THE CALCULATION OF THE EIGENVICTORS OF CODIAGONAL MATRICES PRODUCED BY THE GIVENS AND LANCZOS PROCESSES

by

J.H. Wilkinson

National Physical Laboratory Teddington, Middlesex

1. INTRODUCTION

In both the Givens and the Lanczos methods for calculating the eigenvalues and eigenvectors of a matrix, a collineatory transformation is constructed which reduces the matrix to codiagonal form. Givens[1] has given a complete analysis of the problem of finding the eigenvalues and has described a very satisfactory practical procedure for evaluating them. No such analysis has been given for the eigenvectors, though Givens in an unpublished paper has described a procedure which, in his experience, has given accurate results. In this note an analysis of the problem is given and a method is described which has been used extensively for calculating the vectors on DEUCE.

2. STATEMENT OF THE PROBLEM

The codiagonal forms produced by the Givens and Lanczos processes will be denoted by C_1 and C_2 respectively where we have

$$C_1 = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \beta_4 & & \\ & & \ldots & \ldots & & \\ & & & \ldots & \ldots & \\ & & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & & \beta_n & \alpha_n \end{bmatrix} \qquad (1)$$

and

$$C_2 = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ 1 & \alpha_2 & \beta_3 & & & \\ & 1 & \alpha_3 & \beta_4 & & \\ & & \ldots & \ldots & & \\ & & & \ldots & \ldots & \\ & & & 1 & \alpha_{n-1} & \beta_n \\ & & & & 1 & \alpha_n \end{bmatrix} \qquad (2)$$

The Givens method is appropriate only for symmetric matrices but the Lanczos method may be used for symmetric or unsymmetric matrices. When the matrix is symmetric the Lanczos process is always framed in a form in which it produces positive values of β_i. On an automatic computer it is usual to programme the Lanczos process so as to produce values of α and β in floating form but the Givens process is quite easily programmed using a fixed binary point.

We shall assume that some or all of the eigenvalues of the codiagonal form have been calculated to a high degree of accuracy (on DEUCE, which has a 32 binary digit word, they will usually be correct to about 9 decimal places) and the problem is to find the corresponding eigenvectors. We shall consider the Givens codiagonal form first and later we shall point out the modifications which are necessary to deal satisfactorily with the Lanczos codiagonal form. It is very desirable that the method should be entirely automatic.

Corresponding to an exact eigenvalue, λ_1, of C, there is a non-zero solution of the set of equations

$$C\underline{x} = \lambda_1 \underline{x} \tag{3}$$

determined apart from an arbitrary multiplier. We may solve any (n-1) of the n homogeneous equations corresponding to (3) to obtain the ratios of the n components of $\underline{x}$. The remaining equation is then automatically satisfied. If λ is an approximation to λ_1 then the set of equations

$$C\underline{x} = \lambda \underline{x} \tag{4}$$

has no non-zero solution, but by omitting each of the equations in turn we may determine n vectors

$$\underline{x}^{(1)}, \underline{x}^{(2)}, \ldots\ldots\ldots \underline{x}^{(n)},$$

each of which is a function of λ, and we will expect that as λ tends to λ_1 each of these vectors will tend to $\underline{v}_1$, the eigenvector corresponding to λ_1. For a given approximation to an eigenvalue some of the $\underline{x}^{(i)}$ will be closer approximations to the eigenvector than others. In the next section the factors governing their rate of convergence are examined.

3. CONVERGENCE OF THE APPROXIMATE EIGENVECTORS

For the Givens form, equation (4) may be written

$$\begin{aligned}
(\alpha_1 - \lambda)x_1 + \beta_2 x_2 &= 0 \\
\beta_2 x_1 + (\alpha_2 - \lambda)x_2 + \beta_3 x_3 &= 0 \\
\beta_3 x_2 + (\alpha_3 - \lambda)x_3 + \beta_4 x_4 &= 0 \\
\cdots\cdots\cdots\cdots\cdots\cdots & \\
\beta_{n-1} x_{n-2} + (\alpha_{n-1} - \lambda)x_{n-1} + \beta_n x_n &= 0 \\
\beta_n x_{n-1} + (\alpha_n - \lambda)x_n &= 0
\end{aligned} \tag{5}$$

