Householder’s Method for the Solution of the Algebraic Eigenproblem

By J. H. Wilkinson

In this paper a practical procedure is described for the solution of the eigenproblem based on a method due to Householder. It is claimed that the technique described is the most satisfactory of known methods for general symmetric matrices and may have considerable advantages for unsymmetric matrices. The details of the computation for a simple numerical example are given, and also the results obtained for a 14 × 14 matrix using the ACE computer.

Introduction

Two of the most successful methods for finding the eigenvalues and eigenvectors of a general symmetric matrix, those of Lanczos (1950) and Givens (1954) are based on the reduction of the matrix to triple-diagonal form. This is effected by means of a similarity transformation with an orthogonal matrix. In the Givens method the orthogonal matrix is obtained as the product of a number of simple orthogonal matrices known as plane rotations. Recently Householder (1959) has described an alternative method of reduction to triple-diagonal form in which the orthogonal matrix is obtained as the product of a number of simple orthogonal matrices which are not plane rotations.

The method does not seem to have attracted the attention it deserves. In our experience it is the fastest and most accurate of known methods for solving the eigenproblem for a symmetric matrix of general form. Further, the program is somewhat simpler and the storage requirements less than for either the Givens or the Lanczos method.

The Elementary Orthogonal Transformations

Householder suggested the use of orthogonal matrices, \( P \), of the form defined by

\[
P = I - 2ww^T,
\]

where \( w \) is a column vector such that

\[
w^Tw = 1.
\]

The matrix \( P \) is symmetric and we have

\[
P^TP = (I - 2ww^T)(I - 2ww^T)
= I - 4ww^T + 4w(w^Tw)w^T
= I - 4ww^T + 4ww^T
= I
\]

so that \( P \) is also orthogonal.

We define \( w_r \) to be a vector with its first \((r - 1)\) components equal to zero, so that

\[
w_r^T = (0, 0, \ldots, 0, x_r, x_{r+1}, \ldots, x_n),
\]

and \( P_r \) to be a \( P \) matrix with \( w = w_r \). From the relation (2) we have

\[
x^2 + x_{r+1}^2 + \ldots + x_n^2 = 1.
\]
Details of the Transformations

We name the elements of the matrix $A$ of order four as shown below.

$$A = \begin{bmatrix}
a_1 & b_1 & c_1 & d_1 \\
b_1 & b_2 & c_2 & d_2 \\
c_1 & c_2 & c_3 & d_3 \\
d_1 & d_2 & d_3 & d_4
\end{bmatrix}$$

We wish to determine $P_2$ such that $P_2AP_2$ has zero elements in position (1, 3), (1, 4) and, therefore, (3, 1) and (4, 1). We define the corresponding $w_2$ by

$$w_2^T = (0, x_2, x_3, x_4)$$

so that we have

$$x_2^2 + x_3^2 + x_4^2 = 1.$$  

(8)

The first row of any matrix is unaltered by multiplication on the left by $P_2$, so that $P_2AP_2$ will have zeros in positions 1, 3 and 1, 4, if, and only if, $AP_2$ has zero in these positions. We must, therefore, choose $w_2$ so that this condition is satisfied. Now we have

$$AP_2 = A - 2Aw_2w_2^T$$

and if we write

$$Aw_2 = p$$

where

$$p^T = (p_1, p_2, p_3, p_4)$$

then the elements in the first row of $AP_2$ are

$$a_1, \ b_1 - 2p_1x_2, \ c_1 - 2p_1x_3, \ d_1 - 2p_1x_4$$

where $p_1$ is given by

$$p_1 = b_1x_2 + c_1x_3 + d_1x_4.$$  

(12)

We must have

$$c_1 - 2p_1x_3 = 0 \quad \text{and} \quad d_1 - 2p_1x_4 = 0.$$  

(13)

We must also have

$$b_1 - 2p_1x_2 = \pm S^{1/2}$$  

(14)

where

$$S = b_1^2 + c_1^2 + d_1^2$$  

(15)

since the sum of the squares of the elements in any row must be invariant. Multiplying (14) by $x_2$ and equations (13) by $x_3$ and $x_4$ we have

$$p_1 = 2p_1(x_2^2 + x_3^2 + x_4^2) = \pm x_2S^{1/2},$$

and since

$$x_2^2 + x_3^2 + x_4^2 = 1,$$

$$p_1 = \mp x_2S^{1/2}.$$  

(16)

Equation (14) therefore gives

$$b_1 + 2x_2S^{1/2} = \pm S^{1/2}$$

and

$$x_2^2 = \frac{1}{2} \left[ 1 \mp \frac{b_1}{S^{1/2}} \right].$$  

(17)

