JOURNAL OF COMPUTATIONAL PHYSICS 29, 199–207 (1978)

Eigenvector Eigenvalue Theory: The Diagonalization of Symmetric Matrices Using a Least-Squares Optimized Threshold Method

ROBERT R. BIRGE AND LYNN M. HUBBARD

Department of Chemistry, University of California, Riverside, California 92521

Received July 1, 1977; revised December 5, 1977

A least-squares threshold diagonalization algorithm is presented. The salient features of the algorithm include an internally computed series of threshold values which optimize convergence and an input parameter to simplify programmatic control of the precision of the calculated eigenvectors. The optimized threshold method is from 30 to 40% faster than the most efficient Jacobi diagonalization routines and is roughly 20% faster than previous threshold methods. Furthermore, the algorithm requires no allocation of additional work arrays.

1. INTRODUCTION

The diagonalization of Hermitian matrices using a succession of plane rotations was originally conceived by Jacobi [1]. The Jacobi method has the advantage of a relatively simple algorithm which is fail-safe when properly programmed. It has the disadvantage of a time-consuming iterative process and the roundoff error that accompanies the large number of matrix rotations frequently required. The availability of more sophisticated algorithms [2–6], such as the Givens–Householder method [5], has primarily limited the use of the Jacobi method to instances where small matrices of order 40×40 or less are typically encountered or memory limitations prevent the use of more complex routines. Nevertheless, a large number of applications find the Jacobi method to be fully adequate. Furthermore, the use of Jacobi and modified Jacobi diagonalization routines will probably increase with the expanded use of minicomputers where storage limitations, rather than computation times, are of primary importance. It was, in fact, the above circumstances that prompted our development of the least-squares optimized threshold method.

The standard Jacobi method [6] searches for the largest absolute off-diagonal element and performs a plane rotation which annihilates this element. Although this approach assures that a minimum number of rotations are used to achieve a given accuracy, the search is time consuming. Furthermore, the search algorithm, when programmed efficiently, requires more than half of the total diagonalization source code and two work arrays [7]. A simpler alternative is to operate on the off-diagonal elements in a systematic, predetermined order, annihilating those with absolute values larger than the desired threshold. This cyclical approach was originally proposed by Goldstine, Gregory, Forsythe, and Henrici [8, 9]. This method, however, introduced considerable roundoff error. A modification introduced by Pope and

0021-9991/78/0292-0199$02.00/0

Tompkins [10] to correct this problem used a decreasing sequence of threshold values such that only those off-diagonal elements with absolute values larger than the current threshold value initiated a rotation. These authors proved that the threshold method was convergent and demonstrated that the original Jacobi angle (rather than overrelaxed or underrelaxed angles) provided the most rapid convergence. Rutishauser proposed a more sophisticated threshold method as well as an algorithm for calculating the Jacobi rotational parameters that significantly reduces roundoff error [11]. Although we have adopted Rutishauser's roundoff error-reducing modifications, the threshold method presented here is from 15 to 25 % faster than Rutishauser's method. Furthermore, our formalism serves to further reduce roundoff error by reducing the number of rotations necessary to diagonalize a given matrix. (The 30×30 matrix specified in Ref. [11] provides a convenient example. The Rutishauser threshold method requires 2339 rotations to diagonalize this matrix, while the algorithm presented here requires only 1851 rotations to diagonalize this matrix to identical accuracy.)

We present in this paper a threshold algorithm which has been optimized for the diagonalization of small matrices (20×20–40×40). The method is typically 30–40 % faster than the most efficient Jacobi algorithm we could find [7] and requires roughly one-half the machine code. It is typically 20 % faster than previous threshold methods (see above). The salient features of our threshold algorithm include (a) an internally computed series of threshold values which optimize convergence, (b) an input parameter to simplify programmatic control of the accuracy of the eigenvectors, and an overall algorithm that is (c) optimized for both sparse and dense input matrices and (d) designed to minimize roundoff error when high accuracy is required while providing rapid convergence when low to medium accuracy is desired. Features (b) and (d) are particularly useful for self-consistent-field molecular orbital calculations where lower accuracy eigenvectors and eigenvalues are sufficient during early stages of the SCF iteraction, but where maximum accuracy is important during the final stages where the SCF process is near convergence (see Section 4).

