obtain the next approximation to y. The process may then be repeated.

It will be obvious that the examples given above make extensive use of the restricted forms of the multiplication and substitution programs, and in all the work that has been undertaken with the scheme these restricted forms have been more frequently used than the actual exact routines themselves.

Acknowledgements

The author would like to express his thanks to the Director of the University Mathematical Laboratory for the use of the Titan computer on which all the work has been done, and to Miss J. M. Wright who coded the input routine and the routine to perform the numerical evaluation of the expressions that are manipulated.

References

STRACHEY, C. (1965). "A general purpose macrogenerator", *The Computer Journal*, Vol. 8, p. 225.
BARTON, D. (1966). "Lunar disturbing function", *The Astronomical Journal*, Vol. 71, p. 438.
WOODWARD, P. M., and JENKINS, D. P. (1961). "Atoms and Lists", *The Computer Journal*, Vol. 4, p. 47.

Correspondence

To the Editor,
The Computer Journal.

Sir,

On finding the eigenvalues of real symmetric tridiagonal matrices

By A. J. FOX and F. A. JOHNSON

The Computer Journal has published such significant papers on the Eigenvalue Problem that each new article on this topic is likely to arouse widespread interest. Since I consider the above paper to be misleading I hope you will permit me to make some rather critical remarks.

The Sturm Sequence method (SS) can be made completely reliable and as accurate as the word length permits. The code is brief and involves no *ad hoc* decisions, but it is rather slow. Almost any alternative is faster and several people have concocted rival algorithms. However, the increase of speed was usually bought at the cost of reliability, accuracy, or marked growth of the code.

Criticisms which I make might be considered academic (in the worst sense) were it not for the fact that there already exist published algorithms which are by no means optimal but which are superior to the algorithm proposed by the authors. Perhaps my chief criticism is that even after reading the references which they cite the authors persist in seeing LL^t and QR, not as alternatives, but as supplements to the Sturm Sequence method.

Wilkinson has remarked that "often comparatively minor changes in the details of a practical process have a disproportionate influence on its effectiveness". To any theoretical method there correspond many computer implementations, some careful, some naive. In a comparison of performances who knows which has been used? Although FORTRAN and ALGOL serve well enough to define the order and arrangement of a calculation they are clumsy in treatment of underflow, overflow, and intermediate double precision.

The best of methods when poorly programmed can become a useless algorithm. Surely the following questions should nag any student of comparative algorithmics.

(*a*) Have I implemented the methods properly? Or am I comparing a brilliant realization of one method with a dim caricature of another?
(*b*) Are my comparisons fair? Have I unwittingly loaded the dice (parameters) in favour of one? Are my tests broad enough?

In my opinion these questions did not bother Fox and Johnson sufficiently. Consequently their results are misleading; different realizations have produced quite opposite results.

Let us examine a few aspects of the paper in some detail.

1. *Choice of parameters*

Fox and Johnson remark that the errors in their answers were usually about 10^{-8}. In Table 2 there occurs an error of 10^{-7} (middle eigenvalue should be 0·42773454). This is easily dismissed or overlooked, yet it is a clue. To what?

The authors replace b_i^2 by 0 whenever

$$b_i^2 < \epsilon = 10^{-10}$$

This is equivalent to suppressing b_i whenever

$$|b_i| < \epsilon^{1/2} = 10^{-5}$$

and this can cause changes in the eigenvalues up to 10^{-5}.

Why did the authors not find errors of 10^{-5}? The bound is certainly a realistic one.

The answer is given in a recent result of Kahan: for an $n \times n$ symmetric tridiagonal matrix if

$$\frac{b_n^2}{|a_n - a_{n-1}|} < \epsilon \text{ and } \frac{b_{n-1}^2}{|a_n - a_{n-1}|} < \epsilon$$

then $|\delta\lambda| < 3\epsilon$. Here $\delta\lambda$ is the change produced in any eigenvalue λ by suppressing b_n.

On most of the matrices tested by Fox and Johnson $|a_n - a_{n-1}|$ was eventually greater than 10^{-2} and so their criterion happened to produce errors $< 3\,\epsilon/10^{-2} \doteqdot 10^{-8}$. However, in the case cited above the eigenvalue separation and hence $|a_n - a_{n-1}|$ was 10^{-3} and another iteration would have produced more accuracy. On the other hand for well-separated eigenvalues their criterion would provoke unnecessary iterations.

In the absence of Kahan's result the authors should have either (*a*) set $\epsilon = 10^{-16}$ and guaranteed 8 decimals, or (*b*) stated that with $\epsilon = 10^{-10}$ the user can only be sure of 5 decimals.

