A one-sided transformation method for the singular value decomposition and algebraic eigenproblem

J. C. Nash

Research Division, Economics Branch, Agriculture Canada, Sir John Carling Building, Ottawa, K1A OC5, Canada

Post-multiplication by plane rotations is employed to orthogonalise the columns of a real matrix. This procedure, when carried out in an ordered fashion, produces a singular value decomposition of the matrix and may be used to solve the eigenproblem of a real symmetric matrix.

(Received October 1973)

Kaiser (1972) has proposed a one-sided transformation method for finding the complete eigensystem of a real symmetric matrix. For certain matrices (see **Table 1**), however, his program proved inadequate, and correction of the inadequacy led to the observation that a similar method would accomplish a singular value decomposition (Golub and Reinsch, 1970) of a real rectangular matrix.

The underlying process for both problems is an orthogonalisation by plane rotations of the columns of a matrix A, which we shall consider $m \times n$ with $m \geq n$. The singular value decomposition of a matrix with $m < n$ is found from that of A^T (transpose of A).

Our aim is to find a matrix V as a product of plane rotations such that:

$$AV = B = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \ldots, \mathbf{b}_n) \tag{1}$$

with the columns of B orthogonal, hence:

$$\mathbf{b}_i^T \mathbf{b}_j = S_i^2 \, \delta_{ij} \tag{2}$$

where δ_{ij} is the Kronecker delta, equal to 1 when $i = j$, 0 otherwise. The S_i may be considered as forming an $n \times n$ diagonal matrix, so that B may be written:

$$B = US \text{ with } U^T U = 1_j \, . \tag{3}$$

Where $j < n$ is the rank of A. Consequently:

$$A = US \, V^T \tag{4}$$

because V, as a product of orthogonal matrices, must itself be orthogonal. Thus, if we can construct V, (4) is the singular value decomposition of A. One difficulty in constructing V is that the ordering of the rotations is indeterminate. This can be effectively limited by requiring that the columns of B decrease in norm from left to right. (However, $\|B\| = \|A\|$, Wilkinson, 1965, page 58). Thus the S_i are monotonic non-increasing.

The plane rotations

Consider the plane rotations acting one at a time to bring A into B. Any one rotation acts only on two columns of the current A matrix, say $\mathbf{x}$ and $\mathbf{y}$. Following Kaiser we can write:

$$(\mathbf{x}, \mathbf{y}) \begin{pmatrix} \cos\emptyset & -\sin\emptyset \\ \sin\emptyset & \cos\emptyset \end{pmatrix} = (\mathbf{X}, \mathbf{Y}) \tag{5}$$

Thus:

$$\begin{aligned} \mathbf{X} &= \mathbf{x}\cos\emptyset + \mathbf{y}\sin\emptyset \\ \mathbf{Y} &= -\mathbf{x}\sin\emptyset + \mathbf{y}\cos\emptyset \end{aligned} \tag{6}$$

If we try to maximise $f = \mathbf{X}^T\mathbf{X}$ with respect to $\emptyset$ we find:

$$\frac{df}{d\emptyset} = 2 \, \mathbf{X}^T\mathbf{Y} = 0 \, . \tag{7}$$

Thus, orthogonality results from this first order condition for a maximum; we further require:

$$\mathbf{X}^T\mathbf{X} - \mathbf{x}^T\mathbf{x} > 0 \tag{8}$$

to ensure the calculation always proceeds towards an ordering of column norms. Defining:

$$P = 2 \, \mathbf{x}^T\mathbf{y} \tag{9a}$$

$$Q = \mathbf{x}^T\mathbf{x} - \mathbf{y}^T\mathbf{y} \tag{9b}$$

$$v = (P^2 + Q^2)^{\frac{1}{2}} \tag{9c}$$

a straightforward calculation suggests:

$$\cos\emptyset = ((v + Q)/(2v))^{\frac{1}{2}} = P/(2v \sin\emptyset) \tag{10a}$$

$$\sin\emptyset = \text{sgn}\,(P) \neq (v - Q)/(2v))^{\frac{1}{2}} = P/(2v\cos\emptyset) \tag{10b}$$