A word of caution is necessary before presenting the details of our algorithm. The optimized threshold method cures many, but not all, of the disadvantages of the standard Jacobi method. Eigenvectors are still calculated at an accuracy below that produced by most other diagonalization formalisms although our use of Rutishauser's roundoff error modifications yields a significant improvement over the standard Jacobi procedures. Nevertheless, our experience indicates that for a typical 50×50 matrix, the maximum *guaranteed* accuracy in the eigenvectors is two digits less than the number of significant digits carried in the calculation. The lost significance is due primarily to the roundoff error inherent in the iterative procedure.

2. The Optimized Threshold Algorithm

This section presents the basic algorithm of the optimized threshold method. The actual optimization process is discussed in Section 3, where the final parameters are

presented and our method is compared with conventional Jacobi routines. We assume that the reader is familiar with the standard Jacobi method.

The following algorithm is presented in program flow sequence. The matrix, **A**, is to be diagonalized with the resulting eigenvalues to be formed along the diagonal of **A** with the eigenvectors in corresponding columns of **U**. In matrix notation,

$$\boldsymbol{\lambda} = \mathbf{U}^{\dagger}\mathbf{A}\mathbf{U}, \tag{1}$$

where $\boldsymbol{\lambda}$ is a diagonal matrix of the eigenvalues. Operations are performed such that only the upper triangle of **A** is destroyed.

The Algorithm

(a) *Initialize Variables.* Assign n_t (the optimum rotations per threshold) and the initial threshold parameters $\rho^{(1)}$, $\epsilon^{(1)}$, and $\Delta\rho^{(1)}$ (see Section 3). Adjoin an identity matrix in **U** for formation of the eigenvectors. The maximum threshold parameter, $\rho_{\max}$, and the median absolute magnitude of the nonzero matrix elements in **A**, $a_{\text{mag}}^{(0)}$, are both input parameters and are explained in Section 4.

(b) *Scan Off-diagonal Elements of* **A**. Operate only on the upper triangle in sequence $(a_{1,2}, a_{1,3}, \ldots, a_{N-1,N})$ and if absolute magnitude of the element, $a_{ij}^{(k)}$, is greater than $\epsilon^{(\mu)}$ (the current threshold value), annihilate element via Jacobi rotation [sequences (c)–(f)]. Transfer to (g) upon completion of scan.

(c) *Determine Rotation Parameters* [11].

$$\delta = (a_{jj}^{(k)} - a_{ii}^{(k)})/2a_{ij}^{(k)} \tag{2}$$

$$\tan\theta = \operatorname{sign}(\delta)/[|\delta| + (1 + \delta^2)^{1/2}], \tag{3}$$

$$\cos\theta = [1 + (\tan\theta)^2]^{-1/2}, \tag{4}$$

$$\sin\theta = \cos\theta \tan\theta, \tag{5}$$

$$\tan(\tfrac{1}{2}\theta) = \sin\theta/(1 + \cos\theta). \tag{6}$$

The function "sign (δ)" returns $+1$ for positive or zero δ, -1 for negative δ.