From equations (13) and (16) we have

$$x_3 = \mp \frac{c_1}{2x_2S^{1/2}}.$$  

(18)

where the upper and lower signs in (14), (17), (18) and (19) go together. The form of the equations in the general case for a matrix of order $n$ is quite clear. $x_2$ is determined by an equation of the same form as (17), where $S$ is now the sum of the squares of all elements in the first row other than the first, while $x_3, \ldots, x_n$ are determined by equations of type (18). Whatever the order of the matrix there will be two square roots only to be computed, one for $S^{1/2}$ and the other to determine $x_3$ from $x_2^2$. The elements $x_3, \ldots, x_n$ do not require further square roots.

Practical Considerations

If we are to obtain accurate results it is essential that the calculated matrix $P_2$ should be as accurately orthogonal as possible, otherwise we will not be performing a similarity transformation. The calculation of $x_3$ and $x_4$ requires a division by $x_2$. This suggests that it would be better to choose the sign in equation (17) so that $x_2^2$ is as large as possible. We have therefore

$$x_2^2 = \frac{1}{2} \left[ 1 + \frac{b_1 \text{sgn} b_1}{S^{1/2}} \right].$$  

(20)

This avoids any cancellation in the calculation of $x_2$ and we have

$$x_2^2 > \frac{1}{2}.$$  

If we take the other sign, then when $c_1$ and $d_1$ are small compared with $b_1$, $x_2^2$ is small and we have a loss of accuracy whether we work in fixed-point or floating-point arithmetic. The sign of $x_2$ is of no importance and therefore we take $x_2$ to be the positive square root. The equations for $x_2$ and $x_4$ are then

$$x_3 = \frac{c_1 \text{sgn} b_1}{2x_2S^{1/2}},$$  

(21)

$$x_4 = \frac{d_1 \text{sgn} b_1}{2x_2S^{1/2}}.$$  

If we use equations (20) and (21) and work in floating-point arithmetic, we will obtain values for the $x_i$ which satisfy the relation (8) very accurately. If, however, we work in fixed-point arithmetic, special care is needed if equation (8) is to be accurately satisfied. If all the quantities $b_1, c_1$ and $d_1$ are small, then the direct use of equations (20) and (21) is unsatisfactory. They should be scaled first by multiplying them by the same power of 2 (or 10), and then the computation performed using the scaled quantities.

On a machine such as ACE where scalar products may be accumulated exactly, $S$ is calculated exactly and then shifted 2 places to the left before taking the square root, where 2s is the maximum permissible shift consistent with $(S \times 2^s)^{1/2}$ being representable by a single-precision number. In equations (20) and (21) we work with $(S \times 2^s)^{1/2}$ and $2^s b_1, 2^s c_1$ and $2^s d_1$.
The transformed element in the (1, 2) position, which we have denoted by \( \beta_2 \), is given by
\[
\beta_2 = -\text{sgn} \, b_s(S^{1/2}). \tag{22}
\]
Here we must, of course, use \( S \) without the scale factor.

We now obtain an explicit expression for \( P_2 A P_2 \) in terms of \( w_2 \) and \( p \). We have
\[
(P_2 A P_2) = (I - 2ww^T)(A)(I - 2ww^T)
= A - 2ww^T A - 2Aww^T + 4w(w^T A w)w^T
= A - 2ww^T A - (w^T A w)w^T
\]
\[
- 2[Aw - w(w^T A w)]w^T
= A - 2ww^T - 2gw^T \tag{23}
\]
where \( q = Aw - (w^T A w)w = p - Kw \)
and \( K = (w^T A w)(w^T p) \), which is a scalar.

Since we know the elements of the first row and column of \( P_2 A P_2 \) (they are \( a_1, \beta_2, 0, 0 \)) we need calculate only the elements of rows 2 and columns 2 to 4. Further, because of symmetry only the elements on and above the diagonal are required.

For the general case of order \( n \), there are \( (n - 1)^2 \) multiplications in the calculation of \((P_2, P_3, \ldots, p_n)\), and \( n(n - 1) \) multiplications in the calculation of \((-2ww^T - 2gw^T)\), taking advantage of symmetry. The rest of the computation requires a number of multiplications which is of order \( n \). The total number of multiplications in the reduction to triple-diagonal form is therefore essentially
\[
2[n^2 + (n - 1)^2 + \ldots + 2^2] \approx \frac{3}{2}n^3. \tag{24}
\]
The number required in the Givens transformations is \( \frac{3}{2}n^2 \), and in the Lanczos, when the re-orthogonalizations are performed, \( 2n^3 \). Further, there are approximately \( 2n \) square roots in Householder’s method compared with \( 2n \) in the Givens.

