DIGITAL COMPUTER LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

A NOTE ON THE HOUSEHOLDER METHOD FOR THE REDUCTION OF A SYMMETRIC MATRIX TO A CODIAGONAL FORM

by

C. W. Gear

File No. 500
December 14, 1962

A Note on the Householder Method for the Reduction of a Symmetric Matrix to a Codiagonal Form

Introduction

Rollett and Wilkinson[1] have discussed a rearrangement of the order of calculation of the Givens[2] method for the reduction of a symmetric matrix to a codiagonal form which reduces the number of times that the matrix must be read from a backing store. In their paper they remark that the Householder method (Wilkinson[3]) appears to require twice as many scans of the data, and consequently, for slow backing stores, is slower than Givens method, although taking 1/2 the number of multiplications. This note suggests an organization of the computations of the Householder method which maintains the arithmetic advantage and only requires the same number of scans of the data as the Rollett-Wilkinson method.

Simply stated, the methods consists of calculating during one scan only the two vectors necessary to make the transformation but delaying making the transformation until the matrix is read out for the calculation of the vectors during the next pass.

I. Method

The Householder method involves finding a unit vector

$$w^T = (x_1, x_2, --- , x_n)$$

such that the orthogonal matrix

$$\underline{P}^T = \underline{P} = \underline{I} - 2\,\underline{w}\,\underline{w}^T$$

can be used to transform the matrix A so that $\underline{PAP}$ has zeros in the (1,3); (1,4); (1,5); --- and (1,n) positions. $\underline{w}^T$ can be chosen so that $x_1 = 0$. The process is then repeated on the sub-matrix obtained by deleting the first row and column of PAP. The final result after n - 2 applications is a codiagonal matrix.

The following notation is used:

Lower case letters are data and intermediate results. In particular, a_{ij} are the elements of $\underline{A}$, x_i are the elements of w, and p_i and g_i are the elements of the Wilkinson(3) vectors $\underline{p}$ and $\underline{g}$.

Upper case letters refer to storage locations in the main random access memory. Five banks of storage of n locations each are needed. They are A_i, B_i, C_i, P_i and Q_i. A_i is used to hold the last row of the matrix read from the backing store. B_i is used to put the x_i in after these are calculated. C_i holds the x_i from the last scan. P_i is used to accumulate the elements of $\underline{p}$ as they are formed. [These should be more than single precision since they accumulate an inner product.] Q_i hold the q_i calculated on the last scan. Initially C_i and Q_i must be cleared to zero. (Or better still, terms involving them not calculated on the first scan.)

The calculation steps are given in terms of the storage locations; the contents of the locations are mentioned to aid the reader following the Wilkinson(3) paper. The matrix A is assumed to be in backing store, stored by row in its upper triangular form. The result of one pass is to transform the matrix, reducing the last n - 2 elements of its first row to zero, and to use the storage locations vacated to save the vector $\underline{w}$ for future work. n - 2 applications of the process to successive sub-matrices reduce the matrix to the required form.

The arithmetic and data moving for the r^{th} pass is as follows:

1) Read the r^{th} row:

$$A_j = a_{rj} \quad ; \quad j = r, r+1, \text{--}\ n$$

2) $A'_j = A_j - Q_j C_r - Q_r C_j \quad ; \quad j = r, r+1, \text{--}\ n$

(This step need not be performed when r = 1.)

3) $S = +\sqrt{(A^2_{r+1} + A^2_{r+2} + \text{--} + A^2_n)}$

4) $B_{r+1} = x_{r+1} = +\sqrt{(1/2 + 1/2 \frac{|A_{r+1}|}{S})}$

5) $B_k = x_k = Q\, A_k \quad ; \quad k = r+2,\ r+3,\ \text{--}\ n$

where $Q = \dfrac{\text{sign }(A_{r+1})}{2\, B_{r+1}\, S}$

6) $B_r = -S \text{ sign } (A_{r+1})$ (This is the new $A_{r,r+1}$ element)

7) $P_r = p_r = B_{r+1}\, S \text{ sign } (A_{r+1})$

8) Restore the r^{th} row of the matrix using the contents of $A_r,\ B_{r+1},\ B_{r+2},\ \text{---}\ ,\ B_n$

(The actual value of the r^{th} row is $A_r,\ B_r,\ 0,\ 0,\ \text{---}\ ,\ 0,$ but the codiagonal is being saved in locations $B_1,\ B_2,\ \text{---}\ B_r$ as it is calculated so that there are $r - 1$ locations spare in the r^{th} row to store the $\underline{w}$ vector.)