The most convenient sets of (n-1) equations to solve are the first (n-1) and the last (n-1) and it has frequently been suggested that one of these sets should be used for determining the eigenvectors. If the first (n-1) equations are used we may take $x_1 = 1$ and use the first equation to determine x_2, the second to determine x_3 and the $(n-1)$th to determine x_n. In general the nth equation will not be satisfied exactly by these values. If we omit an equation other than the first or the last, the situation is only slightly less simple. If we omit the rth equation the first $(r-1)$ equations give x_2 x_3 ..... x_r in terms of x_1 and the last $(n-r)$ give x_{n-1}, x_{n-2}, ... x_r in terms of x_n and we can therefore obtain all the variables in terms of x_1. The solution obtained when omitting the ith equation satisfies

$$\begin{aligned} \beta_j x_{j-1} + (\alpha_j - \lambda)x_j + \beta_{j+1} x_{j+1} &= 0 \qquad j \neq i \\ \beta_i x_{i-1} + (\alpha_i - \lambda)x_i + \beta_{i+1} x_{i+1} &= \delta \neq 0 \end{aligned} \tag{6}$$

and since we are not interested in arbitrary multipliers this solution is the same as that to the set of equations

$$(C - \lambda I)\underline{x} = \underline{e}_i \tag{7}$$

where $\underline{e}_i$ is the vector which has zero for all components except the ith, which is unity. If the true eigenvectors of C and $\underline{v}_1$ $\underline{v}_2$ ... $\underline{v}_n$ corresponding to eigen values λ_1, λ_2, ... λ_n then we may express $\underline{e}_i$ in terms of the $\underline{v}_j$s

$$\underline{e}_i = \sum_{j=1}^{n} \alpha_{ij} \underline{v}_j \tag{8}$$

The exact solution of the set of equations (7) is therefore

$$\underline{x} = \sum_{j=1}^{n} \alpha_{ij} (C - \lambda I)^{-1} \underline{v}_j \tag{9}$$

Now $(C - \lambda I)^{-1}$ has the same eigenvectors as C, while its eigenvalues are $1/(\lambda_j - \lambda)$. This gives

$$\underline{x} = \sum_{j=1}^{n} \alpha_{ij} \frac{1}{\lambda_j - \lambda} \underline{v}_j \tag{10}$$

If $\lambda = \lambda_1 + \varepsilon$ we have

$$\underline{x} = \alpha_{i1} \frac{1}{\varepsilon} \underline{v}_1 + \sum_{j=2}^{n} \alpha \frac{1}{\lambda_j - \lambda_1 - \varepsilon} \underline{v}_j \tag{11}$$

Provided α_{i1} is not zero we may deduce from (11) that

$$\underline{x} \to \underline{v} \quad \text{as} \quad \varepsilon \to 0.$$

However, in practice, 2 is always prescribed by the accuracy of the λ which we have calculated. Using single length arithmetic, on most available machines ε will be at best of the order of 10^{-10}. If then α_{i1} happens to be 'small', $\underline{x}$, as given by (11), will be by no means a good approximation to $\underline{v}_1$. If, for example, α_{i1} is itself of the order of 10^{-10} then $\underline{x}$ will contain components of other vectors which are as large as its components of $\underline{v}_1$.

In what follows the assumption will be made that $1/(\lambda_j - \lambda_1)$ is not large, that is, that there are no roots very close to λ_1. It is reasonable however to ignore impurities in an approximation to $\underline{v}_1$ which are of the same order of magnitude as $1/(\lambda_j - \lambda_1)$. The matrix C which we have regarded as exact from the point of view of this investigation will in fact have been derived by the Givens (or Lanczos) process and will therefore contain rounding errors. The errors in the calculated eigenvector $\underline{v}_1$ due to these rounding errors will contain the factors $1/(\lambda_j - \lambda) \times \underline{v}_j$ and such errors are therefore inherent in both the Givens and Lanczos processes, or indeed in any process which derives the eigenvectors of A via a transformed matrix B.