The storage requirements of the Householder method are also very satisfactory, a total of \( \frac{1}{2}n(n + 1) + n \) words being adequate. The successive \( A^{(r)} \) are overwritten on each other in the obvious way. It is necessary also to calculate and store the elements of \( p \) and \( q \) at each step, but they can always be stored in the same \( n \) storage locations. The elements of the \( w \), may be stored in the positions occupied by the zero elements introduced into \( A^{(1)} \), but to make room for all of them we must store the \( \beta_i \) in a separate position. However, the \( \beta_i \) are formed one by one, and when \( r \) of them have been formed, the vector \( p \) contains only \( (n - r) \) elements. The extra \( n \) storage locations may therefore be used to store the \( \beta_i \) as well as the successive \( p \) and \( q \).

A storage space of \( \frac{1}{2}n(n + 1) + n \) is therefore adequate to store all the intermediate matrices \( A^{(r)} \) and to retain complete information on the \( w \), which will subsequently be required if we wish to calculate the vectors.

**Accuracy of the Computation**

On ACE all scalar products in the sums, \( S \), and the vectors, \( p \), are accumulated exactly and then rounded. Similarly, only one rounding error is made when calculating each element of \( P_P A P_2 \), from the equations of which (23) is typical: this makes it possible to restrict the rounding error to a very low level. It is not our intention to give a detailed error analysis here but the principle is as follows. We denote the eigenvalues of the calculated \( A^{(r)} \) by \( \lambda^{(r)}_i \) so that the \( \lambda^{(1)}_i \) are the true eigenvalues of the original matrix. Suppose we can obtain bounds for \( \lambda^{(r)}_i - \lambda^{(i-1)}_i \). If these are \( \delta_i^{(r)} \), then we have
\[
|\lambda^{(i-1)}_i - \lambda^{(i-1)}_i| < |\lambda^{(i-1)}_i - \lambda^{(2)}_i| + |\lambda^{(2)}_i - \lambda^{(3)}_i| + \ldots + |\lambda^{(i-2)}_i - \lambda^{(i-1)}_i| \leq \delta_i^{(i-1)} + \delta_i^{(2)} + \ldots + \delta_i^{(i-2)} \tag{25}
\]
giving us bounds for the errors in the eigenvalues of the triple-diagonal matrix \( A^{(n-1)} \).

The bounds for \( \lambda^{(r)}_i - \lambda^{(i-1)}_i \) may be calculated as follows. Let \( Q_{r-1} \) be the exact orthogonal matrix which would be derived by the application of our algorithm to the calculated matrix \( A^{(r)} \). We can obtain bounds for differences between the calculated elements of \( \boldsymbol{w}_{r-1} \) and their exact values. From these we may obtain bounds for the elements of \( \mathbf{X} \) defined by
\[
\mathbf{X} = \text{Calculated} \left( \mathbf{P}_{r-1} A^{(r)} \mathbf{P}_{r-1} \right)
- \text{Exact} \left( \mathbf{Q}_{r-1} A^{(r)} \mathbf{Q}_{r-1} \right) \tag{26}
\]
Note that \( \mathbf{Q}_{r-1} \) corresponds to calculated \( A^{(r)} \) and not to \( A^{(r)} \) which would have been obtained from exact computation. The matrix \( \mathbf{X} \) is symmetric since Calculated \( \left( \mathbf{P}_{r-1} A^{(r)} \mathbf{P}_{r-1} \right) \) is taken to be symmetric and Exact \( \left( \mathbf{Q}_{r-1} A^{(r)} \mathbf{Q}_{r-1} \right) \) is obviously symmetric. Now the eigenvalues of Exact \( \left( \mathbf{Q}_{r-1} A^{(r)} \mathbf{Q}_{r-1} \right) \) are \( \lambda^{(r)}_i \) and of Calculated \( \left( \mathbf{P}_{r-1} A^{(r)} \mathbf{P}_{r-1} \right) \) are \( \lambda^{(i-1)}_i \). By Lidskii’s theorem we have
\[
|\lambda^{(i-1)}_i - \lambda^{(i-1)}_i| < \text{maximum eigenvalue of } \mathbf{X}, \tag{27}
\]
and we can obtain a bound for this eigenvalue from our bounds on the elements of \( \mathbf{X} \).

