An Efficient Scheme for the Co-diagonalization of a Symmetric Matrix by Givens' Method in a Computer with a Two-level Store

By J. S. Rollett and J. H. Wilkinson

The Givens' process, as it is commonly performed, requires simultaneous rapid access both to rows and to columns of the matrix. By running in parallel the various rotations which produce a row of zeros it is possible to reduce the number of scans of the matrix by a factor of order *n*. The scanning required is then less than that for Householder's process, which is, however, faster in other ways. The number of multiplications and the rounding errors for the modified Givens' process are the same as when the rotations are carried out sequentially, but on Mercury there is an overall gain in speed by a factor of 7 for matrices of order 96.

1. Introduction

The Givens' process in its usual form (Givens, 1954) loses much of its effectiveness when the order n of the matrix is so large that the coefficients cannot be kept in the computing store but must be drawn from the backing store row by row or column by column. The reason for this is that the typical elementary step in which the zero in the (i, j) position is produced consists of the calculation of the matrix $A^{(r+1)}$ defined by

$$A^{(r+1)} = T^{(r)}A^{(r)}(T^{(r)})^T \tag{1}$$

where $T^{(r)}$ is an orthogonal matrix defined by

$$\left.\begin{aligned} t^{(r)}_{i+1,\, i+1} &= t^{(r)}_{jj} = \cos\theta_{i+1,\, j}; \\ t^{(r)}_{i+1,\, j} &= -\, t^{(r)}_{j,\, i+1} = \sin\theta_{i+1,\, j}; \\ t^{(r)}_{kk} &= 1 \quad (k \neq i+1, j); \\ t^{(r)}_{pq} &= 0 \text{ otherwise.} \end{aligned}\right\} \tag{2}$$

This involves replacing rows $i + 1$ and j and columns $i + 1$ and j by linear combinations. If the matrix is stored by rows then rows i to n must be brought to the computing store, though, except for rows $i + 1$ and j, only the elements $i + 1$ and j have to be altered. The position is only slightly better when advantage is taken of the symmetry of the matrix to store only the upper triangle. A detailed account of a technique which overcomes this difficulty is given here, and comparative times are quoted for the two methods applied to large matrices on the magnetic drums of the Mercury computer.

2. Recommended Procedure

It is usual to describe the Givens' process as consisting of $\frac{1}{2}(n - 1)(n - 2)$ steps in which zeros are produced successively in positions (1, 3), (1, 4), . . ., (1, n); (2, 4), (2, 5), . . ., (2, n); . . .; (n − 2, n). The zero in position (i, j) is produced by the transformation defined by equations (1) and (2) with $\theta_{i+1,\, j}$ given by

$$\tan\theta_{i+1,\, j} = a^{(r)}_{ij}/a^{(r)}_{i,\, i+1}. \tag{3}$$

Since each transformation preserves symmetry and also leaves undisturbed the zeros produced by previous transformations, the final matrix is of co-diagonal form.

For the purposes of this paper, it is convenient to group together the $(n - i - 1)$ transformations which produce the zeros in the ith row and column and thus to think of the process as consisting of $(n - 2)$ major steps.

The modified procedure which we now describe is such that in the whole of the ith major step only one transfer to the computing store and back again is made for each of rows i to n. The procedure requires a total of $4n$ registers for numbers in the computing store and these are divided into four groups of n registers each.

The first $(i - 1)$ rows and columns play no part in the ith major step, which is formally similar to the first, but operates on the matrix of reduced order $(n - i + 1)$ in the bottom right-hand corner. There is thus no loss of generality in describing the first major step and this will be done for the sake of simplicity. This step has five main stages.

(1) The first row of A is transferred to the registers in group 1.