9) Set $P_r = P_{r+1} = P_{r+2} = \text{-------} = P_n = 0$

10) A. Read the k^{th} row of the matrix into A

i.e. $A'_j = a_{kj} \quad ; \quad j = k,\ k+1,\ \text{--}\ n$

B. Calculate $A'_j = A_j - Q_j C_k - Q_k C_j,\ j=k,\ \text{--}\ n$

This step and the next can be ommitted when $r = 1$.

C. Store $A_k,\ \text{--}\ A_n$ back as the new k^{th} row

i.e. $a_{kj} = A_j \quad ; \quad j = k,\ \text{--}\ n$

D. $P'_k = P_k + \sum_{j=k}^{n_1} B_j A_j$

$P'_j = P_j + B_k A_j \quad ; \quad j = k+1,\ \text{--}\ n$

These products should be accumulated double precision.

E. Do steps 10A to 10D for $k = r+1, r+2, \text{--} n$

11. $T = \sum_{j=r+1}^{n_1}{}' P_j B_j$ (This is the Wilkinson $K = \underline{w}^T \underline{A} \underline{w}$)

12. $Q_j = 2(P_j - TB_j)$; $j = r+1, \text{--} n$

$Q_r = 2 P_r$

(These are double the elements q_j which are used to form the transformed matric.)

$(\underline{PAP})_{ij} = (\underline{A} - 2 \underline{w} \underline{q}^T - 2 \underline{q} \underline{w}^T)_{ij} = a_{ij} - B_i Q_i - B_j Q_i)$

13. Save the B_j for the next step in storage bank C_j

$C_j = B_j$; $j = r+1, \text{--} n$

When passes $r = 1, r = 2, \text{--} r = n - 2$ have been completed, the last two rows, namely elements $a_{n-1,n-1}$, $a_{n-1,n}$ and $a_{n,n}$ must be updated. The latter is in A_n so can be treated immediately with the formula:

$$A'_n = A_n - 2 Q_n C_n$$

This element can be read back and then the $(n-1)^{th}$ row can be read and treated appropriately. If after this has been done, A_n (now containing a_{n-1n}) is stored in B_{n-1}, the co-diagonal is in locations $B_1, B_2, \text{--}, B_{n-1}$ and the diagonal is in the backing store in its original position. The locations originally holding the off diagonal elements now contain the various $\underline{w}$ used in the transformation.

II. Comparison with other methods.

This process takes approximately $2n^3/3$ multiplications, there are no additional multiplications over the usual Householder method due to this organization of calculation. This compares to approximately $4n^3/3$ multiplications in the symmetric Given's method. The number of backing store references is identical to the Rollett-Wilkinson version of Given's method or to methods taking advantages of special characteristics of the backing store to be able to read rows or columns at full speed.[(4)]

This method uses 6n storage locations holding n double precision and 4n single precision numbers in the main store, against 4n in the Rollett-Wilkinson method. For unsymmetric matrices, a simple extension of this technique uses approximately $5n^3/3$ multiplications, against approximately $10n^3/3$ for Givens, but it now requires 7n storage locations in the main memory since it is now necessary to calculate both $\underline{p} = \underline{A}\,\underline{w}^T$ and $\underline{v} = \underline{w}\,\underline{A}$ as A is unsymmetric, although only one requires a double precision bank of storage, since now a full row of A is read each time so that each element of $\underline{p}$ can be calculated double precision and then rounded to single precision before being stored.

References

1. Rollet, J.S. and Wilkinson, J.H., "A Efficient Scheme for the Co-diagonalization of a Symmetric Matrix by Given's Method in a Computer with a Two-level Store", Comp. Journ., Vol 4 #2, pp. 177-180, 1961.

2. Givens, W., "Numerical Computation of the Characteristic Values of a Real Symmetric Matrix", Oak Ridge National Laboratory Report #1574.

3. Wilkinson, J.H., "Householder's Method for the Solution of the Algebraic Eigenproblem", Comp. Journ., Vol 3 #1, pp. 23-27, 1960.

4. Gear, C.W., University of Illinois Automatic Computer (Illiac) Library Program M20, "Eigenvalues of a Symmetric Matrix by Given's Method", 1957.