A Comparison of Fortran Subroutines for Calculating Latent Roots and Vectors

By D. N. SPARKS and A. D. TODD

Audits of Great Britain Ltd. *Rothamsted Experimental Station*

SUMMARY

This paper compares the accuracy, speed and storage requirements of Algorithm AS 60, published in this issue, with those of the algorithm given by Ortega (1967).

Keywords: LATENT ROOTS AND VECTORS; EIGENVALUES; QL TECHNIQUE; TRIDIAGONALIZATION

1. INTRODUCTION

THE algorithm given by Ortega (1967) is widely available and in common use. As such it presents a suitable yardstick against which to compare Algorithm AS 60 Sparks and Todd (1973). Both algorithms use the Householder method to reduce a real symmetric matrix to tridiagonal form. Ortega then uses the Givens' Sturms Sequence method to find the latent roots and a procedure due to Wilkinson (1965) for finding the vectors. Algorithm AS 60 uses the *QL* technique for finding both roots and vectors. The two subroutines on which Algorithm AS 60 is based are the Algol procedures TRED2 given in Martin *et al.* (1968) and TQL2 given in Bowdler *et al.* (1968) respectively. [These procedures have now appeared in book form: Wilkinson and Reinsch (1971).]

The accuracy, speed and storage requirements of the algorithms were compared on an ICL 4/70. In addition some of the accuracy comparisons were also made on an IBM 360/50 and a KDF 9.

2. TEST MATRICES

Several different matrices were used to test the algorithms. Details of these are as follows:

The two matrices **A** and **B** given in Martin *et al.* (1968) were used since they were used for testing the original Algol procedures.

$$\mathbf{A} = \begin{pmatrix} 10 & 1 & 2 & 3 & 4 \\ 1 & 9 & -1 & 2 & 3 \\ 2 & -1 & 7 & 3 & -5 \\ 3 & 2 & 3 & 12 & -1 \\ 4 & 3 & -5 & -1 & 15 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 5 & 1 & -2 & 0 & -2 & 5 \\ 1 & 6 & -3 & 2 & 0 & 6 \\ -2 & -3 & 8 & -5 & -6 & 0 \\ 0 & 2 & -5 & 5 & 1 & -2 \\ -2 & 0 & -6 & 1 & 6 & -3 \\ 5 & 6 & 0 & -2 & -3 & -8 \end{pmatrix}.$$

The two matrices $\mathbf{W}^+_{2n+1}$ and $\mathbf{W}^-_{2n+1}$ given in Wilkinson (1965) are known to have coincident or close latent roots. $\mathbf{W}^+_{2n+1}$ has

diagonal elements $\alpha_i = n+1-i, \quad i = 1, \ldots, n+1,$

$= i-n-1, \quad i = n+2, \ldots, 2n+1,$

off-diagonal elements $\beta_i = 1, \quad i = 1, \ldots, 2n$

and all other elements zero.

$\mathbf{W}^-_{2n+1}$ is similar except that

$$\alpha_i = n+1-i \quad \text{for } i = 1, \ldots, 2n+1.$$

Extensive tests were made on the set of matrices sometimes known as Pei matrices (Pei, 1962). These are defined as $\mathbf{P}_n = (P_{ij})$, where

$$p_{ij} = K+L \quad \text{for } i = j,$$

and

$$P_{ij} = K \quad \text{for } i \neq j.$$

When K is non-zero it can easily be shown that $\mathbf{P}_n$ has $n-1$ coincident latent roots $\lambda_i = L$ and one latent root $\lambda_n = L+Kn$. The latent vector $\mathbf{e}_n$ corresponding to λ_n is equal to $(1/\sqrt{n}, \ldots, 1/\sqrt{n})$ and the other $n-1$ latent vectors are orthogonal to $\mathbf{e}_n$ satisfying $\sum_{i=1}^n e_{ji} = 0$. When K is zero all the latent roots coincide and are equal to L.