(2) The values of $\cos\theta_{23}$, $\sin\theta_{23}$; . . .; $\cos\theta_{2n}$, $\sin\theta_{2n}$ are computed successively from:

$$\cos\theta_{2j} = a^{(j-1)}_{12}/\sqrt{[(a^{(j-1)}_{12})^2 + a^2_{1\,j}]} \tag{4}$$

$$\sin\theta_{2j} = a_{1\,j}/\sqrt{[(a^{(j-1)}_{12})^2 + a^2_{1\,j}]} \tag{5}$$

where

$$a^{(2)}_{12} = a_{12} \tag{6}$$

$$a^{(j)}_{12} = \sqrt{[(a^{(j-1)}_{12})^2 + a^2_{1\,j}]}. \tag{7}$$

The $\cos\theta_{2\,j}$ may be overwritten on the $a_{1\,j}$, which are no longer required, and the $\sin\theta_{2\,j}$ are stored in the group 2 registers.

(3) The second row is transferred to the registers in group 3. Only those elements on and above the diagonal are used in this and all succeeding rows.

For $k = 3, 4, \ldots, n$ in turn, the operations in stages (4) and (5) are carried out:

(4*a*) The kth row is transferred to the registers in group 4.

(4*b*) The elements a_{22}, a_{2k} and a_{kk} are subjected to the row and column operations involving $\cos\theta_{2k}$ and

$\sin\theta_{2k}$ (here and later we use a_{ij} to denote the number currently occupying the (i, j) storage location).

(4c) For $l = k + 1, k + 2, \ldots, n$ in turn, the part of the row transformation involving $\cos\theta_{2k}$ and $\sin\theta_{2k}$ is performed on a_{2l} and a_{kl}.

(4d) For $l = k + 1, k + 2, \ldots, n$ in turn, the part of the column transformation involving $\cos\theta_{2l}$ and $\sin\theta_{2l}$ is performed on a_{2k} and a_{kl}, taking advantage of the fact that $a_{2k} \equiv a_{k2}$, by symmetry. When (4a), (4b), (4c) and (4d) have been completed for a given k, all the transformations involved in the first major step have been performed on all the elements of row k and on elements $3, 4, \ldots, k$ of row 2. Elements $2, k + 1, k + 2, \ldots, n$ of row 2 have been subjected only to the transformations involving $\theta_{2,3}, \theta_{2,4}, \ldots, \theta_{2k}$.

(5) The completed kth row is transferred to the backing store and we return to (4a) with the next value of k.

When stages (4) and (5) have been completed for all appropriate values of k, the whole of the work on row 2 has also been completed. The $\cos\theta_{2k}$ and $\sin\theta_{2k}$ ($k = 3$ to n) and the modified elements of the first row (i.e. the first two elements of the co-diagonal form) are written on the backing store and row 2 is transferred to the group 1 registers, either physically, or by a change of the group labels. Since the second row plays the same part in the second major step as did the first row in the first major step, everything is then ready for stage (2) in the second major step. Stage (1) is not needed in any of the later major steps because the appropriate row is already in the computing store.

It is not very helpful to express this sequence of operations in the language of matrix multiplications, but it should be appreciated that at the end of each major step each element of the transformed matrix has precisely the same value as it has at the corresponding point in the usual procedure. Although some parts of the later transformations in a major step are done before the earlier transformations have been completed, the elements affected by these later transformations are not subsequently involved in the completion of the earlier transformations. There is essentially no difference between the two schemes as far as the number of arithmetic operations and the rounding errors are concerned, but the number of transfers from and to the backing store is substantially reduced.

3. Storage Requirements

Space is required in the computing store for the four vectors of order n and the program. On the Mercury computer, for which programs using the method have been written, there are 1,024 registers in the computing store and this has limited the method to values of n satisfying $n \leqslant 160$, independent of the size of the backing store. On the Oxford University machine the backing store holds 16,320 numbers on two magnetic drums, and this has limited the method further to matrices of orders up to 115. The size of the computing store is not the main limitation in this case, but if the computing store were smaller (as on Pegasus or DEUCE) or if the backing store were extended by adding further drums or magnetic tapes, this would no longer be true.