If Q is positive, $\cos\emptyset$ is calculated first from v and Q, and $\sin\emptyset$ from P, v and $\cos\emptyset$. If Q is negative, $\sin\emptyset$ is calculated first. This minimises error by cancellation during a subtraction. An alternative procedure, used in the author's program for the singular value decomposition (but not the eigenproblem) is to set $\emptyset = \frac{\pi}{2}$ whenever Q is negative. This reorders the columns of A so that a subsequent rotation has $Q > 0$. Limited practical experience suggests that there is little to choose between the two methods. The latter choice orders very small singular values, which the slightly faster and apparently more straightforward approach may be unable to do owing to the nature of the convergence criterion. The rotations are performed in a set sequence called a sweep (as in the traditional Jacobi algorithm), each sweep consisting of the $n(n-1)/2$ rotations on the column pairs $(1, 2)\,(1, 3), \ldots, (1, n), (2, 3), (2, 4), \ldots, (2, n), (3, 4), \ldots (3, n), \ldots, (n-1, n)$. The process is iterative since orthogonality between columns established in one rotation may be destroyed in subsequent ones. Convergence is guaranteed by condition (8) which always requires that the column norms become more ordered; in fact:

$$\mathbf{X}^T\mathbf{X} - \mathbf{x}^T\mathbf{x} = \mathbf{y}^T\mathbf{y} - \mathbf{Y}^T\mathbf{Y} = \sin\emptyset\,(\cos\emptyset\, P - \sin\emptyset\, Q) = \tfrac{1}{2}(v - Q) \geq 0 \, . \tag{8a}$$

An algorithm very similar to this, except that the angle calculation does not seem to be constrained to imply condition (8), was suggested by Hestenes (1958) who approached the problem from the point of view of orthogonalisation directly. His algorithm does not appear to have had any reported practical trial. Neither does a similar algorithm reported by Chartres (1962).

As a test for convergence, Kaiser counted how many times in any sweep $\mathbf{x}^T\mathbf{y}$ fell below a tolerance, and stopped when the count reached $n(n-1)/2$. This is also the method used in the present algorithm except that instead of $\mathbf{x}^T\mathbf{y}$, which may be small simply due to small eigen or singular values, the parameter:

$$t = \frac{(\mathbf{x}^T\mathbf{y})^2}{(\mathbf{x}^T\mathbf{x})(\mathbf{y}^T\mathbf{y})} \tag{11}$$

is used (with a different tolerance).

Table 1 Tests of Jacobi-like methods for the eigenproblem using matrices of order $n = 10$.

Matrix[1]		*BASIC version of Jacobi algorithm of Rutishauser* (1966)	*BASIC version of Kaiser's algorithm*	*Present*[2] *method with Rayleigh quotient*	*Present method with Eq.* (14) *for eigenvalue*
Hilbert segment	S	9	4	4	4
$A_{ij} = 1/(i + j - 1)$	R	7·26 E-7	0·128	6·68 E-6	6·91 E-6
	P	0	0·997	2·16 E-6	2·16 E-6
Ding Dong[3]	S	8	11	5	5
$A_{ij} = 0{\cdot}5/(n - i - j + 1{\cdot}5)$	R	2·32 E-6	0·914	5·13 E-6	1·56 E-5
	P	0	1·94 E-7	1·21 E-6	1·21 E-6
Moler $A_{ii} = i$	S	7	6	6	6
$A_{i,\,j\neq i} = \min(i,j) - 2$	R	1·74 E-5	1·40 E-2	4·28 E-5	9·72 E-5
	P	1·94 E-7	5·40 E-3	2·34 E-6	2·34 E-6
Frank	S	8	4	4	4
$A_{ij} = \min(i,j)$	R	2·29 E-5	5·72 E-5	5·34 E-5	1·14 E-4
	P	2·09 E-7	5·70 E-5	1·53 E-6	1·53 E-6
Border $A_{ii} = 1$	S	35	5	2	2
$A_{ni} = A_{in} = 2^{1-i}$	R	1·79 E-6	9·54 E-7	1·43 E-6	2·38 E-6
	P	5·34 E-9	6·55 E-7	6·29 E-7	6·29 E-7
Diagonal	S	1	2	1	1
$A_{ii} = i$	R	0	0	0	3·81 E-6
	P	0	0	0	0
Wilkinson[4]	S	7	6	5	5
$W+$	R	2·32 E-6	4·29 E-6	6·85 E-6	1·57 E-5
	P	1·79 E-7	2·98 E-7	1·85 E-6	1·85 E-6
Wilkinson[4]	S	6	15	4	4
$W-$	R	1·94 E-6	1·07	1·21 E-5	2·26 E-5
	P	4·77 E-7	4·58 E-7	2·13 E-6	2·13 E-6
Ones $A_{ij} = 1$	S	6		2	2
	R	4·65 E-6	FAILS	2·38 E-5	1·10 E-5
	P	0		8·54 E-7	8·54 E-7
Total Time (secs.)		395	391	299	258
Code Length (words)		1,081	938	927	880