Three matrices were used to test the performance of the algorithms when the latent roots are small. The tridiagonal matrix $\mathbf{Y}_n$ has

diagonal elements $\alpha_i = 2, \quad i = 1, \ldots, n$

and

off-diagonal elements $\beta_i = 1, \quad i = 1, \ldots, n-1.$

The latent roots are $\lambda_i = 4\sin^2\{\pi i/2(n+1)\}, \quad i = 1, \ldots, n$

with corresponding vectors $\mathbf{e}_i = (e_{1i}, \ldots, e_{ni})$, where

$$\mathbf{e}_{ki} = \mu_{ki} \Big/ \sqrt{\left[\sum_{i=1}^{n} \mu_{ki}^2\right]}$$

and

$$\mu_{ki} = [-1]^{i+1} \sin\{k\pi i/(n+1)\}.$$

As stated in Wilkinson (1965), this type of matrix has small latent roots when n is large.

To examine the results when the matrix elements vary widely in magnitude, the two matrices $\mathbf{X}$ and $\bar{\mathbf{X}}$ given by Martin *et al.* (1968) were used. These are defined by

$$x_{ii} = 10^{2(i-1)}, \quad i = 1, \ldots, 7,$$

$$x_{i+1,i} = x_{i,i+1} = 10^{2i-1}, \quad i = 1, \ldots, 6,$$

$$x_{ij} = 0 \quad \text{otherwise}$$

and

$$\bar{x}_{ii} = 10^{14-2i}, \quad i = 1, \ldots, 7,$$

$$\bar{x}_{i+1,i} = \bar{x}_{i,i+1} = 10^{13-2i}, \quad i = 1, \ldots, 6,$$

$$\bar{x}_{ij} = 0 \quad \text{otherwise.}$$

The latent roots of $\mathbf{X}$ and $\bar{\mathbf{X}}$ are the same, and if $\mathbf{e}_i = (e_{1i}, \ldots, e_{ni})$ is the latent vector corresponding to a root λ_i of $\mathbf{X}$ then $\bar{\mathbf{e}}_i = (e_{ni}, \ldots, e_{1i})$ is the vector corresponding to λ_i for $\bar{\mathbf{X}}$.

3. Accuracy

The results of tests on a KDF 9 computer with the original Algol procedures are given in Bowdler *et al.* (1968) and Martin *et al.* (1968). Algorithm AS 60 itself produced identical results on a KDF 9. The matrices $\mathbf{A}$ and $\mathbf{B}$ were tested on the IBM and ICL machines using single-precision (32 bit) arithmetic but the results were accurate only to four or five decimal places. All subsequent tests were therefore done using double-precision (64 bit) arithmetic.

3.1. *Coincident or Close Latent Roots*

Both algorithms calculated the latent roots of test matrices with pathologically close roots to a high degree of accuracy. The results for $\mathbf{W}_{21}^{+}$ and $\mathbf{W}_{21}^{-}$ agreed with those given in Wilkinson (1965). However, the two algorithms treated the latent vectors differently. For Algorithm AS 60 the calculated latent vectors were always very accurately orthonormal. Ortega's subroutine does not calculate orthogonal vectors corresponding to the multiple roots, but produces a set of independent vectors that can be orthogonalized by the Gram–Schmit process.

Tests were also made on the set of Pei matrices, using various sizes up to order 42. In all cases Algorithm AS 60 gave the expected latent roots and vectors. Ortega's subroutine gave the correct latent roots but was occasionally terminated by a run time error while calculating the vectors. This could be overcome by minor coding changes.

3.2. *Small Latent Roots*

Various orders of the tridiagonal matrix $\mathbf{Y}_n$ up to 54×54 were used to test both algorithms. In all cases the results corresponded to the theoretical values.

Table 1

Latent roots of test matrices $\mathbf{X}$ *and* $\bar{\mathbf{X}}$

Ortega's routine —$\mathbf{X}$ *and* $\bar{\mathbf{X}}$ *LRVT*—$\mathbf{X}$	*LRVT*—$\bar{\mathbf{X}}$
$-9{\cdot}4634741565\ 10^{8}$	$-9{\cdot}4634741565\ 10^{8}$
$-9{\cdot}4634691971\ 10^{2}$	$-9{\cdot}4634674877\ 10^{2}$
$0{\cdot}99989902019$	$1{\cdot}0000517503$
$1{\cdot}0463372148\ 10^{3}$	$1{\cdot}0463373843\ 10^{3}$
$1{\cdot}0098990302\ 10^{6}$	$1{\cdot}0098990302\ 10^{6}$
$1{\cdot}0463377127\ 10^{9}$	$1{\cdot}0463377127\ 10^{9}$
$1{\cdot}0100000098\ 10^{12}$	$1{\cdot}0100000098\ 10^{12}$

