrix suggests the use of block iteration. Property A is lacking, in general, so we must rely on experiment to find the best SOR parameter. As with the surveyor's problem, it seems likely that the major part of the work will be in forming the normal equations, rather than in solving them.

B I B L I O G R A P H Y

Bramble, J.H. and Hubbard, B.E. (1962). On the Formulation of Finite Difference Analogues of the Dirichlet Problem for Poisson's Equation. Num. Math. 4, 313.

Broyden, C.G. (1964). On Convergence Criteria for the Method of Successive Over-relaxation. Math. Comp. 18, 136.

Carré, B.A. (1961). The Determination of the Optimum Accelerating Factor for Successive Over-relaxation. Comp. J. 4, 73.

Courant, R. and Hilbert, D. (1953). Methods of Mathematical Physics. Interscience, New York.

Darwent, T.J. (1963). Numerical Solution of Elliptic Partial Differential Equations for Neutron Flux in a Reactor. M.Sc. Thesis, London.

Engeli, M, Ginsburg, Th., Rutishauser, H. and Stiefel, E. (1959). Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-adjoint Boundary Value Problems. Birkhauser, Basle.

Evans, D.J. and Forrington, C.V.D. (1963). An Iterative Process for Optimizing Symmetric Successive Over-relaxation. Comp. J. 6, 271.

Federenko, R.P. (1961). A Relaxation Method for Solving Elliptic Difference Equations. Zh. vych. mat. 1: 5, 922. Translated into English: U.S.S.R. Comp. Math. and Math. Phys. (Pergamon) 4, 1092.

Forsythe, G.E. and Wasow, W.R. (1960). Finite Difference Methods for Partial Differential Equations. J. Wiley, New York.

Fox, L. et al. (1962). Numerical Solution of Ordinary and Partial Differential Equations. Pergamon.

Fox, L. (1964). An Introduction to Numerical Linear Algebra. Univ. Press, Oxford.

Ginsburg, Th. (1959). See Engeli, M.

Givens, W. (1959). The Linear Equations Problem. Technical Report No. 3, Appl. Math. and Stat. Ser. 225 (37). Stanford University.

Golub, G.H. and Varga, R.S. (1961). Chebyshev Semi-iterative Methods, Successive Over-relaxation Iterative Methods, and Second Order Richardson Iterative Methods. Num. Math. 3, 147.

Habetler, G.J. and Wachpress, E.L. (1961). Symmetric Successive Over-relaxation in Solving Diffusion Difference Equations. Math. Comp. 15, 356.

Householder, A.S. (1958). Unitary Triangularization of a Non-symmetric Matrix. J. ACM. 5, 339.

Kaczmarz (1937). S. Bull. Intern. Acad. Polonaise Sciences, A, 355.

Kahan, W. (1958). Gauss-Seidel Methods of Solving Large Systems of Linear Equations. Doctorial Thesis. University of Toronto.

Khabaza, I.M. (1963). An Iterative Least-square Method Suitable for Solving Large Sparse Matrices. Comp. J. 6, 202.

Kron, G. (1963). Diakoptics. MacDonald, London.

Kulsrud, H.E. (1961). A Practical Technique for the Determination of the Optimum Relaxation Factor of the Successive Over-relaxation Method. Comm. ACM. 4, 184.

Lanczos, C. (1950). An Iterative Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. J. Research N.B.S. 45, 255.

Motz, H. (1946). The Treatment of Singularities of Partial Differential Equations by Relaxation Methods. Q. Appl. Math. 4, 371.

N.P.L. (1961). Modern Computing Methods. Notes on Appl. Sc. 16. London, H.M. Stationery Office.

Ostrowski, A.M. (1957-1959). On the Convergence of the Rayleigh Quotient Iteration for the Computation of Characteristic Roots and Vectors. Archive for Rational Mechanics and Analysis. 1, 233; 2, 423; 3 325; 3, 341; 3, 472; 4, 153.

Rutishauser, H. (1958). Solution of Eigenvalue Problems with the L-R Transformation. N.B.S. Appl. Math. Series 49, 47.

Rutishauser, H. (1959). See Engeli, M.

Steifel, E.L. (1958). Kernel Polynomials in Linear Algebra and their Numerical Applications. N.B.S. Appl. Math. Series 49, 1.

Stein, P. (1951). The Convergence of Seidel Iterants of Nearly Symmetric Matrices. Math. Tab. and other Aids to Comp. 5, 237.

Tee, G.J. (1963). Eigenvectors of the Successive Over-relaxation Process and its Combination with Chebyshev Semi-iteration. Comp. J. 6, 250.

Varga, R.S. (1962). Matrix Iterative Analysis. Prentice-Hall, London.

Walsh, J.E. (1960). Numerical Solution of Partial Differential Equations Using a High-speed Computer. D.Phil. Thesis, Oxford.

Wilkinson, J.H. (1961). Error Analysis of Direct Methods of Matrix Inversion. J. ACM. 8, 231.

Wilkinson, J.H. (1961). Rigorous Error Bounds for Computed Eigensystems. Comp. J. 4, 230.

Wrigley, H.E. (1963). Accelerating the Jacobi Method for Solving Simultaneous Equations by Chebyshev Extrapolation when the Eigenvalues of the Iteration Matrix are Complex. Comp. J. 6, 169.

Young, D. (1954). Iterative Methods for Solving Partial Difference Equations of Elliptic Type. Trans. Amer. Math. Soc. 76, 92.

Young, D. (1962). The Numerical Solution of Elliptic and Parabolic Partial Differential Equations. Chap. 11 of "A Survey of Numerical Analysis", ed. J. Todd. McGraw-Hill.