The latent vectors of Givens and Lanczos matrices are such that small α_{ij}s are quite common, for the following reasons. If one of the βs, β_r say, is exactly zero then the eigenvalues of C split into two groups, those which are the eigenvalues of the principal minor of order (r-1) and those which are the eigenvalues of the matrix of order (n-r+1) in the bottom right hand corner. Eigenvectors corresponding to eigenvalues of the first group have zero components in each of the last (n-r+1) positions. From equation (8) we have

$$\alpha_{ij} = \frac{\underline{v}_j' \cdot \underline{e}_i}{\underline{v}_j' \cdot \underline{v}_j} \qquad (12)$$

and therefore α_{ij} is zero for $i \geq r$ and $j < r$. If β_r is very small then we will have eigenvectors with very small components in the last (n-r+1) positions and these will give rise to small α_{ij}. A common situation with Givens matrices, and even more common with Lanczos matrices, is that a number of the βs are moderately small, and when this is true the roots of some principal minors of the codiagonal form are very close to the roots of the complete matrix. (In the Lanczos process it is common to employ a stratagem which exaggerates this tendency) Again, corresponding to these eigenvalues the eigenvectors will have small components in the lower positions. From this it is clear that none of the $\underline{e}_i$ is likely to produce consistently accurate eigenvectors and, in particular, that $\underline{e}_n$ and $\underline{e}_1$ will in general be the least satisfactory. This means that the first (n-1) equations of the n equations (5) will frequently produce inaccurate vectors.

In order to emphasize that it is not necessary for any of the βs to be pathologically small we may consider the matrix of order 21

$$\begin{bmatrix} 10 & 1 & & & & & \\ 1 & 9 & 1 & & & & \\ & 1 & 8 & 1 & & & \\ & & 1 \cdots\cdots & & & & \\ & & & 1 & -8 & 1 & \\ & & & & 1 & -9 & 1 \\ & & & & & 1 & -10 \end{bmatrix}$$

The largest root of this matrix is clearly greater than 10. If we denote its exact value by λ, then, using this value, we may obtain the eigenvector from any (n-1) equations. Suppose we use the last (n-1) and put $x_{21} = 1$. We have

$$x_{20} = (\lambda + 10)x_{21}$$

$$x_{19} = (\lambda + 9)x_{20} - x_{21}$$

$$x_{18} = (\lambda + 8)x_{19} - x_{20}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

giving

$$x_{20} = (\lambda + 10)$$

$$x_{19} = (\lambda + 9)(\lambda + 10) - 1$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

Since λ is greater than 10 it is clear that the terms of this sequence continue to increase very rapidly for a number of stages so that in the normalised vector corresponding to λ the last element is very small. The vector, correct to eight decimals, is given below. The eigenvector which would be obtained by solving the first 20 equations with a value of λ which was in error by only 10^{-10} would be hopelessly inaccurate.

```
+1.00000000
 0.74619419
 0.30299994
 0.08590250
 0.01880748
 0.00336146
 0.00050815
 0.00006659
 0.00000771
 0.00000080
 0.00000007
 0.00000001
 0.00000000
 0.00000000
 0.00000000
 0.00000000
 0.00000000
 0.00000000
 0.00000000
 0.00000000
+0.00000000
```

The value of the eigenvalue correct to 9 significant decimals is 10.7461942.

4. PRACTICAL PROCEDURE FOR CALCULATING EIGENVECTORS

Instead of solving the equations

$$(C - \lambda I)\underline{x} = \underline{e}_i \qquad (13)$$

for some value of i we analyse the solution of

$$(C - \lambda I)\underline{x} = \underline{b} \qquad (14)$$

for any vector $\underline{b}$. If $\underline{b}$ is expressed in terms of the eigenvectors $\underline{v}_1\ \underline{v}_2 \cdots \underline{v}_n$

$$\underline{b} = \sum_{1}^{n} \gamma_i \underline{v}_i \tag{15}$$

then we have

$$\underline{x} = \sum_{1}^{n} \gamma_i \frac{1}{\lambda_i - \lambda} \underline{v}_i \tag{16}$$

If λ is an accurate approximation to a root λ_j then it is clear from (16) that the corresponding $\underline{x}$ will be an accurate approximation to $\underline{v}_j$ provided $\underline{b}$ is chosen so that it is not particularly defective in $\underline{v}_j$.