Note that at no stage in the analysis we are concerned with the matrices, \( A^{(r)} \), which would have been derived by exact computation throughout. These matrices may in fact differ completely from the computed \( A^{(r)} \). For suppose at any stage all the elements involved in a sum, \( S \), are small as a result of cancellation. Then although the absolute differences between the corresponding elements of \( A^{(r)} \) and \( A^{(r)} \) may be small, the percentage differences may be very considerable. Now from equations (17), (18) and (19) we see that the elements \( x_{r-1}, x_{r-2}, \ldots, x_n \) are determined as ratios of these small numbers and therefore at this stage the computed transformation, \( P_{r-1} \), may differ substantially from the transformation corresponding to exact computation throughout. The above analysis shows this to be of no consequence. It is concerned only when the difference between the computed transformation and the exact transformation corresponding to computed \( A^{(r)} \). If the computation is performed in the manner described, this difference is always small. This demonstrates the ineffectiveness of an automatic error analysis in which
one assesses at each stage the divergence of the calculated values from those which would have resulted from exact computation.

Completion of the Eigenproblem

The completion of the eigensystem of the tripe-diagonal matrix is treated as in the Givens process and has been discussed elsewhere. (Givens 1954, Wilkinson 1958.) To recover the eigenvectors of the original matrix we have to multiply by the \( P_r \). Thus if \( x \) is an eigenvector of the triple-diagonal matrix the corresponding eigenvector, \( v \), of \( A^{(1)} \) is given by

\[
v = P_2 P_3 \ldots P_n x.
\]

The vector \( v \) is calculated from the relations

\[
\begin{align*}
x_{n} &= P_n^{-1} x \\
x_{n-1} &= P_{n-1} x_{n} \\
\vdots \\
v &= x_2 = P_2 x_3
\end{align*}
\]

so that we do not need to form explicitly the product of the \( P_r \). Again at this stage the volume of computation is about half that in the Givens method and on ACE we have the advantage of accumulating a scalar product when we calculate \( P_r x \). For we have

\[
P_r x = (I - 2w_r w^T_r) x - 2(w^T_r x) w_r.
\]

The Unsymmetric Eigenproblem

An analogous process may be used for unsymmetric matrices, though the transformed matrix is now of almost triangular (sometimes called Hessenberg) form. The \( w_r \) are chosen, as before, so as to produce zeros in the successive rows of the matrix but, since we no longer have symmetry, the transformation does not produce corresponding zeros in the successive columns. If the matrix is complex we may use unitary transformations of an analogous type, the matrices now being of the form

\[
P_r = I - 2w_r w^T_r
\]

with

\[
w^T_r w_r = 1
\]

where \( w_r \) is the complex conjugate vector.

Again the method is closely related to that described by Givens (1958) using plane unitary rotations. The number of multiplications in Householder’s method for unsymmetric matrices is approximately \( \frac{3}{2} n^3 \), and in the Givens transformation \( \frac{5}{2} n^3 \). In general, Householder’s method retains all the advantages over Givens’ method which we have noted in the symmetric case.

In a recent paper (Wilkinson 1959) we compared Givens’ method of reduction to almost triangular form with a simpler transformation, using elementary similarity transformations. The latter required \( \frac{3}{2} n^3 \) multiplications and on DEUCE (and ACE) Givens’ method took as long in single-precision arithmetic as did the similarity transformation in double-precision arithmetic. Householder’s method halves the advantage in speed of the similarity transformation. On examples of orders up to 30 we have found the elementary similarity transformations fully as accurate as the unitary transformations even when single-precision is used for both. However, error analysis (Wilkinson 1960) certainly indicates that the unitary transformations have an advantage in numerical stability over the elementary similarity transformations and for matrices of very high order this may become decisive. If this proves to be true, Householder’s method may well be as important for unsymmetric matrices as it is for symmetric.

Numerical Examples

In Table 1 we give the details of the computation for a matrix of order five. The example has been chosen so that scaling is necessary in the first stage. The sum \( S \) is scaled by the factor 10 to and therefore the elements of the first row of \( A^{(1)} \) have been used with a scale factor of 10 to when calculating \( x_2 \). Scaling by an appropriate power of two rather than 10 would have given greater precision, and in this respect binary computation is more satisfactory than decimal. The eigenvalues of \( A^{(1)} \) and the final triple-diagonal form are given and are in very close agreement.