In the case of tridiagonal matrices with diagonal elements varying widely in magnitude, it has been shown by Bowdler *et al.* (1968) that they should be presented with the largest elements towards the bottom right-hand corner. To show the importance of this reordering, the matrices $\mathbf{X}$ and $\bar{\mathbf{X}}$ were used both with Ortega's subroutine and with *LRVT* of Algorithm AS 60. The results were as expected except for the roots and vectors of $\bar{\mathbf{X}}$ calculated by *LRVT*. The results are shown in Table 1.

An alternative Algol procedure *IMTQL2*, in which it is no longer important to reorder the diagonal elements of the tridiagonal matrix, is given by Martin and Wilkinson (1968).

4. Forming and Reconstructing a Matrix

Suppose we have n scalars ρ_i, $i = 1, \ldots, n$, and n orthogonal row vectors $\mathbf{u}_i$, $i = 1, \ldots, n$; then the matrix

$$\mathbf{A} = \sum_{i=1}^{n} \rho_i \mathbf{u}_i' \mathbf{u}_i$$

has latent roots ρ_i and corresponding latent vectors $\mathbf{u}_i$. Using values of n less than 10, matrices of this form were tested and both algorithms gave the correct results.

Now suppose we have the latent roots λ_i and vectors $\mathbf{e}_i$ of a matrix $\mathbf{A}$, then matrix $\mathbf{B} = \sum_i \lambda_i \mathbf{e}_i' \mathbf{e}_i$ is equal to $\mathbf{A}$. The reconstructing of a matrix $\mathbf{A}$ is a good test of the latent roots and vectors when $\mathbf{A}$ is non-singular. Extensive tests of this kind were made with matrices whose elements were obtained from a random number generator. The results were satisfactory.

5. Timing

The investigation into the time taken by the two algorithms was done on an ICL 4/70 computer using the Watfor compiler. Nineteen matrices of various orders up to $n = 52$ were generated using a random number generator and the time taken to calculate the latent roots and vectors was found. Since Ortega's subroutine offers the facility of calculating a subset of the latent roots and vectors, the time taken to calculate a proportion, p, of them was also found.

In Fig. 1 $\log_{10}$ (time) is plotted against $\log_{10} n$ for Algorithm AS 60, and Ortega's subroutine with $p = 1$, $\frac{1}{2}$ and $\frac{1}{4}$.

6. Storage

The amount of core store required to run a program on the ICL 4/70 was divided into three parts, namely:

(i) the subroutine;
(ii) the main program and system routines; and
(iii) arrays for holding a matrix and its latent roots and vectors.

In Table 2 the store for each part is given in bytes.

Table 2

Storage requirements for a matrix of order n when n_r latent roots and n_v latent vectors are calculated

Algorithm	(i)	(ii)	(iii)
Algorithm AS 60	5,016	18,112	$16(n^2+n)$ if original matrix is kept, $8(n^2+2n)$ otherwise
Ortega's subroutine	$6{,}108+60n$	23,372	$8(n^2+n_r+nn_v)$