In the procedure programmed for DEUCE $\underline{b}$ is chosen in a way which makes it unlikely that it is particularly defective in any of the $\underline{v}_i$ and at the same time so as to minimise the computation involved in calculating $\underline{x}$ from (14). The equations may be written in full

$$\begin{aligned}
(\alpha_1 - \lambda)x_1 + \beta_2 x_2 &= b_1 \\
\beta_2 x_1 + (\alpha_2 - \lambda)x_2 + \beta_3 x_3 &= b_2 \\
\beta_3 x_2 + (\alpha_3 - \lambda)x_3 + \beta_4 x_4 &= b_3 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\
\beta_{n-1} x_{n-2} + (\alpha_{n-1} - \lambda)x_{n-1} + \beta_n x_n &= b_{n-1} \\
\beta_n x_{n-1} + (\alpha_n - \lambda)x_n &= b_n
\end{aligned} \tag{17}$$

This set of equations is solved by successive elimination, eliminating the variables x_1, x_2, ... x_{n-1} in their natural order but taking as 'pivotal' row at each stage that equation which has the largest coefficient of the variable which is being eliminated. At each stage there will be only two equations containing that variable. Because the equations are not necessarily used in their natural order the resulting equations must be written

$$\begin{aligned}
p_1 x_1 + q_1 x_2 + r_1 x_3 &= c_1 \\
p_2 x_2 + q_2 x_3 + r_2 x_4 &= c_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\
p_{n-2} x_{n-2} + q_{n-2} x_{n-1} + r_{n-2} x_n &= c_{n-2} \\
p_{n-1} x_{n-1} + q_{n-1} x_n &= c_{n-1} \\
p_n c_n &= c_n
\end{aligned} \tag{18}$$

though usually many of the r_i s will be zero. The c_i s are derived from the b_i s, which are still at our disposal. We may assume that the b_i s were chosen so that $c_i = 1$ for all values of i, and in this way we avoid any necessity for computation of the right hand sides of (18). We therefore solve the set of equations (18) with $c_i = 1$. It is obvious that there is no guarantee that the vector $\underline{b}$ corresponding to the value chosen for $\underline{c}$ will not be deficient in any $\underline{v}_i$, but it is reasonable to assume that this is improbable. The process has been used in exactly the above form on a large number of matrices, many of which have produced codiagonal forms with very small β_i s.

The vectors have been consistently accurate. The time taken to find the eigenvectors of the codiagonal form by this method is, however, quite a small percentage of the total time taken for the whole problem; so that, if it were found on a more extended test that it was not satisfactory, the following alternative, which is somewhat longer, could be used.

By the above process we may calculate $\underline{x}_1$ corresponding to the known approximation λ. This vector $\underline{x}_1$ may now be used as the vector $\underline{b}$ in the sets of equations (17). Even if $\underline{x}_1$ is not a sufficiently good approximation to $\underline{v}_1$ it will certainly contain a substantial component of $\underline{v}_1$ so that the vector $\underline{x}_2$ obtained with $\underline{b} = \underline{x}_1$ will be a very accurate eigenvector. There is about twice as much work in this process as in that which has been described previously, but even with this refinement the time taken to calculate the eigenvectors of the codiagonal form is quite satisfactory. When deriving $\underline{x}_2$ from $\underline{x}_1$ there is no need to recalculate the p_i s, q_i s and r_i s since they are the same as before. If a copy is retained of the multiplying factors used in the initial reduction, the amount of work involved in calculating $\underline{x}_2$ is less than that in calculating $\underline{x}_1$. It is not possible to obtain unlimited accuracy in the vectors of C (regarded for this purpose as an exact matrix) by repeating the above process to obtain vectors $\underline{x}_3$, $\underline{x}_4$ .... etc., but it is quite easy to obtain a vector of more than single length accuracy without resorting to double length arithmetic, while still using the original approximate value of λ and the original values of p_i, q_i and r_i. The process described above is repeated until two successive values of $\underline{x}_i$ (when normalised in some convenient way) agree to within three or four binary places. This will seldom mean producing anything further than $\underline{x}_3$. We then form the vector $\underline{r}$ defined by

