

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/2720820>

Numerical Methods For Simultaneous Diagonalization

Article in *SIAM Journal on Matrix Analysis and Applications* · July 1998

DOI: 10.1137/0614062 · Source: CiteSeer

CITATIONS

122

READS

1,145

3 authors, including:



Angelika Bunse-Gerstner

Universität Bremen

86 PUBLICATIONS 1,707 CITATIONS

[SEE PROFILE](#)



Volker Mehrmann

Technische Universität Berlin

461 PUBLICATIONS 8,761 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



MSO4SC -- www.mso4sc.eu [View project](#)



model reduction of transport phenomena [View project](#)

NUMERICAL METHODS FOR SIMULTANEOUS DIAGONALIZATION *

ANGELIKA BUNSE-GERSTNER [†], RALPH BYERS [‡] AND VOLKER MEHRMANN [§]

Abstract. We present a Jacobi-like algorithm for simultaneous diagonalization of commuting pairs of complex normal matrices by unitary similarity transformations. The algorithm uses a sequence of similarity transformations by elementary complex rotations to drive the off-diagonal entries to zero. We show that its asymptotic convergence rate is quadratic and that it is numerically stable. It preserves the special structure of real matrices, quaternion matrices and real symmetric matrices.

Key words. Simultaneous Diagonalization Jacobi Iteration, Eigenvalues, Eigenvectors, Structured Eigenvalue Problem

1. Introduction. Many of the algorithms outlined in [6] require the simultaneous diagonalization of commuting pairs of normal matrices by unitary similarity transformations. Often there are other structures in addition to normality. Examples, include commuting pairs of real symmetric matrices, pairs of Hermitian matrices, real symmetric – real skew symmetric pairs and quaternion pairs. In this paper we point out some of the difficulties associated with simultaneous diagonalization and propose a family of Jacobi-like algorithms for simultaneously diagonalizing commuting normal matrices.

The term “simultaneous diagonalization” is sometimes used in the literature to denote the diagonalization of a definite matrix pencil by congruence, e.g., [43]. Here, we use the term in the classical sense of simultaneous similarity transformations.

To be viable for finite precision computation, a simultaneous diagonalization algorithm must work with both A and B simultaneously. To see how algorithms that violate this principle can fail, consider the family of *diagonalize-one-then-diagonalize-the-other* (DODO) methods suggested by the classic proof that commuting pairs of diagonalizable matrices can be simultaneously diagonalized [30, p. 404]. If $A \in \mathbb{C}^{n \times n}$ and $B \in \mathbb{C}^{n \times n}$ form a pair of commuting normal matrices, the DODO approach uses a conventional algorithm to diagonalize A alone, then performs the same similarity transformation to B . So, for example, if A and B are Hermitian, then one might use \mathbf{CH} from [39], to find a unitary matrix $U \in \mathbb{C}^{n \times n}$ and a diagonal matrix $D \in \mathbb{R}^{n \times n}$ such that $A = U^H D U$. (The superscript H denotes the Hermitian transpose.) Although \mathbf{CH} does not produce the diagonal entries of D in any particular order, it is easy to order the eigenvalues (say) in decreasing algebraic order along the diagonal of D . Then $E := U B U^T$ is block diagonal with the order of the j -th diagonal block equal to the multiplicity of the j -th distinct eigenvalue of A . In particular, if A has distinct eigenvalues, then E is diagonal and the simultaneous diagonalization is complete. In

* SIAM Journal of Matrix Analysis and Applications, Volume 14, Number 4, 927–949, 1993

[†] Fakultät für Mathematik, Universität Bielefeld, Postfach 8640, D-4800 Bielefeld 1. Partial support received from the Forschungsschwerpunkt Mathematisierung Universität Bielefeld, research project: Steuerungsprobleme.

[‡] Department of Mathematics University of Kansas Lawrence, Kansas 66045. Partial support received from the Forschungsschwerpunkt Mathematisierung Universität Bielefeld, research project: Steuerungsprobleme, Science Foundation grant CCR-8820882, KU International Travel Fund 560478, and KU GRF allocations #3758-20-0038 and #3692-20-0038.

[§] Fakultät für Mathematik, Universität Bielefeld, Postfach 8640, D-4800 Bielefeld 1. Partial support received from the Forschungsschwerpunkt Mathematisierung Universität Bielefeld, research project: Steuerungsprobleme.

any case, a subsequent block diagonal similarity transformation can diagonalize E without disturbing D .

Rounding errors destroy this elegant approach. Suppose, for example, rounding errors perturb the commuting pair (A, B) to the nearly commuting pair

$$\tilde{A} = \begin{bmatrix} 1 - \epsilon & 0 & 0 & 0 \\ 0 & 1 + \epsilon & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$\tilde{B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 - \epsilon & 0 \\ 0 & 0 & 0 & 1 + \epsilon \end{bmatrix}.$$

where ϵ is a small quantity that might be caused by rounding error. If $\epsilon \neq 0$, then \tilde{A} and \tilde{B} do not commute. However, if

$$Q = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix},$$

then the off-diagonal elements of $Q^T \tilde{A} Q$ and $Q^T \tilde{B} Q$ are bounded by ϵ .

For $\epsilon \neq 0, \pm 2$, \tilde{A} has distinct eigenvalues, and its modal matrix of eigenvectors

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

is independent of ϵ and unique up to column scaling. Similarly, for $\epsilon \neq 0, \pm 2$, \tilde{B} has distinct eigenvalues and its modal matrix of eigenvectors

$$V = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is independent of ϵ and unique up to column scaling. Unfortunately, the off-diagonal entries of $U^T B U$ and $V^T A V$ are of the order of 1. The DODO method creates a cyclic sequence with period two.

The example suggests that a viable simultaneous diagonalization algorithm must “do something reasonable” when it is applied to a nearly commuting pair of matrices. Perhaps the most natural approach is to require the algorithm to choose a unitary similarity transformation $U \in \mathbb{C}^{n \times n}$ that minimizes some measure of how much $U^H A U$ and $U^H B U$ differ from being diagonal. This is the approach used in [21]. There, the algebraic eigenvalue problem of a single normal matrix is solved by simultaneously diagonalizing the Hermitian and Skew Hermitian parts. It is known to converge locally quadratically under the serial pivot sequence [36]. When applied to a normal

matrix, the norm reducing method of [14] is also an algorithm that simultaneously diagonalizes the Hermitian and Skew-Hermitian part of a normal matrix. A variation of this method using nonunitary normreducing transformations is shown to be globally quadratically convergent for complex normal matrices in [17]. The simultaneous diagonalization algorithms presented below are adaptations and generalizations of [21] and are influenced by [14].

Interest in Jacobi algorithms declined when it was observed that classic and serial Jacobi algorithms for the symmetric eigenvalue problem do more arithmetic operations than more modern techniques [22, 33, 47]. However, with the advent of parallel and vector computers interest in Jacobi's method has revived. Jacobi methods have high inherent parallelism which allows efficient implementations on certain parallel architectures [4, 5, 8, 9, 13, 15, 16, 28, 27, 37]. This has been demonstrated on parallel and vector machines in [2, 3, 15]. Parallel orderings, block Jacobi methods and other techniques create parallel versions of Jacobi's method. It is easy to see that these techniques also apply to the simultaneous diagonalization algorithms presented here. A short discussion of some special techniques for parallel computation along with a more extensive bibliography can be found in [22, §8.5].

Another virtue of the Jacobi method is its favorable rounding error properties. Improved error bounds for perturbed scaled diagonally dominant matrices [1, 10, 25] show that for this class of matrices small relative perturbations in the matrix entries cause only small relative perturbations in the eigenvalues. In some cases, Jacobi's method has better rounding error properties than the QR algorithm [11]. We conjecture that this very favorable error analysis carries over to the simultaneous diagonalization process as well.

We shall use the following notation.

- A superscript T denotes the transpose of a matrix.
- A superscript H denotes the Hermitian or complex conjugate transpose of a matrix.
- The vector 2-norm and its subordinate matrix norm, i.e., the spectral norm, is denoted by $\|\cdot\|_2$.
- The Frobenius norm or Euclidean norm is $\|M\|_F = \sqrt{\text{trace}(M^H M)}$.
- The k -th column of the identity matrix is denoted by e_k .
- The smallest singular value of a matrix $M \in \mathbf{R}^{n \times n}$ is denoted by $\sigma_{\min}(M)$.

2. A Jacobi-like Algorithm. Following the approach of Jacobi [24], the simultaneous diagonalization algorithm consists of a sequence of similarity transformations by plane rotations. A plane rotation in the (i, j) plane, is a unitary matrix $R = R(i, j, c, s)$ of the form

$$(1) \quad R = R(i, j, c, s) = I + (c - 1)e_i e_i^T - \bar{s}e_i e_j^T + s e_j e_i^T + (\bar{c} - 1)e_j e_j^T$$

where $c, s \in \mathbf{C}$ satisfy $|c|^2 + |s|^2 = 1$. For computational convenience, we will often restrict c to be real and non-negative.

A natural measure of the distance of the pair $(A, B) \in \mathbf{C}^{n \times n} \times \mathbf{C}^{n \times n}$ from diagonality is

$$(2) \quad \text{off}_2(A, B) = \sum_{i \neq j} |a_{ij}|^2 + \sum_{i \neq j} |b_{ij}|^2.$$

If $p, q, r, s \in \mathbf{C}$ and $ps - rq \neq 0$, then A and B are diagonal if and only if $(pA + qB)$ and $(rA + sB)$ are diagonal. Hence, for each ordered quadruple $(p, q, r, s) \in \mathbf{C}^4$ such

that $ps - rq \neq 0$, (2) gives rise to an alternate measure

$$(3) \quad \text{off}_{2-pqrs} = \text{off}_2(pA + qB, rA + sB).$$

Scaling A and B as suggested in [21] is the special case $q = r = 0$.