It is convenient to allocate n^2 locations for the matrix in the backing store, storing each row in full, in spite of the fact that the elements below the diagonal are not used. Some advantages of this are as follows.

(a) The matrix can be derived by previous operations with standard matrix routines (Brooker, Richards, Berg and Kerr, 1959) and needs no special packing before submission to the Givens' library program. This is hardly efficient, but it is highly attractive to relatively unskilled programmers attacking small problems.

(b) The element $\cos\theta_{ij}$ may be overwritten on $a_{i-1,j}$ and $\sin\theta_{ij}$ on $a_{n+2-i,j-i}$, and hence the matrix and the quantities used in calculating the vectors of $\boldsymbol{A}$ from those of the co-diagonal form can all be stored in the same compact block of n^2 locations. The final configuration on the backing store for a matrix of order 5 is then

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cos\theta_{2,3} & \cos\theta_{2,4} & \cos\theta_{2,5} \\ x & a_{2,2} & a_{2,3} & \cos\theta_{3,4} & \cos\theta_{3,5} \\ \sin\theta_{4,5} & x & a_{3,3} & a_{3,4} & \cos\theta_{4,5} \\ \sin\theta_{3,4} & \sin\theta_{3,5} & x & a_{4,4} & a_{4,5} \\ \sin\theta_{2,3} & \sin\theta_{2,4} & \sin\theta_{2,5} & x & a_{5,5} \end{bmatrix} \quad (8)$$

This is very convenient when the backing store is on magnetic drums, as for Mercury, but is less attractive if the backing store is a magnetic tape which can be read only in the forward direction. There is then difficulty in obtaining rapid access to the sines and cosines when calculating the vectors.

The elements of the co-diagonal form are returned to the backing store in this way by the library sub-program which we describe below, so that the main program can be re-entered between the co-diagonalization and the determination of roots and vectors. It would be possible in a complete program to keep a copy in the computing store in the registers of groups 1 and 2 which are not required for work on the successive matrices of order $n - i + 1$.

(c) The address arithmetic for the transfers between the stores is simplified. This is a small advantage, but it helps to shorten the program which competes with the vectors for the computing store.

4. Experience with the Method

One complete program, which punches out as many as are required of the eigenvalues and eigenvectors of a matrix read from paper tape, and one library sub-program (programme—517 of the Autocode system), which writes in the backing store similar results for a matrix which was stored there, have been written (by

J. S. R.) for the Mercury computer. The complete program handles matrices of orders up to 96. The subroutine is not quite so fast as the complete program for a matrix of the same order, but will deal with matrices of orders up to 115. For matrices of the highest orders the subroutine suffers from the slight disadvantage that it is necessary to overwrite the Autocode compiler.

In an earlier version of the complete program, the rotations were carried out one by one, and although this was reasonably efficient for matrices of small orders, for matrices of order 40 or more most of the time was spent on drum transfers. Comparative times for co-diagonalization by this earlier version and the complete program based on the methods of this paper are:

ORDER	ORTHODOX METHOD	REVISED METHOD
64	20 min	6 min
96	140 min	20 min

The modified version not only shows a very substantial saving of machine time, but also removes the need to consider provision for re-starting, since an error-free running time of 20 minutes is reasonably certain.

5. Comparison with the Method of Householder

In a recent paper one of us (Wilkinson, 1960) described a program based on Householder's method (Householder and Bauer, 1959). This is also a method for reducing a symmetric matrix to co-diagonal form, and it involves half as many multiplications as the Givens' process. The method described there requires *two* complete scans of the reduced matrix for each transformation introducing a row of zeros. It does not seem to be possible to reduce this to one scan per row as we have done in the modified Givens' process.

