Numer. Math. 56, 627–633 (1989)

Numerische Mathematik

A Note on a One-sided Jacobi Algorithm*

K. Veselić[1] and V. Hari[2]

[1] Lehrgebiet Mathematische Physik Fernuniversität Hagen, D-5800 Hagen, Federal Republic of Germany

[2] Department of Mathematics University of Zagreb p.p. 187, YU-41000 Zagreb, Yugoslavia

Summary. We propose a "one-sided" or "implicit" variant of the Jacobi diagonalization algorithm for positive definite matrices. The variant is based on a previous Cholesky decomposition and currently uses essentially one square array which, on output, contains the matrix of eigenvectors thus reaching the storage economy of the symmetric QL algorithm. The current array is accessed only columnwise which makes the algorithm attractive for various parallelized and/or vectorized implementations. Even on a serial computer our algorithm shows improved efficiency, in particular if the Cholesky step is made with diagonal pivoting. On matrices of order $n=25$–200 our algorithm is about 2–3.5 times slower than QL thus being almost on the halfway between the standard Jacobi and QL algorithms. The previous Cholesky decomposition can be performed with higher precision without extra time or storage costs thus offering considerable gains in accuracy with highly conditioned input matrices.

Subject Classifications: AMS(MOS): 65F15; CR: G1.3.

"One-sided" or "implicit" variants of the Jacobi eigenreduction on real symmetric matrices have been proposed long ago ([4, 11, 12, 13]). One of their advantages is that updated matrices are accessed only by columns which makes the methods attractive for vectorized or parallel computing [3, 5, 6, 15, 16]. We propose here a version based on a previous factorization of the given matrix which we suppose to be positive definite. This idea, too, has its predecessor in [7].

Let A be symmetric and positive definite and let

$$A = LL^t \tag{1}$$

* This work was partly done during the first author's visit to the Department of Mathematics, The University of Tennessee-Knoxville, while participating in the Special Year on Numerical Linear Algebra sponsored by the UTK Departments of Computer Science and Mathematics, and the ORNL Mathematical Sciences Section, Engineering Physics and Mathematics Division as well as during a second author visit to the Fernuniversität Hagen. Both authors gratefully acknowledge the support of the respective institutions

(this may or may not be a Cholesky decomposition). Set

$$\hat{A} = L^t L \tag{2}$$

and find an orthogonal matrix U diagonalizing $\hat{A}$:

$$U^t \hat{A} U = U^t L^t L U = \alpha, \tag{3}$$

where α is diagonal and positive definite. Set

$$S = LU, \quad F = S\alpha^{-\frac{1}{2}}. \tag{4}$$

Then F is obviously orthogonal and it diagonalizes A:

$$F^t A F = \alpha^{-\frac{1}{2}} U^t L^t L L^t L U \alpha^{-\frac{1}{2}} = \alpha^{-\frac{1}{2}} U^t \hat{A}^2 U \alpha^{-\frac{1}{2}} = \alpha^{-\frac{1}{2}} \alpha^2 \alpha^{-\frac{1}{2}} = \alpha. \tag{5}$$

The Jacobi method consists of putting $U = I$ and iterating the step

$$\hat{A} \to R^t \hat{A} R, \quad U \to UR \tag{6}$$

where R's are conveniently chosen plane rotations. Our method sets

$$S := L \tag{7}$$

and iterates only

$$S \to SR. \tag{8}$$

Such process is called one-sided or implicit, the current matrix being given "implicitly" by the factor S. In order to perform the transformation (8) we need the rotation parameter which is determined by the pivot elements of $\hat{A}$

$$\hat{a}_{kk} = s_k^t s_k, \quad \hat{a}_{km} = s_k^t s_m, \quad \hat{a}_{mm} = s_m^t s_m \tag{9}$$

where

$$S = (s_1, \ldots, s_n).$$

Since the diagonal elements can be simply transformed as

$$\hat{a}_{kk} \to \hat{a}_{kk} + t\,\hat{a}_{km}, \quad \hat{a}_{mm} \to \hat{a}_{mm} - t\,\hat{a}_{km}, \tag{10}$$

where t is the tangent of the rotation angle the diagonal elements can be stored and updated separately at a little extra cost. The main amount of work consists

in computing $s_k^t s_m$ in (9) and in the step (8). This makes $5n$ multiplications per rotation (the classical Jacobi process needs $8n$)[4]. Note that the whole computation is made essentially on one square array which, on output, contains the eigenvectors. Thus, our version of the Jacobi algorithm reaches the storage economy of the symmetric QL. Note also that in this way at the end of the process we obtain two sets of eigenvalues: the one contained in the current diagonal and the one obtained by (9) from the final columns of S. The comparison of these two sets may serve as a test of the stability.