In broad outline, the algorithm we present consists of a sequence of similarity transformations by plane rotations each of which is chosen to minimize one of the measures (2) or (3). This is done as follows. Let $R = R(i, j, c, s) \in \mathbf{C}^{n \times n}$ be a plane rotation with $c \in \mathbf{R}$. The restriction that c be real does not change the amount by which $\text{off}_2(A, B)$ can be reduced. A simple calculation shows that

$$(4) \quad \begin{aligned} \text{off}_2(RAR^H, RBR^H) &= \text{off}_2(A, B) - |a_{ij}|^2 - |b_{ij}|^2 - |a_{ji}|^2 - |b_{ji}|^2 \\ &\quad + |sc(\bar{a}_{ii} - \bar{a}_{jj}) + c^2\bar{a}_{ij} - s^2\bar{a}_{ji}|^2 \\ &\quad + |sc(a_{ii} - a_{jj}) - s^2a_{ij} + c^2a_{ji}|^2 \\ &\quad + |sc(\bar{b}_{ii} - \bar{b}_{jj}) + c^2\bar{b}_{ij} - s^2\bar{b}_{ji}|^2 \\ &\quad + |sc(b_{ii} - b_{jj}) - s^2b_{ij} + c^2b_{ji}|^2 \end{aligned}$$

The choice of c and s that minimizes $\text{off}_2(RAR^H, RBR^H)$ is the choice that minimizes

$$(5) \quad f_{ij}(c, s) = \|M_{ij}z\|_2 =_{\text{def}} \left\| \begin{bmatrix} \bar{a}_{ij} & \frac{\bar{a}_{ii} - \bar{a}_{jj}}{\sqrt{2}} & -\bar{a}_{ji} \\ a_{ji} & \frac{a_{ii} - a_{jj}}{\sqrt{2}} & -a_{ij} \\ \bar{b}_{ij} & \frac{\bar{b}_{ii} - \bar{b}_{jj}}{\sqrt{2}} & -\bar{b}_{ji} \\ b_{ji} & \frac{b_{ii} - b_{jj}}{\sqrt{2}} & -b_{ij} \end{bmatrix} \begin{bmatrix} c^2 \\ \sqrt{2}cs \\ s^2 \end{bmatrix} \right\|_2.$$

This is a constrained minimization problem. The constraint $|c|^2 + |s|^2 = 1$ implies that $\|z\|_2 = 1$. So, for all $z \in \mathbf{R}^3$, $z^H z = 1$,

$$\|M_{ij}z\|_2 \geq \min_{\|x\|_2=1} \|M_{ij}x\| = \sigma_{\min}(M_{ij}).$$

Parameterizing c and s as $c = \cos(\theta)$, $s = e^{i\phi} \sin(\theta)$ for $\theta, \phi \in \mathbf{R}$ makes the minimization of (5) into a two real variable optimization problem. The following lemma shows that only the values $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and $\phi \in [-\pi, \pi]$ need to be searched to minimize (5).

LEMMA 2.1. *All values of (5) occur with $c = \cos(\theta)$, $s = e^{i\phi} \sin(\theta)$ for some value of $(\theta, \phi) \in [-\frac{\pi}{4}, \frac{\pi}{4}] \times [-\pi, \pi]$.*

Proof. Define $g_{ij}(\theta, \phi)$ by $g_{ij}(\theta, \phi) = f_{ij}(\cos(\theta), e^{i\phi} \sin(\theta))$. The functions $\cos^2(\theta)$, $\cos(\theta) \sin(\theta)$ and $\sin^2(\theta)$ are π -periodic. Thus, (5) implies that $g_{ij}(\theta, \phi)$ is π -periodic in θ . The special structure of M_{ij} in (5) along with the trivial observation $\|e^{-2i\phi} M_{ij}z\| = \|M_{ij}z\|$ shows that $g(\theta, \phi) = g(\theta + \frac{\pi}{2}, \phi) = g(-\theta - \frac{\pi}{2}, \phi)$. Hence, for each fixed value of ϕ , $g_{ij}(\theta, \phi)$ assumes all its values on $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. \square

In the terminology of [40], an “inner rotation” is one for which $|c| \geq |s|$. An “outer rotation” is one for which $|c| < |s|$. Corresponding to each inner rotation that minimizes (5), there is an “outer” rotation, that also minimizes (5). By choosing $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, we have made the choice of using only inner rotations. Our proof of quadratic convergence depends upon them.

We have found no “simple” explicit formulae for the minimizers of (5). However, explicit formulae are known in some special cases. Goldstein and Horwitz [21] and Eberlein [13] give explicit formulae for the special case $A = A^H$ and $B = -B^H$. We give explicit formulae for other special cases in Section 6. General optimization

algorithms like those described in [19] provide effective ways to find minimizers $(\theta, \phi) \in [-\frac{\pi}{4}, \frac{\pi}{4}] \times [-\pi, \pi]$. In our MATLAB [29] implementation we have used the exact minimizer if an explicit formula for the exact minimum was available. Otherwise, we used a heuristic approximate minimizer described below.

The work needed to find a minimizer of (5) does not grow with n . The work required to perform a similarity transformation by a rotation is proportional to n . For large enough problems, the work of finding minimizers of (5) is negligible. However, general constrained optimization algorithms are relatively complicated. A computer implementation of Algorithm 1 (described below) would devote the majority of the code to the optimization problem. Fortunately, it is sufficient to approximate the minimizer. Consider the following strategy:

Let $c_1 \in \mathbf{R}$ and $s_1 \in \mathbf{C}$ be minimizers of

$$(6) y_{ij}(c_1, s_1) = |s_1 c_1 (\bar{a}_{ii} - \bar{a}_{jj}) + c_1^2 \bar{a}_{ij} - s_1^2 \bar{a}_{ji}|^2 + |s_1 c_1 (a_{ii} - a_{jj}) - s_1^2 a_{ij} + c_1^2 a_{ji}|^2,$$

and let $c_2 \in \mathbf{R}$ and $s_2 \in \mathbf{C}$ be minimizers of

$$(7) h_{ij}(c_2, s_2) = |s_2 c_2 (\bar{b}_{ii} - \bar{b}_{jj}) + c_2^2 \bar{b}_{ij} - s_2^2 \bar{b}_{ji}|^2 + |s_2 c_2 (b_{ii} - b_{jj}) - s_2^2 b_{ij} + c_2^2 b_{ji}|^2.$$

Explicit formulae for (c_1, s_1) and (c_2, s_2) appear in [13, 21]. Use as approximate minimizers the pair $(c, s) = (c_i, s_i)$, $i = 1, 2$, that yields the smaller value of (5). This strategy has worked well in practice and the proof of local quadratic convergence presented in Section 3 goes through with little modification for this approximate minimizer.

The following algorithm summarizes the procedure for simultaneous diagonalization of a commuting pair of normal matrices.

ALGORITHM 1.

INPUT: $\epsilon > 0$; $A, B \in \mathbf{C}^{n \times n}$ such that $AB = BA$, $AA^H = A^H A$ and $BB^H = B^H B$

OUTPUT: $Q \in \mathbf{C}^{n \times n}$ such that $\text{off}_2(Q^H A Q, Q^H B Q) \leq \epsilon(\|A\|_F + \|B\|_F)$ and $Q Q^H = I$.

1. $Q \leftarrow I$
2. **WHILE** $\text{off}_2(A, B) > \epsilon(\|A\|_F + \|B\|_F)$
 - **FOR** $i = 1, 2, 3, \dots, n$
 - 4. **FOR** $j = i + 1, i + 2, i + 3, \dots, n$
 - 5. Select $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and $\phi \in [-\pi, \pi]$ such that $c = \cos(\theta)$ and $s = e^{i\phi} \sin(\theta)$ minimizes (5). (Or, approximately minimizes (5) as described in (6) and (7)).
 - 6. $R \leftarrow R(i, j, c, s)$
 - 7. $Q \leftarrow QR$; $A \leftarrow R^H A R$; $B \leftarrow R^H B R$

If rotations are stored and applied in an efficient manner similar to that outlined for real rotations in [22, §5.1] or [33, §6.4], then each sweep (Step 2) uses approximately $8n^3$ complex flops to update A and B and approximately $2n^3$ complex flops to accumulate Q . A complex flop is the computation effort required to execute the Fortran statement

$$(8) \quad \mathbf{A}(\mathbf{I}, \mathbf{J}) = \mathbf{A}(\mathbf{I}, \mathbf{J}) + \mathbf{S} * \mathbf{A}(\mathbf{K}, \mathbf{J})$$

where \mathbf{A} and \mathbf{S} are of type **COMPLEX**.

Algorithm 1 needs storage for approximately $3n^2$ complex numbers. This can be shaved to $2n^2$ complex numbers if Q is not required.

As with the serial Jacobi algorithm, to promote rapid convergence in the case of multiple eigenvalues, it is a good idea to use a similarity transformation by a

permutation matrix to put the diagonal entries of A and B in lexicographic order [20]. Such an eigenvalue ordering is required by our proof of local quadratic convergence in Section 3.

In our experience, for randomly chosen examples with $n \leq 80$, rarely does $\epsilon = 10^{-14}$ make Algorithm 1 require more than six sweeps. In Section 3 we show that Algorithm 1 has local quadratic convergence. There are, however, examples for which Algorithm 1 does not converge. For example, if $A \in \mathbf{R}^{n \times n}$ is given by

$$(9) \quad a_{kj} = \begin{cases} \cos((j+k)\pi/n) & \text{if } j \neq k \\ \frac{2-n}{2} \cos(2k\pi/n) & \text{if } j = k \end{cases}$$

and $B \in \mathbf{R}^{n \times n}$ is given by

$$(10) \quad b_{kj} = \begin{cases} \sin((j+k)\pi/n) & \text{if } j \neq k \\ \frac{2-n}{2} \sin(2k\pi/n) & \text{if } j = k \end{cases}$$

and $n > 6$, then it is easily to verify through Theorem 6.1 that no rotation reduces $\text{off}_2(A, B)$, so Algorithm 1 with exact minimization in Step 5 leaves A and B invariant. This example is essentially due to Voevodin [45].

Usually in practice, rounding errors perturb (9) and (10) sufficiently to allow Algorithm 1 (with exact minimization in Step 5) to reduce $\text{off}_2(A, B)$ and ultimately to converge to a simultaneously diagonal form, albeit rather slowly. For example, an early version of our experimental implementation on a Sun 4/60 (with unit round approximately 10^{-16}) applied to the $n = 20$ case needed 47 sweeps to reduce $\text{off}_2(A, B)$ to approximately $10^{-14}(\|A\| + \|B\|)$. Rounding errors did not perturb the $n = 30$ enough to allow convergence.

Example (9) and (10) is not held fixed by the heuristic minimization strategy, but it does sometimes require a few more sweeps than a random example. Algorithm 1 with the approximate minimization heuristic in Step 5 needed from 9 to 10 sweeps to reduce $\text{off}_2(A, B)$ to less than $10^{-14}(\|A\| + \|B\|)$ for $n = 10$, $n = 15$, $n = 20$, $n = 25$, and $n = 30$.