The ACE program (a single-precision fixed-point program) has been used on a moderate number of matrices. We give for comparison the eigenvalues obtained for the matrix of order 14 used by Brooker (1956) as a test matrix. The eigenvalues obtained were

\[
\begin{align*}
1 & \cdot 33403 & 48369 & 5648 \\
0 & \cdot 46276 & 62026 & 9426 \\
0 & \cdot 26773 & 32979 & 5408 \\
0 & \cdot 23163 & 94839 & 7838 \\
0 & \cdot 17735 & 63338 & 2143 \\
0 & \cdot 17130 & 75618 & 0045 \\
0 & \cdot 16632 & 46020 & 8896 \\
0 & \cdot 14342 & 28761 & 4641 \\
0 & \cdot 12278 & 75231 & 8153 \\
0 & \cdot 10321 & 57624 & 3091 \\
0 & \cdot 09720 & 92161 & 6179 \\
0 & \cdot 08422 & 52705 & 6467 \\
0 & \cdot 07359 & 71188 & 5374 \\
0 & \cdot 06437 & 99133 & 6671
\end{align*}
\]

In order that the error made in conversion from binary to decimal should be negligible, we have given 14 decimals though, in fact, they were stored with 45 bits after the binary point. The maximum error is 2 \( \times \) 4, so that there was virtually no accumulation of rounding error. The least accurate vector was correct to 12 decimal places, and those corresponding to the well-separated eigenvalues, almost to 14 decimals. The technique used for the vectors was that described by Wilkinson (1958).

Acknowledgements

The work described here has been carried out as part of the Research Programme of the National Physical Laboratory and this paper is published by permission of the Director of the Laboratory. The author wishes to thank Dr. M. V. Wilkes and Dr. D. J. Wheeler for reading the manuscript and making a number of valuable suggestions.
Table 1: Example of computation for matrix of order five

\[
\begin{array}{cccccc}
A^{(1)} & \begin{pmatrix}
0.81321 & -0.00013 & 0.00014 & 0.00011 & 0.00021 \\
0.93125 & 0.25567 & 0.41235 & 0.41632 & 0.30697 \\
0.18765 & 0.50632 & 0.36079 & 0.27605 & 0.46322 \\
0.27605 & 0.46322 & 0.41931 & 0.30215 & 0.71349 \\
0.84668 & -0.46431 & 2K = 0.84886 & S = 0.07927 \\
-0.27218 & -0.08562 & 2q_2 = 0.21160 & 2q_3 = 0.05980 \\
-0.21386 & -0.03766 & 2q_4 = 0.10622 & 2q_5 = 0.34226 \\
-0.40827 & -0.00215 & & & \\
\end{pmatrix}
\end{array}
\]

Last 4 rows of \(A^{(2)}\)

\[
\begin{array}{cccc}
0.57378 & 0.24276 & 0.36788 & 0.21362 \\
0.22020 & 0.54802 & 0.42454 & 0.42214 \\
0.32148 & 0.69878 & & \\
0.86477 & 0.53547 & & \\
0.43427 & 0.75972 & 2K = 1.98697 & S = 0.23990 \\
0.25217 & 0.79512 & & S^{1/2} = 0.48980 \\
0.33978 & 0.26136 & -0.35412 & \beta_1 = -0.48980 \\
-0.24877 & -0.05879 & 0.14946 & x_2 = 0.74782 \\
0.89270 & -0.19558 & & 2q_2 = -0.64733 \\
-0.45064 & -0.11983 & -0.24199 & 2q_3 = 0.65865 \\
& & 0.09191 & \\
\end{array}
\]

Last 3 rows of \(A^{(3)}\)

\[
\begin{array}{cccc}
0.06519 & 0.17294 & -0.16450 & E_{1/2} = 0.19371 \\
& & & \beta_4 = -0.44013 \\
& & & x_2 = 0.79691 \\
& & & 2q_2 = -0.17585 \\
& & & 2q_3 = -0.34835 \\
\end{array}
\]

Last 2 rows of \(A^{(4)}\)

\[
\begin{array}{cccc}
0.06519 & 0.17294 & -0.16450 & \end{array}
\]

Elements of triple diagonal matrix, \(C\)

\[
\begin{array}{cccc}
\alpha_1 = 0.81321 & 1.67828 & 1.67829 & \\
\alpha_2 = 0.57378 & 0.00030 & 0.81321 & 0.81321 \\
\alpha_3 = 1.33978 & -0.48980 & 0.41985 & 0.41983 \\
\alpha_4 = 0.06519 & -0.44013 & 0.01520 & 0.01520 \\
\alpha_5 = -0.16450 & 0.17294 & -0.29908 & -0.29907 \\
\end{array}
\]

References