It is commonly believed that the matrix of the Householder transformation which produces a row of zeros is the product of the Givens' rotations which produce the same row of zeros, but this is not so. For consider the Householder transformation which produces the zeros in the first row. The matrix $\boldsymbol{P}$, of this transformation is $(\boldsymbol{I} - 2w\,w^T)$ where

$$w^T = (0, x_2, x_3, \ldots, x_n), \tag{9}$$

and, in general, none of the x_i is zero. (See, for example, Wilkinson, 1960, p. 24.) Hence, for a matrix of order five we have,

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \tag{10}$$

where $\boldsymbol{P}$ is symmetric and, in general, none of the elements in the matrix of order four in the bottom right-hand corner is zero.

The Givens' transformations producing the zeros in the first row are plane rotations in the (2, 3), (2, 4) and (2, 5) planes. The product of the three corresponding matrices is of the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & 0 & x & x \\ 0 & x & 0 & 0 & x \end{bmatrix} \tag{11}$$

as may readily be verified by multiplying the identity matrix successively on the right by the appropriate $\boldsymbol{T}^{(r)}$ matrices. In general there is a triangle of zero elements in the positions shown in (11). For a matrix of order n this triangle consists of $\frac{1}{2}(n-3)(n-2)$ zeros. The Givens' and Householder matrices are therefore different in general. However, it *is* true that the final co-diagonal matrices and the product of *all* the transformation matrices are the same in both cases apart from signs. For in each case the product of the transformation matrices is orthogonal and has its first column equal to e_1, the first column of the unit matrix. We now show that if $\boldsymbol{R}$ is an orthogonal matrix having this property and if

$$\boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R} = \boldsymbol{C} \tag{12}$$

where $\boldsymbol{C}$ is co-diagonal, then both $\boldsymbol{R}$ and $\boldsymbol{C}$ are uniquely determined apart from signs. We may write

$$\boldsymbol{R} = [e_1 \,\vdots\, r_2 \,\vdots\, r_3 \,\vdots\, \ldots \,\vdots\, r_n] \tag{13}$$

so that the r_i are the columns of $\boldsymbol{R}$. Clearly $\boldsymbol{C}$ is symmetric in any case.

Writing equation (12) in the form

$$\boldsymbol{A}\boldsymbol{R} = \boldsymbol{R}\boldsymbol{C} \tag{14}$$

and equating the first columns we have

$$\boldsymbol{A}e_1 = c_{11}e_1 + c_{21}r_2. \tag{15}$$

Now the columns of $\boldsymbol{R}$ are of unit length and orthogonal to each other. Hence

$$e_1^T\boldsymbol{A}e_1 = c_{11}, \tag{16}$$

so that c_{11} is uniquely determined. Equation (15) now gives

$$c_{21}r_2 = \boldsymbol{A}e_1 - c_{11}e_1 \tag{17}$$

and the right-hand side is uniquely determined. Since r_2 is to be of unit length, c_{21} and r_2 are uniquely determined apart from their signs. Equating each of the columns in succession and making use of the symmetry of $\boldsymbol{C}$ and the orthonormality of the columns of $\boldsymbol{R}$, we find that each element of $\boldsymbol{C}$ and each column of $\boldsymbol{R}$ is uniquely determined apart from its sign.

Our proof shows incidentally that the co-diagonal form produced by the symmetric Lanczos method (Lanczos, 1950) is also the same if the initial vector is taken to be e_1 and each vector in the orthogonal system is normalized.

6. Conclusions

A program for the Mercury computer, using the scheme described in this paper, has been tested for a matrix of order 96. The fraction of the total time spent on drum transfers is quite small in this case. For a computer with a fast arithmetic unit in which the backing store was magnetic tape, so that the speed of transfer from the backing store was the effective controlling factor, the speed factor gained by minimizing the number of scans through the matrix would be even greater than that of seven obtained with Mercury.