Scaling one of the two commuting matrices, by (say) replacing A by $A/2$ as suggested in [21] is a more reliable way to “break away” from a fixed point like (9) and (10). This is equivalent to using $\text{off}_{2-1,0,0,2}$ in place of off_2 . Motivated in part by the “exceptional shift” strategy often used with the QR algorithm, we changed our experimental code to use $\text{off}_{2-1,0,0,2}$ for one sweep, whenever $\text{off}_2(A, B)$ declines by less than the 1% across a sweep. With this modification, only 6 to 7 sweeps were required to reduce $\text{off}_2(A, B)$ to less than $10^{-14}(\|A\| + \|B\|)$ for the $n = 10$, $n = 15$, $n = 20$, $n = 25$, and $n = 30$ cases of (9) and (10). (The choice of 1% is *ad hoc*. A more cautious approach would require a greater per sweep reduction in $\text{off}_2(A, B)$ and would try other scaling factors, if 1/2 doesn’t work.)

We have not been able to show that Algorithm 1 with the above modification is globally convergent.

3. Convergence Properties. Algorithm 1 shares many of the desirable properties of algorithms related to the serial Jacobi algorithm for the real symmetric eigenvalue problem [20, 23, 36, 38, 41, 46]. In our experience, Algorithm 1, with the above strategy for avoiding stagnation, converges globally and ultimately quadratically.

In this section we establish the local quadratic convergence and numerical stability of Algorithm 1. In parenthetical remarks we give a rough sketch of how the argument can be modified to show the local convergence of the heuristic variation (6), (7).

Unfortunately, we do not have a proof of global convergence. Global convergence for special cases is shown in [21] and, using a modified normreducing algorithm in [17]. The latter however uses nonunitary transformations. In our case the use of unitary similarity transformations confers satisfactory numerical stability.

3.1. Local Quadratic Convergence. Here we follow [20, 36, 46] to show that ultimately, Algorithm 1 converges quadratically. In particular, the resemblance to [36] is unmistakable.

Let the set of eigenvalues of A be $\{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n\}$ and let the set of eigenvalues of B be $\{\beta_1, \beta_2, \dots, \beta_n\}$. Set

$$(11) \quad \delta = \frac{1}{2} \min \left(\min_{\alpha_i \neq \alpha_j} |\alpha_i - \alpha_j|, \min_{\beta_i \neq \beta_j} |\beta_i - \beta_j| \right).$$

Denote the k -th rotation of Algorithm 1 by $R^{(k)} = R(i, j, c^{(k)}, s^{(k)})$, and let $A^{(k)}$ and $B^{(k)}$ be the values of A and B immediately before the k -th rotation. Following common practice, we call $(a_{ij}^{(k)}, a_{ji}^{(k)}, b_{ij}^{(k)}, b_{ji}^{(k)})$ the “ k -th pivots”, because the rotation $R^{(k)}(i, j, c^{(k)}, s^{(k)})$ is chosen to minimize $|a_{ij}^{(k+1)}|^2 + |a_{ji}^{(k+1)}|^2 + |b_{ij}^{(k+1)}|^2 + |b_{ji}^{(k+1)}|^2$.

Define ρ_k by

$$\rho_k = \sqrt{\text{off}_2(A^{(k)}, B^{(k)})}.$$

Partition $A^{(k)}$ and $B^{(k)}$ as

$$A^{(k)} = D_A^{(k)} + E_A^{(k)}$$

and

$$B^{(k)} = D_B^{(k)} + E_B^{(k)}$$

where $D_A^{(k)}, D_B^{(k)} \in \mathbf{C}^{n \times n}$ are diagonal and $E_A^{(k)}, E_B^{(k)} \in \mathbf{C}^{n \times n}$ have zero diagonal. Note that $\rho_k = \sqrt{\text{off}_2(A^{(k)}, B^{(k)})} = \left\| [E_A^{(k)}, E_B^{(k)}] \right\|_F$.

Suppose Algorithm 1 has converged to the point that

$$(12) \quad \rho_k = \sqrt{\text{off}_2(A^{(k)}, B^{(k)})} < \frac{1}{2} \delta.$$

Using a permutation similarity, we may order the eigenvalues α_i and β_i so that

$$(13) \quad |a_{ii}^{(k)} - \alpha_i| \leq \left\| E_A^{(k)} \right\|_F \leq \rho_k < \frac{1}{2} \delta$$

and

$$(14) \quad |b_{ii}^{(k)} - \beta_i| \leq \left\| E_B^{(k)} \right\|_F \leq \rho_k < \frac{1}{2} \delta.$$

Because Algorithm 1 uses inner rotations and $\text{off}_2(A^{(k)}, B^{(k)})$ is monotonically decreasing, (13) and (14) hold throughout the remainder of Algorithm 1 [20, 36, 46]. In particular, the order in which the eigenvalues will eventually appear along the diagonals of A and B is fixed.

(The heuristic (6) and (7) does not necessarily make $\text{off}_2(A^{(k)}, B^{(k)})$ decrease monotonically. However, it can be shown that $\text{off}_2(A^{(k)}, B^{(k)})$ does not increase by

more that $O(\delta^2)$. Thus, under a slightly stronger hypothesis, (13) and (14) continue to hold.)

Each diagonal entry $a_{ii}^{(k)}$ is said to be “affiliated” with α_i and each diagonal entry $b_{ii}^{(k)}$ is “affiliated” with β_i . As Algorithm 1 drives $\text{off}_2(A^{(k)}, B^{(k)})$ to zero, $\lim_{k \rightarrow \infty} a_{ii}^{(k)} = \alpha_i$ and $\lim_{k \rightarrow \infty} b_{ii}^{(k)} = \beta_i$. The affiliation of a diagonal entry does not change during subsequent steps of the algorithm [20, 36, 46].

Inequality (13) implies that if $\alpha_i \neq \alpha_j$, then

$$(15) \quad |a_{ii}^{(k)} - a_{jj}^{(k)}| > \delta.$$

Inequality (14) implies that if $\beta_i \neq \beta_j$, then

$$(16) \quad |b_{ii}^{(k)} - b_{jj}^{(k)}| > \delta.$$

An eigenvalue pair (α, β) has multiplicity m , if there are m distinct integers i_j , $j = 1, 2, 3, \dots, m$ for which $(\alpha, \beta) = (\alpha_{i_j}, \beta_{i_j})$. Using permutation similarity, we may order the diagonal entries so that affiliates of an m -fold multiple eigenvalue pair appear in m adjacent diagonal entries.

We will show the following theorem.

THEOREM 3.1. *Suppose that the h -th rotation starts a sweep of Algorithm 1, i.e., $a_{12}^{(h)}, a_{21}^{(h)}, b_{12}^{(h)}, b_{21}^{(h)}$ are the h -th pivots. If (12) holds and affiliates of m -fold multiple eigenvalue pairs appear in adjacent diagonal entries, then at the end of the sweep*

$$\rho_{h+n(n-1)/2} \leq 2n(9n-13) \frac{\rho_h^2}{\delta}.$$

We will need several lemmas to prove Theorem 3.1. The proof has two parts. The first part establishes that,

$$(17) \quad \sqrt{|a_{ij}^{(k+1)}|^2 + |a_{ji}^{(k+1)}|^2 + |b_{ij}^{(k+1)}|^2 + |b_{ji}^{(k+1)}|^2} \leq O(\rho_k^2).$$

In the second part, we will show that subsequent rotations “preserve” (17) for subsequent values of k .

Lemmas similar to the following are well known tools in the study of quadratic convergence of cyclic Jacobi algorithms [36, 41, 48].

LEMMA 3.2. *If (12) holds, $(\alpha_i, \beta_i) = (\alpha_j, \beta_j)$ and $i \neq j$, then*

$$\sqrt{|a_{ij}^{(k+1)}|^2 + |a_{ji}^{(k+1)}|^2 + |b_{ij}^{(k+1)}|^2 + |b_{ji}^{(k+1)}|^2} \leq \frac{\sqrt{2}\rho_k^2}{2\delta}.$$

Proof. Regardless of the particular choice of $R^{(k)}(i, j, c^{(k)}, s^{(k)})$, the partitioning lemma [36, §2] implies that

$$(18) \quad \sqrt{|a_{ij}^{(k+1)}|^2 + |a_{ji}^{(k+1)}|^2} \leq \frac{\rho_k^2}{2\delta}$$

and

$$(19) \quad \sqrt{|b_{ij}^{(k+1)}|^2 + |b_{ji}^{(k+1)}|^2} \leq \frac{\rho_k^2}{2\delta}.$$

□

The next lemma covers the case $(\alpha_i, \beta_i) \neq (\alpha_j, \beta_j)$.

LEMMA 3.3. *Suppose that $a_{ij}^{(k)}$, $a_{ji}^{(k)}$, $b_{ij}^{(k)}$, $b_{ji}^{(k)}$, are the k -th pivots. If (12) holds and $(\alpha_i, \beta_i) \neq (\alpha_j, \beta_j)$, then*

$$\sqrt{|a_{ij}^{(k+1)}|^2 + |a_{ji}^{(k+1)}|^2 + |b_{ij}^{(k+1)}|^2 + |b_{ji}^{(k+1)}|^2} \leq 5 \frac{\rho_k^2}{\delta}.$$

Proof. Without loss of generality, assume that

$$(20) \quad |\alpha_i - \alpha_j| \geq |\beta_i - \beta_j|.$$

In particular, (20) implies $\alpha_i \neq \alpha_j$. In [36, §3], it is shown that there is a rotation $\hat{R} = R(i, j, c_A, s_A)$ such that for $\hat{A} = \hat{R}^H A^{(k)} \hat{R}$,

$$(21) \quad \sqrt{|\hat{a}_{ij}|^2 + |\hat{a}_{ji}|^2} \leq \frac{\rho_k^2}{2\delta}.$$

In fact, this rotation minimizes the left-hand-side of (21) (and (6)), so

$$(22) \quad \|E_{\hat{A}}\|_F \leq \|E_A^{(k+1)}\|_F.$$

Set $\hat{B} = \hat{R}^H B^{(k)} \hat{R}$. Recall that \hat{A} and \hat{B} commute. The (i, j) -th entry of the equation $\hat{A}\hat{B} - \hat{B}\hat{A} = 0$ is

$$(23) \quad \hat{b}_{ij}(\hat{a}_{ii} - \hat{a}_{jj}) - \hat{a}_{ij}(\hat{b}_{ii} - \hat{b}_{jj}) + \sum_{\substack{k \neq i \\ k \neq j}} (\hat{a}_{ik}\hat{b}_{kj} - \hat{b}_{ik}\hat{a}_{kj}) = 0.$$

Applying Hölder's inequality [35] to the sum and using (21) to bound the second term on the left-hand-side of (23) gives

$$(24) \quad |\hat{b}_{ij}(\hat{a}_{ii} - \hat{a}_{jj})| \leq \rho_k^2 + \frac{\rho_k^2}{2\delta} |\hat{b}_{ii} - \hat{b}_{jj}|.$$