$$\underline{r} = \underline{x}_{i-1} - (C - \lambda I)\underline{x}_i$$

and use $\underline{r}$ as our vector $\underline{b}$ in the equations (17). The vector $\underline{r}$ will be small compared with $\underline{x}_{i-1}$ and we can therefore take more binary places in $\underline{r}$ than in $\underline{x}_{i-1}$. The figures in these extra binary positions will be determined exactly provided all the figures involved in the multiplication of $\underline{x}_i$ by $(C - \lambda I)$ are retained. If $(C - \lambda I)\underline{y} = \underline{r}$ then $(\underline{x}_i + \underline{y})$ is a more accurate vector.

5. THE LANCZOS PROCESS (2)

Although the essential details of what has been described above apply equally well to the Lanczos process for symmetrical matrices, there are some points which call for comment. The codiagonal form is derived as follows. Starting with an arbitrary vectors $\underline{x}_1$, a set of n orthogonal vectors $\underline{x}_1$ $\underline{x}_2$ $\underline{x}_3$ ..... $\underline{x}_n$ is derived by the following relations

$$\begin{aligned}
\underline{x}_2 &= A\underline{x}_1 - \alpha_1 \underline{x}_1 \\
\underline{x}_3 &= A\underline{x}_2 - \alpha_2 \underline{x}_2 - \beta_2 \underline{x}_1 \\
\underline{x}_4 &= A\underline{x}_3 - \alpha_3 \underline{x}_3 - \beta_3 \underline{x}_2 \\
&\cdots\cdots\cdots\cdots\cdots\cdots \qquad (19) \\
\underline{x}_n &= A\underline{x}_{n-1} - \alpha_{n-1} \underline{x}_{n-1} - \beta_{n-1} \underline{x}_{n-2}
\end{aligned}$$

and $$\underline{x}_{n+1} = A\underline{x}_n - \alpha_n \underline{x}_n - \beta_n \underline{x}_{n-1}$$

where $\underline{x}_{n+1}$ should be the null vector.

In practice it is necessary, because of rounding errors, to orthogonalise with respect to all the earlier vectors, each $\underline{x}_i$ as found by the above rules. The set of vectors, $\underline{x}_i$, derived in this manner is usually such that there is a progressive decrease in the value of the moduli with increasing i. It is quite common for $\underline{x}_n$ to be very much smaller than $\underline{x}_1$. Since, as can easily be shown

$$\beta_i = \frac{\underline{x}_i' \, \underline{x}_i}{\underline{x}_{i-1}' \, \underline{x}_{i-1}} \qquad (20)$$

this means that small β_i are quite common. If X is the matrix which has its ith column equal to $\underline{x}_i$ and Y is the matrix which has its ith row equal to $\underline{x}_i/(\underline{x}_i' \, \underline{x}_i)$ then

$$YX = I$$

and

$$YAX = C_2$$

$$= \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ 1 & \alpha_2 & \beta_3 & & & \\ & 1 & \alpha_2 & \beta_4 & & \\ & \cdots & \cdots & \cdots & & \\ & & \cdots & \cdots & \cdots & \\ & & & 1 & \alpha_{n-1} & \beta_n \\ & & & & 1 & \alpha_n \end{bmatrix} \qquad (21)$$

Because of the variation in size of the $\underline{x}_i$ s it is usual to work with a floating binary representation for each element of $\underline{x}_i$ or alternatively to associate a multiplying power of 2 with the whole of each $\underline{x}_i$ (usually called block floating operation).

Suppose λ is a root of C_2 and $\underline{v}$ is the corresponding vector, then the eigenvector of the original matrix is $X\underline{v}$ and this may be expressed in the form

$$v_1 \underline{x}_1 + v_2 \underline{x}_2 + \cdots\cdots + v_n \underline{x}_n$$

that is, as a linear combination of the $\underline{x}_i$ s. The variation in size of the $\underline{x}_i$ leads to rather tiresom requirements on the accuracy of the components of $\underline{v}$. It might be imagined that the use of floating binary arithmetic meets these requirements automatically, but this is not so. A simple example of a matrix of order two will suffice to illustrate this. Suppose

$$x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad x_2 = 1.6 \times 10^{-4} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ and } C_2 = \begin{pmatrix} 4 & 2.56 \times 10^{-8} \\ 1 & 3 \end{pmatrix}$$