The novelty in our method consists in the transition from A to $\hat{A}$ ($\hat{A}$ needs not to be computed!)[5]. In our language [4, 12, 13] put $S:=A$ instead of $S:=L$ in (7) thus diagonalizing implicitly the matrix A^2. Apart of the gains in storage and operational count described above our method avoids the squaring which may cause large relative errors (see our experimental results below). In addition, the transition from A to $\hat{A}$ is in fact a step of the symmetric LR algorithm which usually has non-negligible diagonalization effects.

It is important to note that the stopping criterion should be of the form

$$t_{\max} = \max_{k<m} t_{km} \leqq \varepsilon, \qquad t_{km} = \frac{|\hat{a}_{km}|}{\sqrt{\hat{a}_{kk}\hat{a}_{mm}}}, \tag{11}$$

in other words

$$\max |(F^t F - I)_{km}| \leqq \varepsilon. \tag{12}$$

This guarantees the good orthogonality of the matrix of eigenvectors.

The value ε is usually taken to be, say, 10 macheps (macheps is the machine precision). However, the criterion (11) may fail to stop the process because of rounding errors in computing the scalar product $\hat{a}_{km} = s_k^t s_m$. An additional stopping criterion consists in computing the maximum t_M of all encountered t_{km} during a sweep. It stops, if t_M is less than, say, $\sqrt{\text{macheps}}$ and is decreased by less then 10 times in the next sweep. The value t_M is also used in designing a threshold strategy (we omit the details).

The use of the quantity t_{km} in convergence considerations is further justified by the following interesting property. Set

$$H(A) = \frac{det\, A}{a_{11} a_{22} \ldots a_{nn}}.$$

$H(A)$ is the so-called Hadamard measure of the non-diagonality for a positive definite matrix A. Indeed, one easily sees that

$$0 < H(A) \leqq 1,$$

[4] The operational count is reduced to $3n$, $4n$, respectively if fast rotations are used (see Rath [14] and de Rijk [15])

[5] Cf. [17] where a similar transition was made in a more complicated case of matrix pairs

where the equality sign is taken if and only if A is diagonal. It can be shown that after a Jacobi rotation $H(A)$ is divided by $1-t_{km}^2$. This leads to a "multiplicative" convergence proof for the Jacobi algorithm (cf. [9]).

The choice of the stopping criterion above also guarantees the relative accuracy of the eigenvalues. Indeed, as a consequence of [2], Prop. 2 we have

$$1-n\varepsilon \leqq \frac{\lambda_i}{\hat{a}_{ii}} \leqq 1+n\varepsilon, \tag{13}$$

where λ_i is the eigenvalue associated to $\hat{a}_{ii}$. Of course, this property is useful only if the rounding errors during the process did not spoil the relative accuracy of small eigenvalues (see the numerical results below).

Our implicit version carries efficiency improvements even on a serial computer. We illustrate this on some examples. The experiments were done on an MS-DOS-operated Tandon PCA personal computer with the Intel 80286 and 80287 processors using Turbo Pascal 3.0 with the macheps $\approx 2\cdot 10^{-16}$. We have tested three Jacobi versions:

jac is obtained by the Rutishauser code *jacobi* from [18] by modifying the stopping criterion according to what is said above.

jacf is as jac above but with fast rotations (see Rath [14]).

jacfim consists of the Cholesky step, performed with (optimal) diagonal pivoting and followed by the implicit algorithm, described above. Here fast rotations, combined with the stability improving features of jac were taken over.