Inequality (22) implies that (13) and (15) hold for \hat{A} and

$$(25) \quad |\hat{b}_{ii} - \hat{b}_{jj}| \leq |b_{ii}^{(k)} - b_{jj}^{(k)}| + 2|b_{ij}^{(k)}| + 2|b_{ji}^{(k)}| \leq |b_{ii}^{(k)} - b_{jj}^{(k)}| + 4\rho_k \leq |b_{ii}^{(k)} - b_{jj}^{(k)}| + 2\delta.$$

Inequalities (15) and (24) imply

$$|\hat{b}_{ij}| \leq \frac{\rho_k^2}{2\delta} \left(2 + \frac{\hat{b}_{ii} - \hat{b}_{jj}}{\hat{a}_{ii} - \hat{a}_{jj}} \right).$$

To bound the right-hand-side, apply (25) to get

$$\left| \frac{\hat{b}_{ii} - \hat{b}_{jj}}{\hat{a}_{ii} - \hat{a}_{jj}} \right| \leq \frac{|b_{ii}^{(k)} - b_{jj}^{(k)}| + 2\delta}{|\hat{a}_{ii} - \hat{a}_{jj}|}.$$

The triangle inequality applied to $|(b_{ii}^{(k)} - \beta_i) + (\beta_i - \beta_j) + (\beta_j - b_{jj}^{(k)})|$ and $|(\alpha_i - \alpha_j) + (\alpha_i - \hat{a}_{ii}) + (\alpha_j - \hat{a}_{jj})|$ gives

$$\left| \frac{\hat{b}_{ii} - \hat{b}_{jj}}{\hat{a}_{ii} - \hat{a}_{jj}} \right| \leq \frac{|b_{ii}^{(k)} - \beta_i| + |\beta_i - \beta_j| + |\beta_j - b_{jj}^{(k)}| + 2\delta}{|\alpha_i - \alpha_j| - |\alpha_i - \hat{a}_{ii}| - |\alpha_j - \hat{a}_{jj}|}.$$

Inequalities (13) and (14) imply

$$\left| \frac{\hat{b}_{ii} - \hat{b}_{jj}}{\hat{a}_{ii} - \hat{a}_{jj}} \right| \leq \frac{|\beta_i - \beta_j| + 3\delta}{|\alpha_i - \alpha_j| - \delta}.$$

Multiply and divide the right-hand-side by $(\alpha_i - \alpha_j)^{-1}$ and apply (20), (13) and (14) to get

$$\left| \frac{\hat{b}_{ii} - \hat{b}_{jj}}{\hat{a}_{ii} - \hat{a}_{jj}} \right| \leq \frac{\frac{\beta_i - \beta_j}{\alpha_i - \alpha_j} + \frac{3\delta}{\alpha_i - \alpha_j}}{1 - \frac{\delta}{\alpha_i - \alpha_j}} \leq \frac{1 + 3/2}{1 - 1/2} \leq 5.$$

It now follows that $|\hat{b}_{ij}| \leq \frac{7\rho_k^2}{2\delta}$. Similarly, it can be shown that $|\hat{b}_{ji}| \leq \frac{7\rho_k^2}{2\delta}$. Therefore,

$$(26) \quad \sqrt{|\hat{a}_{ij}|^2 + |\hat{a}_{ji}|^2 + |\hat{b}_{ij}|^2 + |\hat{b}_{ji}|^2} \leq \frac{\sqrt{99}\rho_k^2}{2\delta} \leq 5\frac{\rho_k^2}{\delta}.$$

Now, $R^{(k)}$ is chosen to minimize $|a_{ij}^{(k+1)}|^2 + |a_{ji}^{(k+1)}|^2 + |b_{ij}^{(k+1)}|^2 + |b_{ji}^{(k+1)}|^2$, so inequality (26) gives an upper bound on the minimum value. (If $R^{(k)}$ is chosen using the heuristic (6), (7), then (26) is one of the two choices of $\text{off}_2(A^{(k+1)}, B^{(k+1)})$ over which the heuristic minimizes. Hence, inequality (26) also gives an upper bound on the minimum value for this case.) \square

Lemmas 3.2 and 3.3 imply (17). It remains to show that subsequent rotations preserve (17). The following lemma gives a bound on the angles of rotation that occur in Algorithm 1.

LEMMA 3.4. *If (12) holds and $(\alpha_i, \beta_i) \neq (\alpha_j, \beta_j)$, then*

$$(27) \quad |s^{(k)}| \leq \sqrt{2} \left(1 + \frac{5\rho_k}{\delta} \right) \frac{\rho_k}{\delta} \leq \frac{9\rho_k}{\delta}.$$

Proof. For ease of notation, set $(c, s) = (c^{(k)}, s^{(k)})$. Without loss of generality, we may assume that $\alpha_i \neq \alpha_j$. Lemma 3.3 implies that

$$|a_{ij}^{(k+1)}| = |sc(a_{ii}^{(k)} - a_{jj}^{(k)}) - s^2 a_{ji}^{(k)} + c^2 a_{ij}^{(k)}| \leq \frac{5\rho_k^2}{\delta}.$$

The triangle inequality gives

$$(28) \quad \begin{aligned} |sc(a_{ii}^{(k)} - a_{jj}^{(k)})| &\leq \frac{5\rho_k^2}{\delta} + |s^2 a_{ji}^{(k)} + c^2 a_{ij}^{(k)}| \\ &\leq \frac{5\rho_k^2}{\delta} + |s^2 a_{ji}^{(k)}| + |c^2 a_{ij}^{(k)}| \\ &\leq \frac{5\rho_k^2}{\delta} + \rho_k. \end{aligned}$$

Algorithm 1 chooses $c = \cos(\theta)$ for some $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, so $|c| \geq \frac{\sqrt{2}}{2}$. Thus, (15) and (28) together imply the left-hand inequality of (27). The right-hand inequality follows from the left-hand inequality and (12). \square

Proof of Theorem 3.1:

Fix a particular choice of (i, j) and consider $a_{ij}^{(k)}$ during a sweep of Algorithm 1 as k varies from $k = h$ through $k = h + n(n - 1)/2$.

Note that ρ_k is monotonically decreasing in k , so if $(\alpha_i, \beta_i) = (\alpha_j, \beta_j)$, then Lemma 3.2 implies that, for $k = h, h + 1, h + 2, \dots, h + n(n - 1)/2$, $|a_{ij}^{(k)}| \leq \frac{\sqrt{2}\rho_k^2}{2\delta} \leq \frac{\sqrt{2}\rho_h^2}{2\delta}$. (As noted in the proof of Lemma 3.2, a slightly weaker inequality holds for the heuristic (6), (7).)

Suppose $(\alpha_i, \beta_i) \neq (\alpha_j, \beta_j)$ and the k -th pivots are $a_{ij}^{(k)}$, $a_{ji}^{(k)}$, $b_{ij}^{(k)}$ and $b_{ji}^{(k)}$. Fix i and j . Lemma 3.3 implies that $|a_{ij}^{(k+1)}| \leq 5\frac{\rho_k^2}{\delta}$. Affiliates of multiple eigenvalue pairs appear in adjacent diagonal entries, so subsequent pivots in row i satisfy the hypothesis of Lemma 3.4. If $j < n$ and (for ease of notation) $(c, s) = (c^{(k+1)}, s^{(k+1)})$, then

$$\begin{aligned} |a_{ij}^{(k+2)}| &= |ca_{ij}^{(k+1)} + \bar{s}a_{j+1,j}^{(k+1)}| \\ &\leq |a_{ij}^{(k+1)}| + |\bar{s}||a_{j+1,j}^{(k+1)}| \\ &\leq 5\frac{\rho_k^2}{\delta} + \frac{9\rho_{k+1}^2}{\delta} \\ &\leq 14\frac{\rho_k^2}{\delta} \leq 14\frac{\rho_h^2}{\delta} \end{aligned}$$

Similarly, each subsequent pivot in row i may increase $|a_{ij}^{(k)}|$ by at most $9\frac{\rho_h^2}{\delta}$.

Suppose that the last pivot in row i is the q -th pivot. There are at most $n - i - 1 \leq n - 2$ pivots in row i subsequent to the k -th pivot. Hence, for $j \neq i$, $|a_{ij}^{(q)}| \leq (5 + 9(n - 2))\frac{\rho_h^2}{\delta}$. We may bound the q -th off-diagonal row sum by

$$r_{iq} =_{\text{def}} \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}^{(q)}|^2 \leq (n - 1)(5 + 9(n - 2))^2 \frac{\rho_h^4}{\delta^2}.$$

Pivoting in rows other than row i leaves the off-diagonal row sum invariant, so for $q < k < h + n(n - 1)/2$, $r_{ik} = r_{iq} \leq (n - 1)(9n - 13)^2 \frac{\rho_h^4}{\delta^2}$.

The same argument applied to B yields the identical bound for the off-diagonal row sums of B . Adding the bounds of the off-diagonal row sums of both A and B at the end of the sweep, we get

$$\rho_{h+n(n-1)/2} \leq \sqrt{2n(n-1)(9n-13)} \frac{\rho_h^2}{\delta} \leq 2n(9n-13) \frac{\rho_h^2}{\delta}.$$

□

It is clear from the derivation that the bound of Theorem 3.1 is not tight.

The theorem assumes that affiliates of multiple eigenvalue pairs appear in adjacent positions along the diagonal. This assumption guarantees that large angles of rotation do not cause “fill-in” of small off-diagonal entries. As suggested in [36], a threshold strategy would have much the same effect.

3.2. Rounding Errors. Algorithm 1 uses only unitary similarity transformations. A classical rounding error analysis of the construction and application of real

rotations appears in [47, p.131ff]. It is easily extended to the complex case. Applying it to Algorithm 1 results in the following theorem.

THEOREM 3.5. *Suppose the rotations in Algorithm 1 are constructed using methods similar to those described in [22, §5.1] or [33, §6.4]. If $A, B \in \mathbf{C}^{n \times n}$ are a commuting normal pair input to Algorithm 1 and $Q^{(k)}, A^{(k)}, B^{(k)} \in \mathbf{C}^{n \times n}$ are the computed versions of Q, A and B after k sweeps, then there are matrices $F_A, F_B \in \mathbf{C}^{n \times n}$ such that*

$$(A + F_A) Q^{(k)} = Q^{(k)} A^{(k)},$$

$$(B + F_B) Q^{(k)} = Q^{(k)} B^{(k)}$$

and

$$\|F_A\|_F + \|F_B\|_F + \|Q^{(k)H} Q - I\|_F \leq p(n)\mu$$

where μ is the machine precision and $p(n)$ is a modest polynomial that depends on the details of the arithmetic. Thus, the effects of rounding errors are equivalent to making small normwise perturbations in the original data matrices A and B . Algorithm 1 is backward numerically stable.