(d) *Perform Plane Rotation of Matrix.* Annihilate off-diagonal element, a_{ij}, using Rutishauser's algorithm [11].

$$a_{ij}^{(k+1)} = 0, \tag{7}$$

$$a_{ii}^{(k+1)} = a_{ii}^{(k)} - (\tan\theta)\, a_{ij}^{(k)}, \tag{8}$$

$$a_{jj}^{(k+1)} = a_{jj}^{(k)} + (\tan\theta)\, a_{ij}^{(k)}, \tag{9}$$

$$a_{il}^{(k+1)} = a_{il}^{(k)} - \sin\theta[a_{jl}^{(k)} + a_{il}^{(k)} \tan(\tfrac{1}{2}\theta)], \tag{10}$$

$$a_{jl}^{(k+1)} = a_{jl}^{(k)} + \sin\theta[a_{il}^{(k)} - a_{jl}^{(k)} \tan(\tfrac{1}{2}\theta)]. \tag{11}$$

The manipulations in Eqs. (10) and (11) are performed over all values of $l\ (\neq i, \neq j)$ from 1 to N such that only the upper diagonal elements of **A** are modified [4].

(e) *Calculate Eigenvectors.*

$$u_{li}^{(k+1)} = u_{li}^{(k)} - \sin\theta[u_{lj}^{(k)} + u_{li}^{(k)} \tan(\tfrac{1}{2}\theta)], \tag{12}$$

$$u_{lj}^{(k+1)} = u_{lj}^{(k)} + \sin\theta[u_{li}^{(k)} - u_{lj}^{(k)} \tan(\tfrac{1}{2}\theta)]. \tag{13}$$

Equations (12) and (13) are calculated for all values of l from 1 to N. The eigenvectors are formed in columns with the appropriate eigenvalue in the corresponding diagonal of **A**. (Note that pseudo-ordering is not performed and that the eigenvalues and eigenvectors must be ordered upon completion of the diagonalization procedure.)

(f) *Update Counter and Branch.*

$$n^{(\mu)} = n^{(\mu)} + 1. \tag{14}$$

If all off-diagonal elements of **A** have not been scanned, return to (b) and continue scan at next sequential off-diagonal element. If a_{ij} is last (sequential) off-diagonal ($a_{N-1,N}$), continue to next step (g).

(g) *Manipulate Threshold Parameters.* If the *first* scan of the matrix at a given threshold has initiated $n^{(\mu)}$ rotations such that $n^{(\mu)} < n_t$, increase the delta threshold parameter, $\Delta\rho$ (Eq. (15)).

$$\Delta\rho^{(\mu+1)} = 2\,\Delta\rho^{(\mu)} \text{ (first scan at } \epsilon^{(\mu)} \text{ and } n^{(\mu)} < n_t). \tag{15}$$

If the number of rotations during any scan (irrespective of the number of previous scans at present threshold) is greater than n_t, retain present threshold [$\epsilon^{(\mu+1)} = \epsilon^{(\mu)}$], set $n^{(\mu+1)}$ to zero, and reinitiate scan at (b). Otherwise, decrease threshold by increasing the value of ρ.

$$\rho^{(\mu+1)} = \rho^{(\mu)} + \Delta\rho^{(\mu)}, \tag{16}$$

$$\epsilon^{(\mu+1)} = a_{\text{mag}}^{(0)}[6^{-\rho(\mu+1)} + 0.2(\rho^{(\mu+1)})^{-6}]. \tag{17}$$

If $\rho^{(\mu+1)}$ is larger than $\rho_{\max}$, the maximum precision threshold parameter, exit diagonalization routine. Otherwise, set $n^{(\mu+1)}$ to zero and reinitiate scan at (b). Note that $\Delta\rho$ is a positive number and that the threshold, ϵ, decreases as ρ, the threshold parameter, increases. The rationale behind Eq. (17) is discussed in Section 4.

3. Optimization of Parameters

Prior to performing an optimization it was necessary to determine the dependencies of the threshold parameters with respect to the order of the matrix N. The following relationships were obtained after the trial-and-error assessment of a number of theoretically reasonable possibilities:

$$\rho^{(1)} = a, \tag{18}$$

$$\Delta\rho^{(1)} = bN^{-1}, \tag{19}$$

$$n_t = c + dN^2, \tag{20}$$

where a, b, c, and d are the parameters to be optimized. The initial threshold parameter, $\rho^{(1)}$, will determine how many matrix rotations will occur at the first threshold level. As the matrix size increases, a given value of $\rho^{(1)}$ will generally increase the number of such rotations with an N^2 dependence. This is what was desired (see discussion of n_t) and hence $\rho^{(1)}$ is defined independent of N. $\Delta\rho$ must decrease with increasing N to prevent "overshooting" the desired number of rotations per threshold. The $1/N$ dependence was obtained by trial and error. The most important parameter, n_t, determines the canonical number of rotations that are to occur at a given threshold setting. Note that the algorithm in Section 2 adjusts $\Delta\rho^{(u)}$ such that $n^{(u)}$ approaches n_t. A high value of n_t will minimize scan time, but increase the total number of rotations required to diagonalize the matrix. This will not only increase calculation time but roundoff error as well. Too low a value for n_t will cause the routine to spend too much time in scanning the matrix and will, at low enough values, simply emulate the standard Jacobi method. We determined that n_t should be proportional to the number of off-diagonal elements and therefore display an N^2 dependence.

The parameters, a, b, c, and d, appearing in Eqs. (18)–(20) were simultaneously optimized for a set of four matrices consisting of two sparse "Hückel" matrices and two dense "random" matrices of order $N = 20$ and $N = 40$. Diagonalization time was the variable to be optimized, and our procedure consisted of an interpolated gradient search over selected ranges in a–d. Trial and error quickly determined the appropriate ranges to study and the possibilities of false minima repeatedly checked. The two sparse matrices were defined by the Kronecker delta operator, $a_{ij} = \delta_{i,j\pm1}$, and represent the Hückel approximation to linear polyenes. Matrices of similar character are frequently encountered in simple, restricted basis set molecular orbital calculations. The two dense matrices were calculated using a pseudorandom number generator with all elements restricted to the range ± 1. (These matrices are available upon request.) Because of the nature of diagonalization by rotation, the computation times were very similar for sparse and dense matrices with the diagonalization of the former generally 5–15 % faster than the latter. The parameters were optimized to minimize the diagonalization time for each matrix and upon averaging gave the following results.

$$\rho^{(1)} = 0.91, \tag{21}$$

$$\Delta\rho^{(1)} = 2.8N^{-1}, \tag{22}$$

$$n_t = 5.7 + 0.009N^2. \tag{23}$$

The "uncertainties" in each parameter are large, but the difference in diagonalization time when using the optimized versus the averaged parameters was rarely more than 8 %. Consequently, the optimized threshold method is not overly sensitive to our choice of parameters with the exception that gross changes in Eqs. (21)–(23) will produce significant deterioration in efficiency.

The optimized threshold method using the above parametrization is compared to an efficient standard Jacobi method in Figs. 1 and 2. The Jacobi method, written by Corbato and Merwin [7], uses a very efficient scan algorithm requiring only N

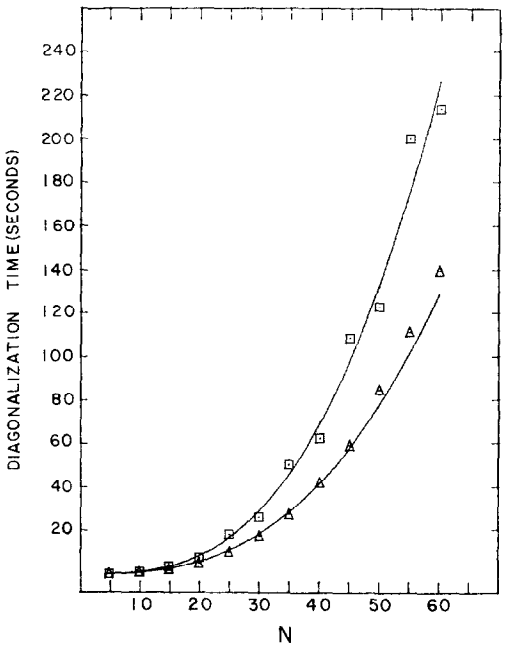

FIG. 1. Comparison of the diagonalization times of sparse ("Hückel") matrices using the standard Jacobi (□) and the optimized threshold (△) methods. Least-squares regression power curves are shown and yield the equations t(msec) $= 1.55N^{2.91}$ (Jacobi) and $t = 1.86N^{2.73}$ (optimized threshold). Calculations were done in Fortran on an HP-3000 minicomputer.