Now (*b*) could be disastrous if eigenvectors were desired as well and (*a*) would alter the time comparisons with other methods. See below.

(*continued on* page 420)

Downloaded from https://academic.oup.com/comjnl/article/9/4/344/390168 by guest on 09 April 2024

Correspondence (*continued from p.* 344)

2. *LL^t or secant method?*

Although they do not describe it in these terms the authors' algorithm is essentially as follows. Begin with Sturm Sequence binary chop. At some stage (another *ad hoc* decision) switch to the secant method. Finally when the approximation (origin shift) is close enough to the lowest eigenvalue then the LL^t transformation will deflate the matrix. The computation then proceeds to the next eigenvalue.

What is novel (and unfortunate) is that for much of the calculation the authors are doing the whole LL^t transformation when they only use the value of the characteristic polynomial. This is somewhat wasteful. During the binary chop and secant phases little use is being made of the transformed matrices except when secondary factorization occurs.

The first point then is that their algorithm is essentially the secant method with deflation. Deflation by LL^t turned out to be faster than deflation by QR. Not too surprising. To my mind a true LL^t or QR technique makes use of the transformed matrix at each stage to determine where the next origin shift will be.

The second point is that the secant method has been studied for a long time (Ostrowski, *Solutions of Equations and Systems of Equations*, Academic Press, N.Y., or Ralston, *A First Course in Num. Anal.*, Chap. 8, McGraw-Hill). It can be made into a viable algorithm but only with much care and analysis; without this it has grave defects near root clusters. The convergence rate drops sharply and premature underflow can halt it altogether. At this juncture the authors' algorithm becomes LL^t without shifts; linear convergence and a ratio close to unity.

On a nearly diagonal 30 × 30 matrix of the form

$$\begin{bmatrix} 1 & 10^{-4} & & & & \\ 10^{-4} & -1 & 10^{-4} & & & \\ & 10^{-4} & 1 & 10^{-4} & & \\ & & 10^{-4} & -1 & 10^{-4} & \\ & & & \cdot & \cdot & \cdot \end{bmatrix}$$

the following results occur on a B5500 computer (11 decimals).

		Time
QR (P.A. Businger, Algorithm 253, *Comm. ACM*, Vol. 8 (1965), pp. 217–218)	T_{Bu} =	1·2 sec.
Sturm Sequence (J. H. Wilkinson, *Numer. Math.*, Vol. 4, No. 4 (1962), p. 362)	T_{Wi} =	13 sec.
Fox–Johnson ($\epsilon = 10^{-22}$) *was stopped after*		4 *min.*

All these programs were set to give answers of maximal accuracy. However, if we take the value of ϵ given by Fox and Johnson then their program takes only 0·13 seconds to get answers with errors up to 10^3 in the last place and no warning that this has occurred. In fact two eigenvalues are given multiplicity 14. Of course the other programs obtain answers of this accuracy in comparable times.

The first two algorithms were chosen because they have been published. It is not claimed that they are preferred realizations of their methods, and in fact better implementations do exist. Note that if the above matrix is multiplied by 10^4 then Wilkinson's Sturm Sequence algorithm (being straight ALGOL) results in overflow on many computers.

My main point is that much better (and more compact) algorithms than the authors' are already in the literature. However, I would like to draw attention to the careful QR algorithm recently submitted to the *Communications of the ACM* by Kahan and Varah. It took 0·8 seconds on the above matrix to give maximal accuracy.

On a variety of other matrices the times taken by the above programs were related by $T_{Wi} > 2T_{FJ} > 4T_{Bu}$.

Further comments could be made on

2. Avoiding overflow/underflow in Sturm Sequence algorithms,
4. Implementing the QR method,

and

5. Brevity of program.

Yours faithfully,

BERESFORD PARLETT

P.S. I am grateful to Jim Varah for expert programming.

Mathematics Department,
University of California,
Berkeley.

13 September 1966.

Correction

Editorship of The Computer Journal

We regret that owing to a printer's error on p. xx of the May 1966 issue of this *Journal*, the name of the honorary editor was given as Mr. M. Bridger of Leicester. Mr. Bridger was at that time honorary editor of our associated publication *The Computer Bulletin*. We regret the inconvenience that has been caused by this mistake, and we wish to confirm that contributions to this *Journal*, other than to the Algorithms Supplement, should continue to be addressed to

Mr. E. N. Mutch,
University Mathematical Laboratory,
Corn Exchange Street,
CAMBRIDGE.

Downloaded from https://academic.oup.com/comjnl/article/9/4/344/390168 by guest on 09 April 2024