4. Real Matrices. If Algorithm 1 is applied to a pair of real, commuting normal matrices A and B , then it will use nontrivial complex arithmetic on what is essentially a real problem. Thus, storage must be set aside for two complex arrays instead of two real arrays and complex arithmetic must be used instead of real arithmetic. Complex rounding errors may perturb the eigenvalues of A and B so that they do not appear in complex conjugate pairs. This section presents a modification of Algorithm 1 that avoids complex arithmetic.

Of course, there may be no real similarity transformation that simultaneously diagonalizes A and B . However, the following theorem shows that there is a real similarity transformation that block diagonalizes A and B with 1-by-1 and 2-by-2 blocks.

THEOREM 4.1. *If $A, B \in \mathbf{R}^{n \times n}$, $A^T A = A A^T$, $B^T B = B B^T$, and $AB = BA$, then there exists $Q \in \mathbf{R}^{n \times n}$, such that $Q^T Q = I$, $D_A = Q^T A Q$ is block diagonal and $D_B = Q^T B Q$ is block diagonal with 2-by-2 blocks and at most one, trailing 1-by-1 block.*

Proof. If $n = 1$ or $n = 2$, then the theorem holds with $Q = [1] \in \mathbf{R}^{1 \times 1}$ or $Q = I \in \mathbf{R}^{2 \times 2}$.

Assume the induction hypothesis that the theorem holds for all real, normal, k -by- k , commuting pairs A and B for $k < n$ and $3 \leq n$. Let $A \in \mathbf{R}^{n \times n}$ and $B \in \mathbf{R}^{n \times n}$ be commuting normal matrices, and let $x \in \mathbf{C}^n$ be a simultaneous eigenvector of A and B with eigenvalues λ_A and λ_B respectively. There are two cases to consider: (i) x is linearly independent of \bar{x} and (ii) x and \bar{x} are linear dependent.

If x and \bar{x} are linearly independent, then $A\bar{x} = \bar{\lambda}_A \bar{x}$ and $B\bar{x} = \bar{\lambda}_B \bar{x}$. Thus, x and \bar{x} are simultaneous, linearly independent eigenvectors of A and B . The vectors $y = x + \bar{x}$ and $z = i(x - \bar{x})$ span a two dimensional, real, invariant subspace of A and B . Let $U \in \mathbf{R}^{n \times n}$ be an orthogonal matrix whose first two columns form an orthonormal basis of $\text{span}(y, z)$ and set $\tilde{A} = U^T A U$ and $\tilde{B} = U^T B U$. Thus, $\tilde{A} = \text{diag}(A_{11}, A_{22})$, $\tilde{B} = \text{diag}(B_{11}, B_{22})$ where $A_{11}, B_{11} \in \mathbf{R}^{2 \times 2}$ and $A_{22}, B_{22} \in \mathbf{R}^{n-2 \times n-2}$. Observe that \tilde{A} and \tilde{B} are real, normal, commuting matrices and hence A_{22} and B_{22} are

commuting, normal $(n-2)$ -by- $(n-2)$ matrices. By the induction hypothesis, there exists an orthogonal matrix $V_{22} \in \mathbf{R}^{n-2 \times n-2}$ such that $V_{22}^T A_{22} V_{22}$ and $V_{22}^T B_{22} V_{22}$ are block diagonal with 2-by-2 and at most one, trailing 1-by-1 block. If V_{11} is the 2-by-2 identity matrix and $V = \text{diag}(V_{11}, V_{22})$ then the matrix $Q = UV$ satisfies the conclusion of the theorem.

If x and \bar{x} are linearly dependent, then $\lambda_A \in \mathbf{R}$, $\lambda_B \in \mathbf{R}$ and x is a scalar multiple of some real, simultaneous eigenvector $y \in \mathbf{R}^n$. Without loss of generality we may choose y such that $\|y\|_2 = 1$. Let $U \in \mathbf{R}^{n \times n}$ be an orthogonal matrix whose last column is y and set $\tilde{A} = U^T A U$ and $\tilde{B} = U^T B U$. Observe that \tilde{A} and \tilde{B} are real, normal, commuting matrices and that $\tilde{A} = \text{diag}(A_{11}, a_{nn})$, $\tilde{B} = \text{diag}(B_{11}, b_{nn})$ where $a_{nn} = \lambda_A$, $b_{nn} = \lambda_B$ and $A_{11}, B_{11} \in \mathbf{R}^{n-1 \times n-1}$. By the induction hypothesis, there exists an orthogonal matrix $V_{11} \in \mathbf{R}^{n-1 \times n-1}$ such that $V_{11}^T A_{11} V_{11}$ and $V_{11}^T B_{11} V_{11}$ are block diagonal with 2-by-2 and at most one, trailing 1-by-1 block. If $V = \text{diag}(V_{11}, [1])$ then the matrix $Q = UV$ satisfies the conclusion of the theorem. Note that at this point that if n is even, it is necessary to logically combine the two trailing 1-by-1 blocks into a single trailing 2-by-2 block. \square

Partition each matrix $M \in \mathbf{R}^{n \times n}$ into a k -by- k block matrix as

$$(29) \quad M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & \cdots & M_{1k} \\ M_{21} & M_{22} & M_{23} & \cdots & M_{2k} \\ M_{31} & M_{32} & M_{33} & \cdots & M_{3k} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ M_{k1} & M_{k2} & M_{k3} & \cdots & M_{kk} \end{bmatrix}$$

where $k = \lfloor \frac{n+1}{2} \rfloor$ and for $i, j \leq \lfloor \frac{n}{2} \rfloor$ $M_{ij} \in \mathbf{R}^{2 \times 2}$. If n is odd, then for $j \leq k-1$, $M_{kj} \in \mathbf{R}^{1 \times 2}$, for $i \leq k-1$, $M_{ik} \in \mathbf{R}^{2 \times 1}$ and $M_{kk} \in \mathbf{R}^{1 \times 1}$. Theorem 4.1 states that there is a real orthogonal similarity transformation that simultaneously block diagonalizes A and B conformally with (29).

The next algorithm makes extensive use of elementary orthogonal matrices $Z = Z(i, j, U) \in \mathbf{R}^{n \times n}$ that are partitioned conformally with (29). Define the block rotation $Z(i, j, U)$ to have the form

$$(30) \quad Z(i, j, U) = \begin{bmatrix} I & & & & & \\ & \ddots & & & & \\ & & I & & & \\ & & & Z_{ii} & & Z_{ij} \\ & & & & I & \\ & & & & & \ddots \\ & & & & & & I \\ & & & Z_{ji} & & & & Z_{jj} & & \\ & & & & & & & & I \\ & & & & & & & & & \ddots \\ & & & & & & & & & & I \end{bmatrix}$$

Here,

$$\begin{bmatrix} Z_{ii} & Z_{ij} \\ Z_{ji} & Z_{jj} \end{bmatrix} =: U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}$$

is an orthogonal matrix. If $j < k$ or if n is even, then $U \in \mathbf{R}^{4 \times 4}$. Otherwise, $U \in \mathbf{R}^{3 \times 3}$. A similar definition of “blockwise rotation” appears in [18].

The Jacobi annihilators of [31] and the orthogonal matrices proposed in [42] are block rotations.

We will use block rotations $Z(i, j, U)$ where U has the form

$$(31) \quad U = R(2, 4, c_4, s_4)R(1, 4, c_3, s_3)R(2, 3, c_2, s_2)R(1, 3, c_1, s_1)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_4 & 0 & -s_4 \\ 0 & 0 & 1 & 0 \\ 0 & s_4 & 0 & c_4 \end{bmatrix} \begin{bmatrix} c_3 & 0 & 0 & -s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s_3 & 0 & 0 & c_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & -s_2 & 0 \\ 0 & s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ 0 & 1 & 0 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

or

$$(32) \quad U = R(2, 3, c_2, s_2)R(1, 3, c_1, s_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_2 & -s_2 \\ 0 & s_2 & c_2 \end{bmatrix} \begin{bmatrix} c_1 & 0 & -s_1 \\ 0 & 1 & 0 \\ s_1 & 0 & c_1 \end{bmatrix}$$

where for $l = 1, 2, 3, 4$; $c_l, s_l \in \mathbf{R}$ and $c_l^2 + s_l^2 = 1$. It is convenient to parameterize c_l and s_l in (31) and (32) as $c_l = \cos(\theta_l)$ and $s_l = \sin(\theta_l)$ where $\theta_l \in \mathbf{R}$. Note that the Jacobi annihilators of [31] use a different order of the factors in (31) and (32).

In broad outline, Algorithm 2 (presented below) is a block version of Algorithm 1 using the partition (29). It measures progress with a block version of (2) defined by

$$\text{off}_B(A, B) = \sum_{i \neq j} \|A_{ij}\|_F^2 + \|B_{ij}\|_F^2$$

where A and B are partitioned as in (29). At the (i, j) -th step of a sweep, it chooses a block rotation $Z = Z(i, j, U)$ to minimize $\text{off}_B(Z^T A Z, Z^T B Z)$. It is easy to verify that for $i < j$,

$$(33) \quad \text{off}_B(Z^T A Z, Z^T B Z) =$$

$$\text{off}_B(A, B) - \|A_{ij}\|_F^2 - \|A_{ji}\|_F^2 - \|B_{ij}\|_F^2 - \|B_{ji}\|_F^2$$

$$+ \|U_{12}^T A_{ii} U_{11} + U_{22}^T A_{ji} U_{11} + U_{12}^T A_{ij} U_{21} + U_{22}^T A_{jj} U_{21}\|_F^2$$

$$+ \|U_{11}^T A_{ii} U_{12} + U_{21}^T A_{ji} U_{12} + U_{11}^T A_{ij} U_{22} + U_{21}^T A_{jj} U_{22}\|_F^2$$

$$+ \|U_{12}^T B_{ii} U_{11} + U_{22}^T B_{ji} U_{11} + U_{12}^T B_{ij} U_{21} + U_{22}^T B_{jj} U_{21}\|_F^2$$

$$+ \|U_{11}^T B_{ii} U_{12} + U_{21}^T B_{ji} U_{12} + U_{11}^T B_{ij} U_{22} + U_{21}^T B_{jj} U_{22}\|_F^2.$$

Minimizing (33) is equivalent to minimizing the sum of the last four terms. General optimization algorithms like those described in [19] provide effective ways to find minimizers $(c_l, s_l) = (\cos(\theta_l), \sin(\theta_l))$ for the parameters in (31) or (32).