We shall use floating decimal arithmetic with 9 significant decimals (roughly the equivalent of DEUCE floating binary). The roots of C_2 are

4.00000003 and 2.99999997

with errors beyond the last figure quoted. If we take the first root and obtain a vector from the first equation we have

$$3 \times 10^{-8}\, v_1 = 2.56 \times 10^{-8}\, v_2$$

giving as normalised $\underline{v}$ the vector

$$\begin{bmatrix} 2.56/3 \\ 1 \end{bmatrix}$$

where 2.56/3 should be calculated to 9 significant decimals. This is obviously a very inaccurate eigenvector of C_2 since the correct vector has, instead of the component 2.56/3, the component 2.56/3.a where we use a to denote the figures which should follow the 3 in the root 4.00000003. In fact, this vector is wrong in the second decimal place. The corresponding vector of the original matrix is given as

$$1 \times \binom{1}{0} + \frac{2.56}{3} \times 1.6 \times 10^{-4} \binom{0}{1}$$

and from this it is clear that the error obtained in the vector of the original matrix is in the fifth decimal.

If we had used the second equation we would have produced accurate vectors of both C_2 and the original matrix. With the second root the situation is more misleading. Suppose we use the second equation. We obtain

$$v_1 = -3.0 \times 10^{-8}\, v_2$$

giving as the normalised $\underline{v}$

$$\begin{bmatrix} -3 \times 10^{-8} \\ 1 \end{bmatrix}$$

This vector is correct to 9 decimal places and is therefore perfectly satisfactory as a vector of C_2. For the vector of the original matrix it produces

$$-3 \times 10^{-8} \binom{1}{0} + 1.6 \times 10^{-4} \binom{0}{1}$$

To normalise this we have to multiply by a factor which is of the order of 10^4. We accordingly obtain a normalised vector with an error in the fifth decimal.

It is obvious from this that it will not be satisfactory to solve $(C_2 - \lambda I)\underline{x} = \underline{b}$ with $\underline{b}$ chosen as defined for the Givens process. The simplest way of avoiding the difficulty is to work with the set of normalised vectors $\underline{z}_i$ derived from $\underline{x}_i$ by the relation

$$\underline{z}_i = \underline{x}_i / (\underline{x}_i' \underline{x}_i)^{\frac{1}{2}}$$

If Z is the matrix which has its i th column equal to $\underline{z}_i$ then

$$Z' A Z = \begin{bmatrix} \alpha_1 & \sqrt{\beta_2} & & & & \\ \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & & & \\ & \sqrt{\beta_3} & \alpha_3 & & & \\ & & \cdots & \cdots & & \\ & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} & \sqrt{\beta_n} \\ & & & & \sqrt{\beta_n} & \alpha_n \end{bmatrix} \qquad (22)$$

and $Z' Z = I$

The problem is now identical with that in the Givens method. Lanczos has described a transformation similar to that given above which may be applied to an unsymmetric matrix. Starting from two arbitrary vectors x_1 and x_1^* a biorthogonal sequence is formed from the relations

$$\begin{aligned} \underline{x}_{r+1} &= A\underline{x}_r - \alpha_r \underline{x}_r - \beta_r \underline{x}_{r-1} \\ \underline{x}^*_{r+1} &= A'\underline{x}^*_r - \alpha_r \underline{x}^*_r - \beta_r x^*_{r-1} \end{aligned} \qquad (23)$$

If X is the matrix, the ith column of which is $\underline{x}_i$ and Y is the matrix the ith row of which is $y_i/(\underline{x}_i' \underline{y}_i)$ then

$$\begin{aligned} YX &= I \\ \text{and} \quad YAX &= C_2. \end{aligned} \qquad (24)$$