The test matrices are the type

$$A=\sum vv^t,$$

where the sum runs over n vectors of length n whose components are random numbers from the interval $(-1/2, 1/2)$. Such matrices have fairly scattered eigenvalues, beginning from rather small ones. The computing times were

n	25	50	100	200
jac	5.2	6.3	–	–
jacf	3.1	3.7	–	–
jacfim	2.5	2.5	2.7	3.4

Here the unit is the computing time needed by the standard QL algorithm (including all eigenvectors) for the same matrix. The void positions mean that in this case either the storage or the real time requirements were too large. No significant differences in the computed eigenvalues were observed.

Thus, although the implicit Jacobi algorithm cannot reach the efficiency of QL, it is clearly superior to the classical version.

The preparatory Cholesky decomposition (with or without pivoting) can serve as a good preconditioner, if it is performed with higher precision. In fact, the condition of the Cholesky factor is the square root of the condition of A.

As a typical example take the matrix

$$A = \begin{pmatrix} 27 & 27 & 27 & 27 & 27 & 27 \\ 27 & 91 & 91 & 91 & 91 & 91 \\ 27 & 91 & 216 & 216 & 216 & 216 \\ 27 & 91 & 216 & 432 & 432 & 432 \\ 27 & 91 & 216 & 432 & 432 & 432 \\ 27 & 91 & 216 & 432 & 432 & 432 \end{pmatrix} + 10^{-8} I,$$

The two smallest eigenvalues of A are equal to 10^{-8}. With diagonal pivoting the implicit matrix is (rounded to 4 decimal places)

$$\hat{A} = \begin{pmatrix} 1424.8565 & 128.8565 & 34.5693 & 5.6607 & 0.0044 & 0.0025 \\ 128.8565 & 128.8565 & 34.5693 & 5.6607 & 0.0000 & 0.0000 \\ 34.5693 & 34.5693 & 57.2980 & 9.3826 & -0.0000 & -0.0000 \\ 5.6607 & 5.6607 & 9.3826 & 18.9890 & 0.0000 & 0.0000 \\ 0.0044 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0025 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix}.$$

Here the small and the large eigenvalues are neatly separated. It is clear that a Jacobi method will work with increased efficiency on such matrices.

We tried here jsmfip with double precision Cholesky decomposition and single precision Jacobi algorithm[6]. The smallest eigenvalues were reproduced with 7 correct decimal places! Omitting the pivoting with the Cholesky step did not change this result. The values obtained by double precision algorithms were not more accurate. On the other hand, the single precision jsm, jsmf, QL obtained no significant digit here whereas single precision jsmfip even monitored the non-definiteness during the Cholesky decomposition.

The Cholesky decomposition with diagonal pivoting is closely related to the QR decomposition with column pivoting preceding the SVD Jacobi algorithm [10]. The counterexample given in [8], Example 6.4.2. shows that the Cholesky step – even with pivoting – does not necessarily detect all small eigenvalues. As a rule, however, the diagonalizing effect of the diagonal pivoting on matrices with high condition number is quite considerable.

If the diagonal pivoting is omitted, dangerous cancellations with the updating formulae (10) can be expected if there are pathologically small eigenvalues. The consequence are uncorrect rotating angles and the slowing down of the convergence. This effect was indeed observed, but is simply cured due to the fact that the condition number of the current matrix S is only the square root of that of A or $\hat{A}$ so that the current implicit diagonals, obtained by (9) are much more accurate[7].

[6] Here Turbo Pascal 4.0 was used with single precision $\approx 10^{-7}$ and double precision $\approx 10^{-16}$

[7] Note that even quite wrong rotation angles cannot ruin a reasonably conditioned matrix S because of the orthogonality of the transformation

Refreshing the current diagonals by (9) after each sweep proved to be quite sufficient in the practice. This point might be of interest if the diagonal pivoting is undesirable with a parallel implementation.

The double precision Cholesky decomposition followed by the single precision implicit Jacobi in jsmfip can be implemented without extra time or storage costs[8]. Indeed, the Cholesky decomposition needs only the double precision upper triangle which is the same amount of storage as the single precision square array needed for the implicit Jacobi algorithm[9]. The appropriate storage management may take some more time. However, the Cholesky decomposition time is anyhow almost negligible in comparison with that for the implicit Jacobi algorithm that follows.