For $U \in \mathbf{R}^{4 \times 4}$ as above define \tilde{U} by

$$\tilde{U} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}.$$

Observe that $Z = Z(i, j, U)$ and $S = Z(i, j, \tilde{U})$ are both block rotations and $\text{off}_B(Z^T AZ, Z^T BZ) = \text{off}_B(S^T AS, S^T BS)$. Thus, we may choose a block rotation so that $\|U_{11}\|_F \geq \|U_{12}\|_F$. Such block rotations are sometimes called “inner” block rotations. The use of inner block rotations promotes convergence [18]. The 3-by-3 case is more complicated. However, the 3-by-3 case only arises in case n is odd. As in [44], it can be eliminated simply adding an extra row and column of zeros to A and B .

The following theorem implies that (33) is minimized by a block rotation of the form of (31) or (32). There is no need to use more general block rotations. A similar theorem about a different set of block rotations was proved in [42] using different methods.

THEOREM 4.2.

1. If $Q \in \mathbf{R}^{4 \times 4}$ is orthogonal, then there is a block rotation $Z \in \mathbf{R}^{4 \times 4}$ of the form of (31) and an orthogonal, block diagonal matrix $D \in \mathbf{R}^{4 \times 4}$ with 2-by-2 blocks such that $ZQ = D$.
2. If $Q \in \mathbf{R}^{3 \times 3}$ is orthogonal, then there is a block rotation $Z \in \mathbf{R}^{3 \times 3}$ of the form of (32) and an orthogonal, block diagonal matrix $D \in \mathbf{R}^{3 \times 3}$ with 2-by-2 and 1-by-1 blocks such that $ZQ = D$.

Proof. We will prove Statement 1. The proof of the Statement 2 is similar.

Select $c_1 = \cos(\theta_1)$ and $s_1 = \sin(\theta_1)$ such that

$$c_1 \det \begin{bmatrix} q_{21} & q_{22} \\ q_{31} & q_{32} \end{bmatrix} + s_1 \det \begin{bmatrix} q_{21} & q_{22} \\ q_{11} & q_{12} \end{bmatrix} = 0.$$

Set $Q^{(1)} = R(1, 3, c_1, s_1)Q$. With this choice of c_1 and s_1 ,

$$\det \begin{bmatrix} q_{21}^{(1)} & q_{22}^{(1)} \\ q_{31}^{(1)} & q_{32}^{(1)} \end{bmatrix} = 0,$$

so there is a choice of $c_2 = \cos(\theta_2)$ and $s_2 = \sin(\theta_2)$ such that

$$s_2 q_{21}^{(1)} + c_2 q_{31}^{(1)} = s_2 q_{22}^{(1)} + c_2 q_{32}^{(1)} = 0.$$

Thus, $Q^{(2)} = R(2, 3, c_2, s_2)Q^{(1)}$ has zeros in the (3, 1) and (3, 2) entries. Select $c_3 = \cos(\theta_3)$ and $s_3 = \sin(\theta_3)$ such that

$$c_3 \det \begin{bmatrix} q_{21}^{(2)} & q_{22}^{(2)} \\ q_{41}^{(2)} & q_{42}^{(2)} \end{bmatrix} + s_3 \det \begin{bmatrix} q_{21}^{(2)} & q_{22}^{(2)} \\ q_{11}^{(2)} & q_{12}^{(2)} \end{bmatrix} = 0.$$

Set $Q^{(3)} = R(1, 4, c_3, s_3)Q^{(2)}$. With this choice of c_3 and s_3 we get

$$\det \begin{bmatrix} q_{21}^{(3)} & q_{22}^{(3)} \\ q_{41}^{(3)} & q_{42}^{(3)} \end{bmatrix} = 0.$$

Hence there is a choice of $c_4 = \cos(\theta_4)$ and $s_4 = \sin(\theta_4)$ such that

$$s_2 q_{21}^{(1)} + c_2 q_{41}^{(1)} = s_2 q_{22}^{(1)} + c_2 q_{42}^{(1)} = 0.$$

Thus, $Q^{(4)} = R(2, 4, c_4, s_4)Q^{(3)}$ has zeros in the (3, 1), (3, 2), (4, 1) and (4, 2) entries.

Set

$$Z = R(2, 4, c_4, s_4)R(1, 4, c_3, s_3)R(2, 3, c_2, s_2)R(1, 3, c_1, s_1)$$

and set $D = Q^{(4)} = ZQ$. Observe that D is an orthogonal, block upper triangular matrix with 2-by-2 blocks. Hence, D is block diagonal with 2-by-2 blocks. \square

The following algorithm summarizes the procedure for simultaneous diagonalization of a commuting pair of real normal matrices.

ALGORITHM 2.

INPUT: $\epsilon > 0$; $A, B \in \mathbf{R}^{n \times n}$ such that $AB = BA$, $AA^T = A^T A$ and $BB^T = B^T B$

OUTPUT: $Q \in \mathbf{R}^{n \times n}$ such that $\text{off}_B(Q^T A Q, Q^T B Q) \leq \epsilon(\|A\|_F + \|B\|_F)$ and $QQ^T = I$.

1. $Q \leftarrow I$

2. **WHILE** $\text{off}_B(A, B) > \epsilon(\|A\|_F + \|B\|_F)$

• **FOR** $i = 1, 2, 3, \dots, \lfloor \frac{n+1}{2} \rfloor$

– **FOR** $j = i + 1, i + 2, i + 3, \dots, \lfloor \frac{n+1}{2} \rfloor$

* Select an inner block rotation $Z \leftarrow Z(i, j, U)$ that minimizes $\text{off}_B(Z^T A Z, Z^T B Z)$

* $Q \leftarrow QZ$; $A \leftarrow Z^T A Z$; $B \leftarrow Z^T B Z$

If rotations are stored and applied in the efficient manner described in [22, §5.1] or [33, §6.4], then each sweep (Step 2) uses approximately $8n^3$ real flops to update A and B and approximately $2n^3$ real flops to accumulate Q . A real flop is the computation effort required to execute the Fortran statement (8) where **A** and **S** are of type **REAL**. The rough estimate that one complex flop is equivalent to four real flops implies that Algorithm 2 does about half the work of Algorithm 1.

Algorithm 1 also needs storage for approximately $3n^2$ real numbers. This can be shaved to $2n^2$ real numbers if Q is not required.

5. Quaternions. A matrix $H \in \mathbf{C}^{2n \times 2n}$ has *quaternion* structure if it is of the form

$$(34) \quad JH = \bar{H}J$$

where $J \in \mathbf{R}^{2n \times 2n}$ is defined by

$$(35) \quad J = \text{diag}(E, E, E, \dots, E)$$

and

$$E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

If a quaternion matrix H is partitioned into an n -by- n block matrix with 2-by-2 blocks, then each block is of the form

$$(36) \quad H_{ij} = \begin{bmatrix} u_{ij} & -\bar{v}_{ij} \\ v_{ij} & \bar{u}_{ij} \end{bmatrix}$$

where $u_{ij}, v_{ij} \in \mathbf{C}$. It is easy to show that (34) and (36) are equivalent.

If $P \in \mathbf{R}^{2n \times 2n}$ is the permutation matrix

$$P = [e_1, e_3, e_5, \dots, e_{2n-1}, e_2, e_4, e_6, \dots, e_{2n}],$$

then

$$(37) \quad PHP^T = \begin{bmatrix} U & -\bar{V} \\ V & \bar{U} \end{bmatrix}.$$

In other treatments, e.g., [7], quaternion matrices are defined to have the form of (37), but in this context, (36) makes the explication somewhat simpler. Quaternion matrices arise naturally from quantum mechanical problems that have time reversal symmetry [12, 26, 34], and in eigenvalue problems that have two special structures [6].

The eigenvalues and eigenvectors of quaternion matrices appear in pairs. If (λ, x) is an eigenvalue-eigenvector pair of a quaternion matrix, then $(\bar{\lambda}, J\bar{x})$ is also an eigenvalue-eigenvector pair. A quaternion matrix H has a quaternion, Schur decomposition as $HQ = QT$ where $Q \in \mathbf{C}^{2n \times 2n}$ is a unitary, quaternion matrix and $T \in \mathbf{C}^{2n \times 2n}$ is an upper triangular, quaternion matrix [7]. If $H \in \mathbf{C}^{2n \times 2n}$ is also normal, then T is a diagonal, quaternion matrix. From this it is easy to show the following theorem.

THEOREM 5.1. *If $A, B \in \mathbf{C}^{2n \times 2n}$ are normal, commuting, quaternion matrices, then there is a unitary, quaternion matrix $Q \in \mathbf{C}^{2n \times 2n}$ and quaternion, diagonal matrices $D_A, D_B \in \mathbf{C}^{2n \times 2n}$ such that $AQ = QD_A$ and $BQ = QD_B$.*

Algorithm 1 applied to a commuting pair of normal, quaternion matrices will destroy the quaternion structure. General $2n$ -by- $2n$ matrices must be stored and updated. Rounding errors may destroy the pairing of eigenvalues and eigenvectors. Algorithm 1 takes no advantage of the pairing of eigenvalues.

Theorem 5.1 suggests that if quaternion structure were preserved throughout Algorithm 1, then work and storage requirements would be cut in half and the special pairing of eigenvalues and eigenvectors will be preserved despite rounding errors.

The key to preserving quaternion structure is the observation that products and inverses of quaternion matrices are quaternion matrices. In particular, there is a rich class of quaternion, unitary matrices to use in a modified version of Algorithm 1.

A quaternion rotation is a quaternion unitary matrix $W \in \mathbf{C}^{2n \times 2n}$ of the form $W = W_1 W_2 W_3$ where

$$(38) \quad W_1 = R(2i, 2j, \bar{c}_4, \bar{s}_4)R(2i-1, 2j-1, c_4, s_4),$$

$$(39) \quad W_2 = R(2j-1, 2j, c_3, s_3)R(2i-1, 2i, c_2, s_2)$$

and

$$(40) \quad W_3 = R(2i, 2j, \bar{c}_1, \bar{s}_1)R(2i-1, 2j-1, c_1, s_1).$$

The matrices W_1, W_2, W_3 are quaternion, unitary matrices. (The two rotations in (39) are quaternion, but the individual rotations in (38) and (40) are not.) Hence, W is quaternion and unitary. In terms of the block 2-by-2 partitioning (29), for

$1 \leq i < j \leq k = n$, the quaternion rotation $W = W(i, j, U)$ is of the form

$$(41) \quad W = \begin{bmatrix} I & & & & & \\ & \ddots & & & & \\ & & I & & & \\ & & & W_{ii} & & W_{ij} \\ & & & & I & \\ & & & & & \ddots \\ & & & & & & I \\ & & & W_{ji} & & & & W_{jj} \\ & & & & & & & & I \\ & & & & & & & & & \ddots \\ & & & & & & & & & & I \end{bmatrix},$$

where

$$(42) \quad \begin{bmatrix} W_{ii} & W_{ij} \\ W_{ji} & W_{jj} \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \\ = R(2, 4, \bar{c}_4, \bar{s}_4)R(1, 3, c_4, s_4) \\ R(3, 4, c_3, s_3)R(1, 2, c_2, s_2) \\ R(2, 4, \bar{c}_1, \bar{s}_1)R(1, 3, c_1, s_1).$$

Up to column scaling, quaternion unitary matrices are products of quaternion rotations. This was proved for symplectic matrices in the permuted form (37) in [32], but the methods carry over to the quaternion case with few modifications.