It is worth noting that in this single respect the Givens' process is better than that of Householder which needs two scans to produce each row of zeros. In other ways the Householder process still has the advantage, requiring only about $\frac{2}{3}n^3$ multiplications as against $\frac{4}{3}n^3$, and $\frac{1}{2}n(n+1)$ backing stores as against n^2. The requirement for n^2 stores can be reduced to $\frac{1}{2}n(n+1)$ if the smaller of $\cos\theta_{ij}$ and $\sin\theta_{ij}$ is stored with an identifying flag, instead of both $\cos\theta_{ij}$ and $\sin\theta_{ij}$. There are consequent losses both in speed and accuracy. If the eigenvectors are not required then there is no need to store the cosines and sines, and the Givens' process may become quite attractive when a tape backing store is being used. The technique described in this paper may well have counterparts in other transformations of a similar nature.

Acknowledgements

We wish to thank Dr. L. Fox for reading the manuscript of this paper and making some helpful suggestions. One of us (J. H. W.) has contributed to the work as part of the Research Programme of the National Physical Laboratory and the paper is published by permission of the Director of the Laboratory.

References

Brooker, R. A., Richards, B., Berg, E., and Kerr, R. H. (1959). *The Manchester Mercury Autocode System*, The Computing Machine Laboratory, University of Manchester.

Givens, W. (1954). *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, Oak Ridge National Laboratory Report, No. 1574.

Householder, A. S., and Bauer, F. L. (1959). "On Certain Methods for Expanding the Characteristic Polynomial," *Numerische Mathematik*, 1 Band, 1 Heft, p. 29.

Lanczos, C. (1950). "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," *Journal of Research of the National Bureau of Standards*, Vol. 45, p. 255.

Wilkinson, J. H. (1960). "Householder's Method for the Solution of the Algebraic Eigenproblem," *The Computer Journal*, Vol. 3, p. 23.

Correspondence

To the Editor,
The Computer Journal.

Sir,

A few comments on the significance of Allen's useful approximation (Allen, 1959) may be helpful to other readers. Given an empirical function $f(x, y)$, the problem is to fit it numerically by finding functions $a_r(x)$ and $b_r(y)$ such that

$$f(x, y) \simeq \sum_{r=1}^{n} a_r(x) b_r(y).$$

If the variables are defined over a rectangular lattice, we may write this in matrix notation $F \simeq AB'$ where A and B each have n columns. The mathematical interest of the problem is that a solution can be found, as Allen shows, in terms of the eigenvectors of FF' such that $A'A$ and $B'B$ are simultaneously diagonal. This gives the hint that a *least squares fit* of F to AB' has been obtained, though this is nowhere stated in the paper.

Allen fits FF' to AA', and the significance of this is obscure. If, instead, we look for a least squares fit of F to AB' from the start, the eigenvalue equation is very simply obtained. Thus, differentiating the sum of squares of the elements of $F - AB'$ with respect to the elements of A and B gives

$$(F - AB')B = 0 \qquad (1) \qquad (F' - BA')A = 0 \qquad (2)$$

from which we immediately obtain the eigenvalue equation

$$FF'A = A(B'B)(A'A).$$

By choosing $B'B$ to be diagonal without assuming it to be normalized as a unit matrix, we may write the admissible solutions in the form

$$A = VT \qquad (3) \qquad B = F'A \qquad (4)$$

where V is the unitary matrix of eigenvectors of FF' and the matrix

$$T = \begin{bmatrix} I_n \\ 0 \end{bmatrix}$$

selects n of these for inclusion in A. In this form of solution, we are not troubled by factors $\sqrt{\lambda}$ because they have been absorbed by F' in equation (4) above.

There is a further result of importance to the numerical analyst, pointed out to me independently by Mr. E. D. Farmer and Dr. D. P. Jenkins. By considering the trace of $(F - AB')(F' - BA')$, and denoting the complete diagonal eigenvalue matrix by Λ, it is not difficult to show that the *sum of the squared errors* in the fit is given by

$$Tr(\Lambda - TT'\Lambda),$$

which is simply the *sum of the omitted eigenvalues.*

Malvern. Yours faithfully,
17 April 1961. P. M. Woodward.

Reference

Allen, C. D. (1959). "A Method for the Reduction of Empirical Multi-Variable Functions," *The Computer Journal*, Vol. 1, p. 196.