The β_i are now no longer necessarily positive so that the matrix C_2 cannot be transformed into a symmetric matrix. However if T is the matrix defined by

$$T = \begin{bmatrix} 1 & & & \\ & \sqrt{\beta_n} & & \\ & & \sqrt{\beta_2 \ \beta_3} & \\ & & & \sqrt{\beta_2 \ \beta_3 \ \beta_n} \end{bmatrix} \qquad (25)$$

$$\text{then } TC_2T^{-1} = \begin{bmatrix} \alpha_1 & \varepsilon_2\sqrt{\beta_2} & & & \\ \sqrt{\beta_2} & \alpha_2 & \varepsilon_3\sqrt{\beta_3} & & \\ & \sqrt{\beta_3} & \alpha_3 & & \\ & & & \sqrt{\beta_n} & \alpha_n \end{bmatrix}$$

where $\varepsilon_i = \dfrac{\beta_i}{\beta_i} = \pm 1$. The same technique may now be used though the roots are, of course, not necessarily real.

NUMERICAL EXAMPLES

Two examples are given which illustrate the above points. The first is derived from the matrix of order 14 given by Brooker(3).

MATRIX 1

0.25000	0.06675	0.04000	0.02475	0.07050	0.06375	0.06925	0.02050	0.03600	-0.01025	-0.00175	0.02750	0.02300	0.00200
	0.25000	0.10400	0.07475	0.03625	0.11675	0.11050	0.06225	0.05100	0.03250	0.02400	0.03600	0.06350	0.03300
		0.25000	0.14575	0.03725	0.07157	0.07800	0.12200	0.11275	0.09375	0.10175	0.09600	0.14300	0.11550
			0.25000	0.05375	0.07000	0.05225	0.12800	0.12475	0.10550	0.13000	0.14575	0.13975	0.13375
				0.25000	0.04575	0.05750	0.05700	0.05050	0.01475	0.04500	0.07150	0.05300	0.01600
					0.25000	0.08625	0.08800	0.07150	0.04850	0.03200	0.04475	0.03300	0.04500
						0.25000	0.08725	0.06950	0.03725	0.04025	0.04300	0.04075	0.01450
							0.25000	0.14100	0.13275	0.15550	0.13050	0.11825	0.09125
								0.25000	0.07425	0.10750	0.09175	0.10725	0.08225
									0.25000	0.15500	0.09625	0.09950	0.09425
										0.25000	0.13350	0.14850	0.13050
											0.25000	0.11100	0.10075
												0.25000	0.14325
													0.25000

GIVENS CODIAGONAL FORM AND ITS ROOTS

α_i	β_i	λ_i
+0.250000000	+0.000000000	+1.334034844
0.768491173	0.153660746	0.462766202
0.919556756	0.467260328	0.267733297
0.230938895	0.119256498	0.231639484
0.133053788	0.080763539	0.177356337
0.222549575	0.033947196	0.171307560
0.116127856	0.036090904	0.166324602
0.120339373	0.035022375	0.143422880
0.123719912	0.029157561	0.122787524
0.128561407	0.037453705	0.103215761
0.107768089	0.016090599	0.097209219
0.137039203	0.023826467	0.084225269
0.138057030	0.029468449	0.073597119
+0.103796943	+0.007646394	+0.064379910

First four Eigenvectors of codiagonal form calculated by DEUCE programme.

+0.12180621	+0.72220468	+0.49474263	-0.34701603
+0.85930971	+1.00000000	+0.05709601	+0.04146403
+1.00000000	-0.89179298	-0.22388762	+0.06647838
+0.10864611	-0.50225957	+1.00000000	-0.54593352
+0.00731255	-0.12487537	+0.78617724	-0.10289858
+0.00022356	-0.01793071	+0.73992634	+1.00000000
+0.00000663	-0.00188653	+0.18686459	+0.34864816
+0.00000019	-0.00019439	+0.04640116	+0.11940974
+0.00000001	-0.00001692	+0.01011081	+0.03703434
+0.00000000	-0.00000190	+0.00275399	+0.01375120
+0.00000000	-0.00000009	+0.00028519	+0.00188770
+0.00000000	-0.00000000	+0.00005483	+0.00052743
+0.00000000	+0.00000000	+0.00001250	+0.00016690
+0.00000000	+0.00000000	+0.00000059	+0.00000998