S = number of sweeps
R = magnitude of largest element of residual matrix $(AV - V\Lambda)$
P = magnitude of largest inner product $\mathbf{V}_i^T\mathbf{V}_j$ $(i \neq j)$

Notation: 1·23 E-5 = $1{\cdot}23 \times 10^{-5}$
[1]All matrices were calculated on the NOVA computer and punched in six significant figures on paper tape. Timing is for all nine matrices and includes input/output. The same paper tape was used for all four programs. Machine precision = 2^{-22}.
[2]Angle calculation: $\alpha = ((v + Q)/2v)^{\frac{1}{2}}$, $\beta = p/(2\ \alpha v)$
If $Q \geq 0$, $\cos\emptyset = \alpha$, $\sin\emptyset = \beta$; if $Q < 0$, $\cos\emptyset = \beta$, $\sin\emptyset = \alpha$.
[3]The name and matrix were invented by Dr. F. N. Ris of IBM Thomas J. Watson Research Centre, while he and the author were both at Oxford. This Cauchy matrix has few trailing zeros in any elements, so is always represented inexactly on the machine. However, it is very stable under inversion by elimination. In fact, its eigenvalues have the property of clustering at $\pm\pi/2$.
[4]Wilkinson (1965) page 308. For even n the zero diagonal element is omitted.

Whenever t falls below the tolerance or $(\mathbf{x}^T\mathbf{x}) \cdot (\mathbf{y}^T\mathbf{y})$ underflows, a counter, initially set to $n(n - 1)/2$, is decremented by 1, so if it is 0 at the end of any sweep, convergence is assumed. Typically this occurs about the fourth sweep.

Real symmetric matrices

The eigenproblem of real symmetric matrices is solved by applying the same technique when $m = n$. Failure may occur, however, if the matrix is singular, since a null column may appear in the matrix B. In the singular value decomposition this is not important—one of the singular values is zero and the product USV^T is still A. However, in the case of a real symmetric matrix a null column in B, and hence in U, gives only the trivial solution to the eigenproblem

$$A\mathbf{u} = \lambda\mathbf{u} \ . \tag{12}$$

To avoid this possibility we may look at the eigenproblem of the matrix

$$A^1 = A + k\mathbf{1} \tag{13}$$

which has the same eigenvectors as A, with k chosen to make A^1 positive definite. If S_i^1, $i = 1, 2, \ldots, n$, are the singular or eigenvalues of A^1, then the eigenvalues of A are:

$$\lambda_i = S_i^1 - k \ . \tag{14}$$

Alternatively, the Rayleigh quotient, $(\mathbf{u}^T\mathbf{A}\mathbf{u})/(\mathbf{u}^T\mathbf{u})$, may be used to find the eigenvalues of A. This appears to be a slightly more satisfactory procedure than Equation (14) even on a one precision machine, but requires a copy of A to be stored. For positive definite matrices the algorithm is especially attractive.

Comparisons with other algorithms

One of the major motivations for development of this singular value decomposition algorithm at Agriculture Canada was its compactness; the Golub/Reinsch algorithm will not fit easily into the memory of two minicomputers owned by the Department: a Data General NOVA and a Hewlett Packard 9830. Both these machines operate in the BASIC computer language. While access is available to larger machines, cost and turnaround factors make it attractive to use the minis. The algorithm has been built into a larger program which allows a number of regression calculations to be performed on different combinations and permutations of variables loaded into the machine at the beginning of a run (either by hand or paper tape). A number of statistics other than the regression coefficients are produced by the program. The major shortcoming of the minis, apart from size, is the lack of a double precision feature. Thus all the test computations, including residuals and Rayleigh quotients, are performed in single precision.

As a check of program performance, the matrices U^TU, V^TV, and VV^T were formed. In all cases, so far, these have been unit matrices (perhaps augmented by zeros in the case of U^TU when rank $(A) < n$) to within a small multiple of the machine precision. Similarly, the quantities:

$$q_j = \left\{ \frac{\sum_{i=1}^{m} \left(A_{ij} - \sum_{k=1}^{n} U_{ik} S_{kk} V_{jk} \right)^2}{\sum_{i=1}^{m} A_{ij}^2} \right\}^{\frac{1}{2}} \tag{15}$$

were computed, and in every case were a small multiple of the machine precision. When one column of A was null, the numerator of the fraction in Equation (15) was printed and always was extremely close to, or identically zero.