Suppose $A, B \in \mathbf{C}^{2n \times 2n}$ form a commuting pair of normal quaternion matrices. In broad outline, Algorithm 3 (presented below) is a block version of Algorithm 1 partitioned into 2-by-2 blocks. It measures progress with a block version of (2) defined by

$$\text{off}_Q(A, B) = \sum_{i \neq j} \|A_{ij}\|_F^2 + \|B_{ij}\|_F^2.$$

At the (i, j) -th step of a sweep, it uses a quaternion rotation $W = W(i, j, U)$ to minimize $\text{off}_Q(W^H A W, W^H B W)$.

Partitioning U as in (42), it is easy to verify that for $i < j$,

$$(43) \quad \begin{aligned} \text{off}_Q(W^H A W, W^H B W) = & \text{off}_Q(A, B) - \|A_{ij}\|_F^2 - \|A_{ji}\|_F^2 - \|B_{ij}\|_F^2 - \|B_{ji}\|_F^2 \\ & + \|U_{12}^H A_{ii} U_{11} + U_{22}^H A_{ji} U_{11} + U_{12}^H A_{ij} U_{21} + U_{22}^H A_{jj} U_{21}\|_F^2 \\ & + \|U_{11}^H A_{ii} U_{12} + U_{21}^H A_{ji} U_{12} + U_{11}^H A_{ij} U_{22} + U_{21}^H A_{jj} U_{22}\|_F^2 \\ & + \|U_{12}^H B_{ii} U_{11} + U_{22}^H B_{ji} U_{11} + U_{12}^H B_{ij} U_{21} + U_{22}^H B_{jj} U_{21}\|_F^2 \\ & + \|U_{11}^H B_{ii} U_{12} + U_{21}^H B_{ji} U_{12} + U_{11}^H B_{ij} U_{22} + U_{21}^H B_{jj} U_{22}\|_F^2 \end{aligned}$$

Minimizing (43) is equivalent to minimizing the sum of the last four terms. General optimization algorithms like those described in [19] provide effective ways to find minimizers $(c_l, s_l) = (\cos(\theta_l), \sin(\theta_l))$ for the parameters in (38), (39) and (40).

For $U \in \mathbf{R}^{4 \times 4}$ partitioned as in (42) define \tilde{U} by

$$\tilde{U} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}.$$

Observe that $W = W(i, j, U)$ and $S = W(i, j, \tilde{U})$ are both quaternion rotations and $\text{off}_Q(W^H A W, W^H B W) = \text{off}_Q(S^H A S, S^H B S)$. Thus, we may choose the quaternion rotation so that $\|U_{11}\|_F \geq \|U_{21}\|_F$. Such block rotations are sometimes called “inner” block rotations. The use of inner block rotations promotes convergence [18].

The following modification of Algorithm 1 summarizes the procedure for simultaneous diagonalization of a commuting pair of quaternion normal matrices. Note that the output of Algorithm 3 includes a pair of block diagonal matrices with 2-by-2 blocks. The trivial step of simultaneously diagonalizing the commuting 2-by-2 blocks is omitted.

ALGORITHM 3.

INPUT: $\epsilon > 0$; $A, B \in \mathbf{C}^{2n \times 2n}$ such that $AB = BA$, $JA = \bar{A}J$, $JB = \bar{B}J$, $AA^H = A^H A$ and $BB^H = B^H B$

OUTPUT: $Q \in \mathbf{C}^{2n \times 2n}$ such that $\text{off}_Q(Q^H A Q, Q^H B Q) \leq \epsilon(\|A\|_F + \|B\|_F)$, $JQ = \bar{Q}J$ and $QQ^H = I$.

1. $Q \leftarrow I$
2. **WHILE** $\text{off}_Q(A, B) > \epsilon(\|A\|_F + \|B\|_F)$
 - **FOR** $i = 1, 2, 3, \dots, \frac{n}{2}$
 - **FOR** $j = i + 1, i + 2, i + 3, \dots, \frac{n}{2}$
 - * Select an inner quaternion rotation $W \leftarrow W(i, j, U)$ that minimizes $\text{off}_Q(W^H A W, W^H B W)$
 - * $Q \leftarrow QZ$; $A \leftarrow W^H A W$; $B \leftarrow W^H B W$

Note that Q is a product of quaternion rotations, so Q is a quaternion unitary matrix. The quaternion structure of A and B is preserved throughout. Equations (36) and (37) show that only $2n^2$ complex numbers are needed to represent a quaternion matrix. With this economy, Algorithm 3 needs approximately $6n^2$ storage. If rotations are stored and applied in an efficient manner similar to [22, §5.1] or [33, §6.4], then each sweep (Step 2) uses approximately $24n^3$ complex flops to update A and B and approximately $6n^3$ complex flops to accumulate Q .

6. Additional Symmetry Structure. In [6], simultaneous diagonalization problems usually have an additional special structure in addition to normality. At times, A and B are Hermitian, real symmetric, real skew symmetric, or quaternion. In this section, we outline how to modify Algorithm 1 to take advantage of additional special structure.

6.1. Hermitian Case. If $(A, B) \in \mathbf{C}^{n \times n} \times \mathbf{C}^{n \times n}$ is a pair of commuting Hermitian matrices, then A is the Hermitian part and iB is the skew Hermitian part of the normal matrix $H = A + iB$. All normal matrices are of this form. Algorithm 1 reduces to the Jacobi method for normal matrices for which explicit formulae for minimizers of (5) are known [21].

6.2. Real Symmetric Case. The following theorem shows that if A and B are real symmetric, then no complex arithmetic is required by Algorithm 1. Moreover, there is an explicit formula for the minimizer of (5).

THEOREM 6.1. *Suppose $A, B \in \mathbf{R}^{n \times n}$, $A = A^T$, $B = B^T$ and $AB = BA$. If $c, s \in \mathbf{R}$, $c^2 + s^2 = 1$, and $w = \begin{bmatrix} c^s - s^2 \\ 2cs \end{bmatrix}$ is a right singular vector of $L_{ij} =$*

$\begin{bmatrix} a_{ij} & \frac{a_{ii}-a_{jj}}{2} \\ b_{ij} & \frac{a_{ii}-b_{jj}}{2} \end{bmatrix}$ corresponding to its smallest singular value, then (c, s) is a minimizer of (5). Furthermore, if L_{ij} has distinct singular values and $(\tilde{c}, \tilde{s}) \in \mathbf{C} \times \mathbf{C}$ minimizes (5), then $\tilde{w} = \begin{bmatrix} \tilde{c}s - \tilde{s}^2 \\ 2\tilde{c}s \end{bmatrix}$ is a scalar multiple of w .

Proof. In this case, (5) simplifies to

$$(44) \quad f_{ij}(c, s) = \sqrt{2} \|L_{ij} w\|_2 =_{\text{def}} \sqrt{2} \left\| \begin{bmatrix} a_{ij} & \frac{a_{ii}-a_{jj}}{2} \\ b_{ij} & \frac{a_{ii}-b_{jj}}{2} \end{bmatrix} \begin{bmatrix} c^2 - s^2 \\ 2cs \end{bmatrix} \right\|_2.$$

Using the double angle formulae for $s = \sin(\theta)$ and $c = \cos(\theta)$, any two-dimensional real vector of Euclidean length 1 may be written in the form of w for some choice of $c, s \in \mathbf{R}$, $c^2 + s^2 = 1$ and $c \geq s$. In particular, there is a choice of c and s such that w is a right singular vector of L_{ij} corresponding to the smallest singular value of L_{ij} . Note that L_{ij} is real, so w may be chosen to be real. Clearly, (44) is bounded below by $\sigma_{\min}(L_{ij})$, so this choice of c and s minimizes $f_{ij}(c, s)$. If L_{ij} has distinct singular values then its singular vectors are unique up to scalar multiples. \square

In addition to simplifying the calculation of the minimizing c and s , the theorem shows that only real rotations are needed. The real symmetric structure of A and B is preserved throughout. Consequently, only the real part of the upper triangle of A and B and the real part of Q need to be stored and updated. These modifications cut storage requirements of Algorithm 1 by a factor of three. The work requirements are cut to $6n^3$ real flops per sweep.

6.3. Real Symmetric — Real Skew-symmetric. If $A \in \mathbf{R}^{n \times n}$ is symmetric, $B \in \mathbf{R}^{n \times n}$ is skew-symmetric and $AB = BA$, then $H = A + B \in \mathbf{R}^{n \times n}$ is normal. All real, normal matrices are of this form. In this case, Algorithm 2 applies with $A := A + B$ and $B := 0$. Note that $B := 0$ is invariant throughout Algorithm 2, so no work need to expended updating B and no storage need be allocated to store B . This cuts the cost of Algorithm 2 down to approximately $4n^3$ real flops per sweep to update $A := A + B$ and approximately $2n^3$ flops per sweep to update Q . A similar algorithm using a different set of block rotations was suggested in [42]

6.4. Real Skew-symmetric — Real skew-symmetric. The skew-symmetric structure of A and B is preserved throughout Algorithm 2. It is necessary to store and update only the upper triangular part of A and B . This cuts the work required by Algorithm 2 to the same level as the symmetric — skew-symmetric case.

6.5. Hermitian, Quaternion. If (A, B) are Hermitian, quaternion matrices, then A is the Hermitian part and iB is the skew-Hermitian part of the normal matrix $H = A + iB$. However, H is not quaternion. (The matrix iB is antiquaternion [7]. See Subsection 6.6.) Nevertheless, Algorithm 3 preserves both the quaternion and Hermitian structure, only the upper triangles of A and B need be stored and updated. Thus, both the storage requirement for A and B and the work to update A and B are cut in half.

6.6. Hermitian Quaternion — Hermitian Anti-quaternion. A matrix $H \in \mathbf{C}^{2n \times 2n}$ is said to be antiquaternion if and only if iH is quaternion. If $A \in \mathbf{C}^{2n \times 2n}$ is Hermitian and quaternion and $B \in \mathbf{C}^{2n \times 2n}$ is Hermitian and antiquaternion, then A is the Hermitian part and iB is the skew-Hermitian part of the quaternion, normal matrix $H = A + iB$. Moreover, all quaternion, normal matrices are of this form. Algorithm 3 applies with $A := A + iB$ and $B := 0$. Note that $B := 0$ is invariant, so it

need not be stored or updated. This economy cuts the work and storage requirements of Algorithm 3 down to approximately $18n^3$ complex flops per sweep. If Q is not required, then the work requirement drops to $12n^3$ complex flops per sweep.