The eigenvectors of the codiagonal matrix illustrate the phenomenon referred to. The later vectors do not provide any points of interest and are quite normal in form. All vectors were correct to within 2 units of the seventh decimal, the largest errors being in the vectors corresponding to the smallest eigenvalues. For the four vectors quoted above the maximum error is 6 units in the eighth decimal place. In each of the first two vectors the last component is so small that we would expect the solution of the first $(n-1)$ equations to be quite valueless. This is illustrated below where the vector obtained by inserting the approximate value of λ_1 in the first $(n-1)$ equations is given. To emphasize that the results is not due to solving this equations inaccurately with this value of λ_1 the $(n-1)$ equations were solved to double length accuracy. The resulting vector is given below in normalised form.

```
 + .0000 0000
   .0000 0000
   .0000 0000
   .0000 0000
   .0000 0000
   .0000 0000
   .0000 0000
   .0000 0000
   .0000 0000
   .0000 0004
   .0000 0306
   .0001 5755
   .0063 9731
 +1.0000 0000
```

As a second example we consider the matrix given below which arose in the theory of molecular arbitals.

MATRIX 2

$$\begin{matrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{matrix}$$

The eigenvalues produced by the DEUCE programme included the three values.

$$\lambda_5 = 1 + 2^{-26}$$

$$\lambda_6 = 1 \text{ exactly}$$

$$\lambda_7 = 1 - 2^{-27}$$

which were the approximations produced for the triple root $\lambda = 1$. The corresponding normalised vectors were

v_6	v_7	v_8
-01108773	+04540761	-04116218
-03033805	+01960617	+00402348
-01925032	-02580145	+04518566
+01108773	-04540761	+04116218
+03033805	-01960617	-00402348
+00000000	+00000000	+00000000
-03033805	+01960617	+00402348
+01530415	+03486117	+00720252
+04564220	+01525500	+00317904
+03033805	-01960617	-00402348
-01530414	-03486117	-00720253
-04564220	-01525500	-00317905
+00000000	+00000000	+00000000
+01925032	+02580145	-04518566
+02639188	-01054644	+04836470

These vectors were produced entirely automatically and it may easily be verified that they are accurate eigenvectors. They are also independent and therefore form a complete set of vectors corresponding to $\lambda = 1$. That this should happen is not surprising when we consider the meaning of equation (16) for coincident roots. An interesting feature of this example is that we may easily guess three eigenvectors of A from any one of the vectors v_6, v_7, or v_8.

For example v_6 is of the form

$$\begin{array}{c} -a \\ -b \\ (a-b) \\ a \\ b \\ 0 \\ -b \\ c \\ (b+c) \\ b \\ -c \\ (-b-c) \\ 0 \\ (-a+b) \\ (a+c) \end{array} \qquad \begin{array}{l} \text{where } a = .1108773 \\ \phantom{\text{where }} b = .3033805 \\ \phantom{\text{where }} c = .1530415 \end{array}$$

Since the a, b, and c do not appear to be simply related we might expect the vector v_6 to split into

$$a\underline{x} + b\underline{y} + c\underline{z}$$

where $\underline{x}$, $\underline{y}$ and $\underline{z}$ were eigenvectors. $\underline{x}$ $\underline{y}$ and $\underline{z}$ are in fact

-1	0	0
0	-1	0
1	-1	0
1	0	0
0	1	0
0	0	0
0	-1	0
0	0	1
0	1	1
0	1	0
0	0	-1
0	-1	-1
0	0	0
1	1	0
1	0	1

It may readily be verified that $\underline{x}$ $\underline{y}$ and $\underline{z}$ are independent vectors of the original matrix. The above device has frequently been used with "exact" matrices with "exact" coincident roots.

REFERENCES

No.	Author	Title
1.	Givens, W.	"Numerical Computation of the characteristic values of a real symmetric matrix". Oak Ridge Laboratory Report No. 1574.
2.	Lanczos, C.	"On Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators". Journal of Research of the National Bureau of Standards 1950, 45, p.255.
3.	Brooker, R.A. and Sumner, F.H.	"The Method of Lanczos for Calculating the Characteristic Roots and Vectors of a Real Symmetric Matrix". Paper presented at a convention on Digital Computer Techniques April 1956 at the Institution of Electrical Engineers. To be published by the Institution.

* * * * *