For a general symmetric matrix A a spectral shift λ (computed e.g. by the Gershgorin theorem) will make $A-\lambda I$ positive definite. Then our method is applicable, but the described improvements in accuracy will be lost. Thus, a "shift-free" approach is desirable. This matter will be the subject of a future work.

References

1. Argyris, J.H., Brønlund, O.E.: The natural factor formulation of stiffness for the matrix deplacement method. Comput. Method Appl. Mech. Eng. **40**, 97–119 (1975)
2. Barlow, J., Demmel, J.: Computing accurate eigensystems of scaled diagonally dominant matrices. Courant Institute Techn. Rep. 421. SIAM J. Numer. Anal. 1988 (to be published)
3. Berry, M., Sameh, A.: Parallel algorithms for the singular value and dense symmetric eigenvalue problems. Center for Supercomputing Research and Development, University of Illinois Urbana 1988
4. Chartres, B.A.: Adaptation of the Jacobi method for a computer with magnetic tape backing store. Comput. J. **5**, 51–60 (1962)
5. Eberlein, P.J.: On one-sided Jacobi methods for parallel computation. SIAM J. Alg. Disc. Method **8**, 379–386 (1987)
6. Eberlein, P.J.: On using the Jacobi method on the hypercube. Department of Computer Science, SUNY University at Buffalo preprint 86-23, November 1986
7. Falk, S.: Über halbimplizite und implizite Algorithmen zur endlichen und iterativen Transformationen von Matrizenpaaren. ZAMM **64**, 437–439 (1984)
8. Golub, G.H., van Loan, C.F.: Matrix computation. J. Hopkins University Press, Baltimore 1983
9. Gose, G.: Das Jacobi-Verfahren für $Ax=\lambda Bx$. ZAMM **59**, 93–101 (1979)
10. Hari, V., Veselić, K.: On Jacobi methods for singular value decompositions. SIAM J. Sci. Stat. Comput. **8**, 741–754 (1987)
11. Hestenes, M.R.: Inversion of matrices by biorthogonalization and related results. J. SIAM **6**, 51–90 (1958)
12. Kaiser, H.F.: The JK method: a procedure for finding the eigenvalues and eigenvectors of a real symmetric matrix. Comput. J. **15**, 271–273 (1972)
13. Nash, J.C.: A one-sided transformation method for the singular value decomposition and algebraic eigenvalue problem. Comput. J. **18**, 74–76 (1974)
14. Rath, W.: Fast Givens rotations for orthogonal similarity transformations. Numer. Math. **40**, 47–56 (1982)
15. Rijk P., P. de M., A one sided Jacobi algorithm for computing the singular value decomposition on a vector computer. SIAM J. Sci. Stat. Comput. **10**, 359–371 (1989)

[8] There are other possibilities to get at a more accurate Cholesky factor e.g. for matrices generated by finite elements [1]

[9] Some further low-cost stability improvements consist in the double-precision keeping of the implicit diagonal and in accumulating the scalar products in (9)

16. Sameh, A.H.: On Jacobi and Jacobi-like algorithms for a parallel computer. Math. Comput. **25**, 579–590 (1971)
17. Veselić, K.: An eigenreduction algorithm for definite matrix pairs and its applications to over-damped linear systems. Fernuniversität Hagen 1987
18. Wilkinson, J.H., Reinsch, C.: Handbook of automatic computation II linear algebra, 1. Ed. Berlin Heidelberg New York: Springer 1971

Received May 26, 1989

Note Added in Proof

In the meantime a further extensive analysis of our Jacobi method has been made in the forthcoming paper J. Demmel and K. Veselić, The Jacobi's method is more accurate than QR, Courant Institute preprint 1989, where it is shown that the relative errors of the computed eigenvalues *are not essentially larger than those already produced when initially storing the matrix in the working array*. This property is not shared by any known method, based on previous tridiagonalization (QR, divide-and-conquer etc.). In fact, it appears that *all essential rounding errors are contained in the preparatory Cholesky decomposition, if made with pivoting*. The reasons for this important phenomenon go far beyond those considered in this paper thus complementing them.