6.7. Commuting m -tuples. Algorithm 1 extends to the problem of simultaneously diagonalizing m commuting, normal matrices. In this case the minimization problem (5) uses a $2m$ -by-3 matrix coefficient matrix. Algorithm 1 must apply similarity transformations to all m commuting matrices.

7. Conclusions. We have presented a Jacobi-like algorithm for simultaneous diagonalization of commuting pairs of complex normal matrices. The algorithm uses a sequence of similarity transformations by elementary complex rotations to drive the off-diagonal entries to zero. Convergence and rounding error properties are similar to the serial Jacobi algorithm [20, 38, 46]. We have shown that its asymptotic convergence rate is quadratic, and that it is numerically stable in the sense that the computed eigenvalues and eigenvectors are correct for a perturbation of the data. Empirically, it appears to converge globally, but we have not been able to give a proof. The algorithm can easily be modified to preserve and exploit the additional special structure of real matrices, quaternion matrices and real symmetric matrices.

Acknowledgment. We would like to thank Gabriele Steinmeyer for help testing some of the algorithms and Prof. Vjeran Hari for comments on an early draft of this paper. We also thank an anonymous referee for bringing [45] to our attention, and Prof. K. Veselic for providing us with a copy of the paper and a translation.

REFERENCES

- [1] J. BARLOW AND J. DEMMEL, *Computing accurate eigensystems of scaled diagonally dominant matrices*, Tech. Rep. 421, Courant Institute of Mathematical Sciences, Department of Computer Science, New York University, 251 Mercer Street, New York, N. Y. 10012, December 1988.
- [2] M. BERRY AND A. H. SAMEH, *Multiprocessor Jacobi algorithms for dense symmetric eigenvalue and singular value decompositions*, in Proceedings of the '86 International Conference on Parallel Processing, August 1986.
- [3] ———, *A parallel algorithm for the singular value and dense symmetric eigenvalue problem*, tech. rep., University of Illinois at Urbana-Champaign, Center for Supercomputing Research and Development, Urbana-Champaign, Illinois, March 1988.
- [4] C. H. BISCHOF, *The two-sided block Jacobi method on a hypercube architecture*, in Hypercube Multiprocessors, M. T. Heath, ed., Philadelphia, PA, 1987, SIAM, pp. 612–618.
- [5] R. P. BRENT AND F. T. LUK, *The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays*, SIAM Journal of Scientific and Statistical Computing, 6 (1985), pp. 69–84.
- [6] A. BUNSE-GERSTNER, R. BYERS, AND V. MEHRMANN, *A chart of numerical methods for structured eigenvalue problems*, tech. rep., FSP Mathematisierung, Universität Bielefeld, Postfach 8640, D-4800 Bielefeld 1, FRG, 1989. To appear in SIAM Journal of Matrix Analysis.
- [7] ———, *A quaternion QR algorithm*, Numerische Mathematik, 55 (1989), pp. 83–95.
- [8] K. CHEN AND K. IRANI, *A Jacobi algorithm and its implementation on parallel computers*, in Proceedings of the 18-th Annual Allerton Conference on Communication Control and Computing, 1980.

- [9] R. O. DAVIES AND J. J. MODI, *A direct method for computing eigenproblem solutions on a parallel computer*, *Linear Algebra and Its Applications*, 77 (1986), pp. 61–74.
- [10] J. W. DEMMEL AND W. KAHAN, *Computing small singular values of bidiagonal matrices with guaranteed high relative accuracy*, *SIAM Journal of Scientific and Statistical Computing*, (to appear).
- [11] J. W. DEMMEL AND K. VESELIĆ, *Jacobi's method is more accurate than QR*, Tech. Rep. 468, Courant Institute of Mathematical Sciences, Department of Computer Science, New York University, 251 Mercer Street, New York, N. Y. 10012, October 1989.
- [12] J. J. DONGARRA, J. R. GABRIEL, D. D. KÖLLING, AND J. H. WILKINSON, *The eigenvalue problem for Hermitian matrices with time reversal symmetry*, *Linear Algebra and Its Applications*, 60 (1984), pp. 27–42.
- [13] P. J. EBERLEIN, *A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix*, *Journal of the Society for Industrial and Applied Mathematics*, 10 (1962), pp. 74–88.
- [14] P. J. EBERLEIN, *Solution to the complex eigenproblem by norm reducing Jacobi-type method*, *Numerische Mathematic*, 14 (1970), pp. 232–245.
- [15] ———, *On one-sided Jacobi methods for parallel computation*, *SIAM Journal on Algebraic and Discrete Methods*, 8 (1987), pp. 790–796.
- [16] ———, *On using the Jacobi method on the hypercube*, in *Second Conference on Hypercube Multiprocessors*, M. T. Heath, ed., Philadelphia, PA, 1987, SIAM, pp. 605–611.
- [17] A. ERDMANN, *Über Jacobi-ähnliche Verfahren zur Lösung des Eigenwertproblems nicht-normaler komplexer Matrizen*, Dissertation, Fernuniversität Hagen, Hagen, FRG, 1984.
- [18] K. V. FERNANDO AND S. J. HAMMARLING, *On block Kogbetliantz methods for computation of the SVD*, in *SVD and Signal Processing: Algorithms, Applications and Architectures*, F. Deprettere, ed., North-Holland, 1988.
- [19] R. FLETCHER, *Practical Methods of Optimization, Volumes 1 and 2*, John Wiley and Sons, New York, N.Y., 1981.
- [20] G. E. FORSYTHE AND P. HENRICI, *The cyclic Jacobi method for computing the principal values of a complex matrix*, *Transactions of the American Mathematical Society*, 94 (1960), pp. 1–23.
- [21] H. H. GOLDSTEIN AND L. P. HORWITZ, *A procedure for the diagonalization of normal matrices*, *Journal of the Association for Computing Machinery*, 6 (1959), pp. 176–195.
- [22] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins Press, Baltimore, Maryland, second ed., 1989.
- [23] V. HARI, *On sharp quadratic convergence bounds for the serial Jacobi methods*, tech. rep., University of Zagreb, Department of Mathematics, YU-41001 Zagreb, P. O. Box 187, Yugoslavia, 1989.
- [24] C. G. JACOBI, *Über ein leichtes Verfahren, die in der Theorie der Säkulärstörungen vorkommenden Gleichungen numerisch aufzulösen*, *Journal Reine Angewandte Mathematik*, (1846), pp. 51–95.
- [25] W. KAHAN, *Accurate eigenvalues of a symmetric tridiagonal matrix*, Tech. Rep. CS 41, Stanford University, Computer Science Department, Stanford, California, 1966. revised 1968.
- [26] M. LAX, *Symmetry Principles in Solid State and Molecular Physics*, John Wiley and Sons, New York, N. Y., first ed., 1974.
- [27] F. T. LUK, *A triangular processor array for computing singular values*, *Linear Algebra and Its Applications*, 77 (1986), pp. 259–273.
- [28] F. T. LUK AND H. P. , *A proof of convergence for two parallel Jacobi SVD algorithms*, *IEEE Transactions on Computers*, 38 (1989), pp. 806–811.
- [29] C. MOLER, *MATLAB user's guide*, Tech. Rep. CS81-1, Computer Science, University of New Mexico, Albuquerque, NM, 1980.
- [30] B. NOBLE AND W. DANIEL, *Applied Linear Algebra*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.
- [31] M. H. C. PAARDEKOOPER, *An eigenvalue algorithm for skew-symmetric matrices*, *Numerische Mathematik*, 17 (1971), pp. 189–202.
- [32] C. PAIGE AND C. F. VAN LOAN, *A Schur decomposition for Hamiltonian matrices*, *Linear Algebra and Its Applications*, 41 (1981), pp. 11–32.
- [33] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall Series in Computational Mathematics, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1980.
- [34] N. RÖSCH, *Time-reversal symmetry, Kramers' degeneracy and the algebraic eigenvalue problem*, *Chemical Physics*, 80 (1983), pp. 1–5.
- [35] W. RUDIN, *Real and Complex Analysis*, Higher Mathematics, McGraw-Hill Book Company, New York, N.Y., 1966.

- [36] A. RUHE, *On the quadratic convergence of the Jacobi method for normal matrices*, BIT, 7 (1967), pp. 305–313.
- [37] A. H. SAMEH, *On Jacobi and Jacobi-like algorithms for a parallel computer*, Mathematics of Computation, 25 (1971), pp. 579–590.
- [38] A. SCHÖNHAGE, *Zur Konvergenz des Jacobi-Verfahrens*, Numerische Mathematik, 3 (1961), pp. 374–380.
- [39] B. T. SMITH, J. M. BOYLE, J. J. DONGARRA, B. S. GARBOW, Y. IKEBE, V. C. KLEMA, AND C. B. MOLER, *Matrix Eigensystem Routines — EISPACK Guide*, Lecture Notes in Computer Science, Springer Verlag, New York, N.Y., 1976.
- [40] G. W. STEWART, *A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix*, SIAM Journal of Scientific and Statistical Computing, 6 (1985), pp. 853–864.
- [41] H. P. M. VAN KEMPEN, *On the convergence of the classical Jacobi method for real symmetric matrices with non-distinct eigenvalues*, Numerische Mathematik, 9 (1966), pp. 11–18.
- [42] K. VESELIĆ, *On a new class of elementary matrices*, Numerische Mathematik, 33 (1979), pp. 173–180.
- [43] ———, *An eigenreduction algorithm for definite matrix pairs and its applications to overdamped linear systems*, tech. rep., FernUniversität, Fakultät für Mathematik, Fernuniversität, Postfach 940, D-5800 Hagen, FRG, 1988.
- [44] K. VESELIĆ AND H. J. WENZEL, *A quadratically convergent Jacobi-like method for real matrices with complex eigenvalues*, Numerische Mathematik, 33 (1979), pp. 425–435.
- [45] V. VOEVODIN, *An extension of the method of Jacobi (in russian)*, Comp. Methods and Programming VIII, Comp. Center of the Moscow University, (1967), pp. 216–228.
- [46] J. H. WILKINSON, *Note on the quadratic convergence of the cyclic Jacobi process*, Numerische Mathematik, 4 (1962), pp. 296–300.
- [47] ———, *The Algebraic Eigenvalue Problem*, Clarendon Press, Walton Street, Oxford, U. K., 1965.
- [48] ———, *Almost diagonal matrices with multiple or close eigenvalues*, Tech. Rep. CS 59, Stanford University, Department of Computer Science, Stanford, California, 1967.