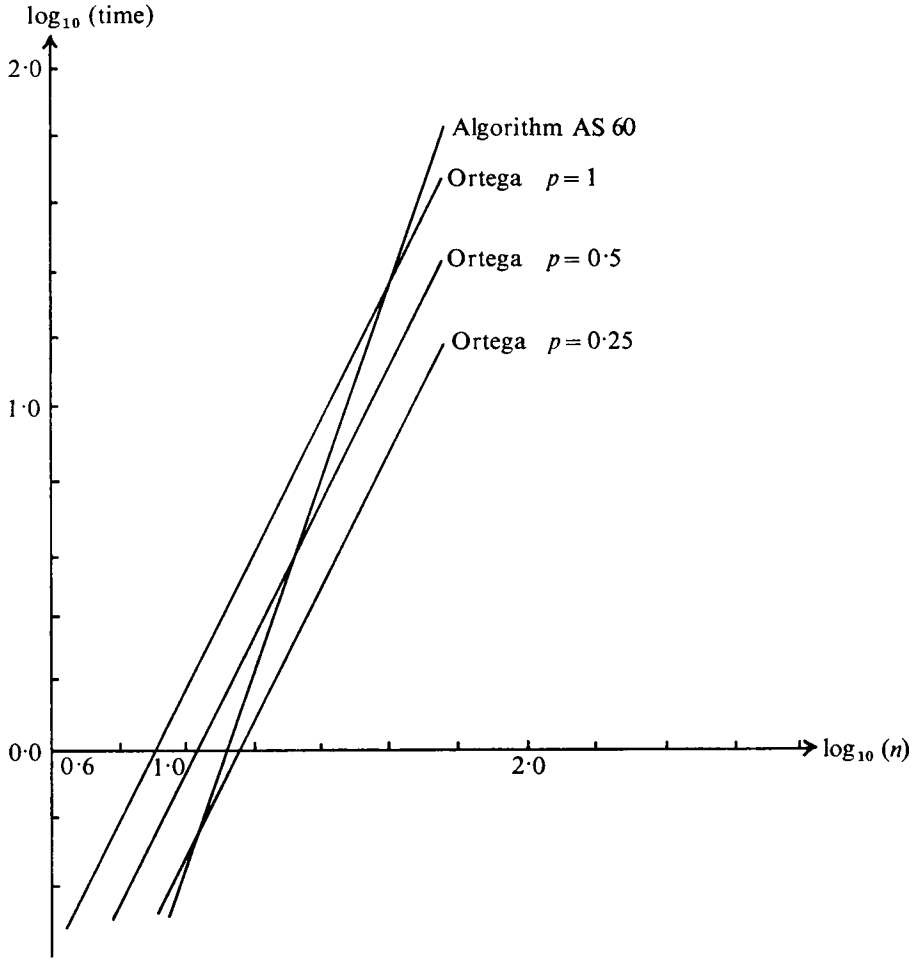

FIG. 1. Time taken to calculate latent roots and vectors.

7. CONCLUSIONS

Algorithm AS 60 always produces accurate results even when there are pathologically close or coincident latent roots. The only noticeable inaccuracies were from matrices whose elements varied widely in value. Ortega's subroutine usually calculates reasonably good results but it fails with some matrices that have coincident or pathologically close latent roots.

When the complete set of latent roots and vectors is required Ortega's subroutine is more efficient than Algorithm AS 60 for matrices of orders exceeding about 48 but much less efficient for orders less than about 35. When only a few of the roots are required Ortega's routine may also be more efficient for orders less than 35. Reference to Fig. 1 will give an indication of which algorithm is the more efficient.

Algorithm AS 60 has been shown to be both extremely accurate and fast. It is recommended as a good general-purpose routine for the calculation of latent roots and vectors.

REFERENCES

BOWDLER, H., MARTIN, R. S., REINSCH, C. and WILKINSON, J. H. (1968). The *QR* and *QL* algorithms for symmetric matrices. *Numer. Math.*, **11**, 293–306.

MARTIN, R. S., REINSCH, C. and WILKINSON, J. H. (1968). Householder's tri-diagonalisation of a symmetric matrix. *Numer. Math.*, **11**, 181–195.

MARTIN, R. S. and WILKINSON, J. H. (1968). The implicit *QL* algorithm. *Numer. Math.*, **12**, 377–383.

ORTEGA, J. (1967). The Givens Householder method for symmetric matrices. In *Mathematical Methods for Digital Computers* (A. Ralston and H. Wilf, eds), Vol. II, pp. 94–115. New York: Wiley.

PEI, M. L. N. (1962). Test matrix for inversion problems. *Commun. Ass. Comp. Mach.*, **5** (10), 508.

SPARKS, D. N. and TODD, A. D. (1973). Algorithm AS 60. Latent roots and vectors of a symmetric matrix. *Appl. Statist.*, **22**, 260–265.

WILKINSON, J. H. (1965). *The Algebraic Eigenvalue Problem.* London: Oxford University Press.

WILKINSON, J. H. and REINSCH, C. (1971). *Handbook for Automatic Computation*, Vol. 2. *Linear Algebra.* Wien–New York: Springer-Verlag.