Regression calculations were carried out on the NOVA (machine precision 2^{-22}) using the least squares test data of Golub and Reinsch (1970), some labour statistics given by Longley (1967), and polynomials suggested by Wampler (1970).

For the Golub/Reinsch data, agreement with the published results was obtained to machine precision. For Longley's data, similar agreement between the Golub/Reinsch method and the present algorithm was obtained for both the singular values and regression coefficients (the calculations were run in FORTRAN on a Univac 1108 to allow comparison). For Wampler's polynomials, for which the regression coefficients are known in advance, the author has observed no program which when run in single precision provided results correct to more than one or two digits in all polynomials. The present algorithm performed approximately as well as BASIC versions of Bauer's algorithm (1965; this normally requires double precision to compute residuals for an iterative improvement sequence) and a Forsythe orthogonal polynomial program (Forsythe, 1957), somewhat better than the Golub/Reinsch program (the reason why is unclear) and naturally much better than conventional elimination techniques for regression. Timing comparisons of the FORTRAN programs run with Longley's data on the Univac 1108 show the present algorithm to be slightly faster than that of Golub and Reinsch and the code to occupy about half the space, inclusive of input and output instructions. Data storage requirements of both methods are approximately the same, though differences in approach make it difficult to make exact comparisons; for instance, Golub and Reinsch do not produce the matrix U. In solving the least squares problem (via their program MINFIT):

$$\text{minimise } \|A\mathbf{x} - b\|_2 \tag{16}$$

with the additional condition that $\|\mathbf{x}\|_2$ be minimised if the result of (16) is not unique, they form $U^T\mathbf{b}$ implicitly, which the present program calculates explicitly. Also, as mentioned above, all variables of interest are entered at the beginning of a run for user flexibility in the author's program. This requires some additional storage. It is the author's current opinion that the Golub/Reinsch algorithm will prove more robust and faster for problems in more than 10 regressors, but further experience will be required to confirm this conjecture. One advantage of the present method is that the singular values are automatically ordered.

Table 1 gives comparisons of various methods for the eigenproblem of order 10 real symmetric matrices. In summary, there is the usual trade-off of speed and storage for accuracy. Considerable improvement in accuracy should be obtainable by using double precision accumulation of products, particularly in the calculation of Rayleigh quotients. Similar results to those presented were found for order 4 matrices. The principal timing comparison (for the complete set of 9 matrices) is between the present method and a BASIC version of Rutishauser's Jacobi algorithm, which is efficient primarily for small matrices. Both methods have computing time approximately proportional to n^3 and the present method cannot be recommended for solution of the eigenproblem of matrices much larger than order 10.

Acknowledgements

The author would like to acknowledge the encouragement of Dr. K. J. McKenzie, Dr. F. N. Ris, L. P. Lefkovitch and K. Price and the inspiration of the participants in the Eigensystem Workshop at the Argonne National Laboratory, 25-29 June 1973, particularly Dr. J. H. Wilkinson and Professor G. H. Golub.

References

BAUER, F. L. (1965). *Numer. Math.*, Vol. 7, pp. 338-352.
CHARTRES, B. A. (1962). *The Computer Journal*, Vol. 5, pp. 51-60.
FORSYTH, G. E. (1957). *J. Soc. Ind. Appl. Math.*, Vol. 5, pp. 75-88.
GOLUB, G. H., and REINSCH, C. (1970). *Numer. Math.*, Vol. 14, pp. 403-420. See also Wilkinson and Reinsch (1971).
HESTENES, M. R. (1958). *J. Soc. Ind. Appl. Math.*, Vol. 6, pp. 51-90.
KAISER, H. F. (1972). *The Computer Journal*, Vol. 15, pp. 271-273.
LONGLEY, J. W. (1967). *J. Amer. Stat. Assoc.*, Vol. 62 (319), pp. 819-841.
RUTISHAUSER, H. (1966). *Numer. Math.*, Vol. 9, pp. 1-10.
WAMPLER, R. H. (1970). *J. Amer. Stat. Assoc.*, Vol. 65 (330), pp. 549-565.
WILKINSON, J. H. (1965). *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford.
WILKINSON, J. H., and REINSCH, C. (1971). Linear Algebra, Vol. 2 of *Handbook for Automatic Computation*, Springer, Berlin.