comparisons per search. Calculations were performed using identical "final thresholds" and the pseudo-ordering feature in Corbato and Merwin's routine was removed to increase the Jacobi method's computation speed. In the range $N = 20$ to $N = 60$, the optimized threshold method decreased computation time by 32–40 % over the Jacobi method. The two methods are effectively identical in execution time for $N \leqslant 6$. One reviewer of this manuscript suggested that the simple power curve regressions shown in Figs. 1 and 2 should be replaced with the equation $aN^3 + bN^k$ where $k \sim 2$. Since there are N^2 independent off-diagonal elements each requiring N operations to annihilate, the reviewer noted that both the Jacobi and threshold methods should display a N^3 component in their diagonalization time. Regression curves based on the above equation were calculated and although they consistently provided an improved fit (r^2 was always greater than 0.996), the results were useful only for the sparse matrices (Fig. 1) where the Jacobi method diagonalization time (milliseconds) fits the equation $0.94N^3 + 7.7N^2$, while the threshold method fits the equation $0.62N^3 + 2.5N^2$. However, the regression variable b turned out to be negative for the random matrices which is not very illustrative. Furthermore, the above regression equations might incorrectly suggest that the threshold method is faster both in terms of choosing elements to rotate (the N^2 component) and in performing the necessary rotations (the N^3 component). In fact, the enhanced speed of the threshold method is due solely

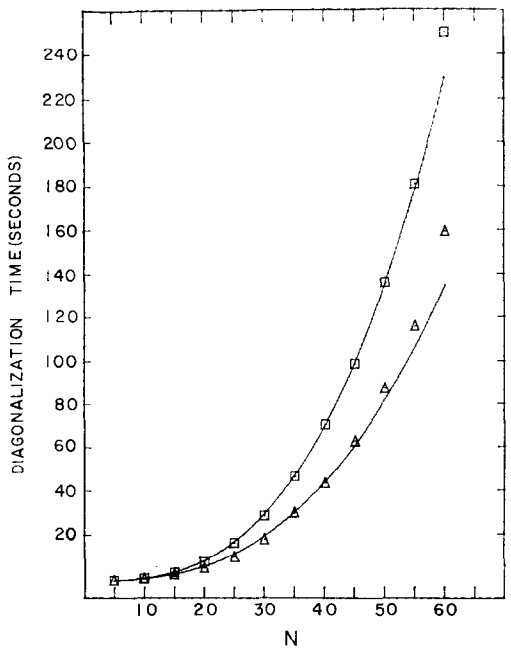

FIG. 2. Comparison of the diagonalization times of dense (random) matrices using the standard Jacobi (□) and the optimized threshold (△) methods. Least-squares regression power curves are shown and yield the equations $t = 1.59N^{2.90}$ (Jacobi) and $t = 2.08N^{2.71}$ (threshold).

to its efficient search algorithm since the time required to perform an individual rotation is essentially identical in the two methods, and the threshold method invariably requires more rotations.

Diagonalization times using Rutishauser's threshold method are not shown in Figs. 1 and 2. Sample calculations indicate that the above procedure requires diagonalization times roughly intermediate to those observed for the optimized threshold method and the Jacobi method of Ref. [7].

4. COMMENTS ON APPLICATIONS

The optimized threshold method was designed not only for speed but also to provide a simple and straightforward means of manipulating the precision with which the eigenvalues and eigenvectors are calculated. The input parameters, $a_{\text{mag}}^{(0)}$ and $\rho_{\max}$, are used to control precision and are discussed in this section.

$a_{\text{mag}}^{(0)}$ is defined as the median of the absolute magnitudes of the nonzero matrix elements in **A**. In practice, this value need not be accurately assigned and an order-of-magnitude estimate is sufficient.

$\rho_{\max}$ determines the final threshold value. Equation (17) was derived so that $\frac{1}{2}\rho_{\max}$ would approximately equal the number of significant digits of eigenvector

precision to the right of the decimal point. The limitations imposed by roundoff error must be taken into account, however. It is a common fallacy of many Jacobi and modified Jacobi routines to continue matrix rotations after the off-diagonal elements have decreased in magnitude by a number, the common logarithm of which is greater than the number of significant digits carried in the calculation. In most such instances, the resulting eigenvectors have actually deteriorated during the subsequent rotations because the roundoff error is larger than the calculated rotational correction. Consequently, there is little to be gained by setting $\rho_{\max}$ larger than twice the number of significant digits carried in the computation. The eigenvalues are calculated to an accuracy greater than that of the eigenvectors. Consequently, the number of significant digits in the eigenvalues is always larger than $\frac{1}{2}\rho_{\max}$.

A useful application of $\rho_{\max}$ can be found in self-consistent-field molecular orbital calculations where a number of iterations, each involving a matrix diagonalization, are encountered in the course of reaching "self-consistence." The number of such iterations is dependent upon both the formalism and the wavefunction, but an empirical rule of thumb is to increase $\rho_{\max}$ by one for each SCF iteration starting at 5 and stopping incrementation when $\rho_{\max}$ reaches the roundoff limit (see above paragraph). If self-consistency is obtained prior to this condition, $\rho_{\max}$ is immediately set to its highest value and iterations are continued to verify self-consistency. This technique has reduced the computation time required for INDO-AFAOS-SCF-MO [12] calculations by 5–12 % over and above the 30–40 % reduction observed by replacing the efficient Jacobi routine [7] with our optimized threshold routine.

Acknowledgments

This work was supported in part by Research Corporation and the Committee on Research, University of California, Riverside. L. H. thanks the National Center for Atmospheric Research, Boulder, Colorado for a UCAR graduate student fellowship. R.R.B. gratefully acknowledges a Regents Faculty Fellowship.

References

1. C. G. J. Jacobi, *J. Reine Angew. Math.* **30** (1846), 51.
2. J. H. Wilkinson, "The Algebraic Eigenvalue Problem," Oxford Univ. Press, London, 1965.
3. Program catalogue, Quantum Chemistry Program Exchange, Indiana University, Bloomington, Ind.
4. B. Carnahan, H. A. Luther, and J. O. Wilkes, "Applied Numerical Methods," John Wiley and Sons, New York, 1969.
5. James Ortega, *in* "Mathematical Methods for Digital Computers," Vol. II, pp. 94–115, (A. Ralston and H. S. Wilf, Eds.) John Wiley and Sons, New York, 1967, and references therein.
6. G. E. Forsythe, *SIAM Rev.* **9** (1967), 489.
7. F. J. Corbato and M. Merwin, A Fortran Program for the Diagonalization of a Real Symmetric Matrix, *in* P. O'D. Offenhartz, "Atomic and Molecular Orbital Theory," pp. 342–344, McGraw–Hill, New York, 1970.

8. R. T. GREGORY, *Math. Tables Aids Comput.* 7 (1953), 215.
9. G. E. FORSYTHE AND P. HENRICI, *Trans. Amer. Math. Soc.* **94** (1960), 1.
10. D. A. POPE AND C. TOMPKINS, *J. Assoc. Comput. Mach.* **4** (1957), 459.
11. H. RUTISHAUSER, *in* "Handbook for Automatic Computation," Vol. II (Linear Algebra), pp. 202–211, (J. H. Wilkinson and C. Reinsch, Eds.) Springer–Verlag, Berlin, 1971.
12. R. R. BIRGE, *J. Amer. Chem. Soc.* **95** (1973), 8241.