An Efficient Method for Computing Eigenvalues of a Real Normal Matrix*

B. B. Zhou

School of Computing & Mathematics
Deakin University
Geelong, VIC 3217, Australia

R. P. Brent

Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford OX1 3QD, UK

Abstract

Jacobi-based algorithms have attracted attention as they have a high degree of potential parallelism and may be more accurate than QR-based algorithms. In this paper we discuss how to design efficient Jacobi-like algorithms for eigenvalue decomposition of a real normal matrix. We introduce a block Jacobi-like method. This method uses only real arithmetic and orthogonal similarity transformations and achieves ultimate quadratic convergence. A theoretical analysis is conducted and some experimental results are presented.

Keywords Eigenvalue decomposition, normal matrix, Jacobi algorithm and QR algorithm, parallel computing.

1 Introduction

A real matrix A is said to be normal if it satisfies the equation

$$AA^T = A^T A$$

where A^T is the transpose of matrix A. A normal matrix can be reduced to diagonal form using unitary similarity transformations

$$Q^H AQ = D$$

where D is diagonal, Q is unitary and Q^H is the conjugate transpose of Q. The standard sequential method for eigenvalue decomposition of this kind of matrix is the QR algorithm. However, when massively parallel computation is considered, the parallel version of the QR-based algorithms for solving unsymmetric eigenvalue problems may not be very efficient because the algorithms are difficult to scale.

One alternative to the QR method is a Jacobi method. Jacobi-based algorithms have recently attracted attention as they have a higher degree of potential parallelism and may be more accurate than QR-based algorithms [4]. The Jacobi method, though originally designed for symmetric eigenvalue problems, can be extended to solve eigenvalue problems for unsymmetric normal matrices [5, 23]. A problem is that we have to use complex arithmetic even for real-valued normal matrices. Complex operations are expensive and should be avoided if possible. A quaternion-Jacobi method was introduced in [17]. In this method a 4×4 symmetric matrix can be reduced to a 2×2 block diagonal form using one orthogonal similarity transformation. This method can also be extended to compute eigenvalues of a general normal matrix. Difficulties

*To appear in *Journal of Parallel and Distributed Computing* (accepted 11 November 2002). rpb207

with this method are that the original matrix has to be divided into a sum of a symmetric matrix and a skew-symmetric matrix, and that the algorithm cannot be used to solve the eigenvalue problem of near-normal matrices. Another parallel Jacobi-like algorithm, named the RTZ (Real Two-Zero) algorithm, was proposed in [18]. This method uses real arithmetic and orthogonal similarity transformations. It is claimed that quadratic convergence can be obtained when computing eigenvalues of a real near-normal matrix with real distinct eigenvalues. However, a serious difficulty with this algorithm is that the process may fail to converge if the matrix has complex eigenvalues.

In this paper we discuss a block Jacobi-like method for computing the eigenvalue decomposition of a real normal matrix. The method uses only real arithmetic and orthogonal similarity transformations. The theoretical analysis and experimental results show that ultimate quadratic convergence can be achieved for general real normal matrices with distinct eigenvalues. Our aim is to design scalable algorithms which can be implemented efficiently on parallel machines.

Real normal matrices are generalisations of real symmetric matrices. A real symmetric matrix is normal, but a real normal matrix is not necessarily symmetric. We focus our attention on the unsymmetric case although the method to be described applies to both cases.

The paper is organised as follows. The basic idea of the block method is described in Section 2. In Section 3 we give a theoretical analysis which indicates how to choose orthogonal similarity transformations so that ultimate quadratic convergence can be obtained. Some experimental results are presented in Section 4. In Section 5 we describe a parallel ring Jacobi ordering which is equivalent to the sequential cyclic Jacobi ordering in terms of both ordering and sorting and can thus directly be applied for parallel implementation of our block method. Finally, the conclusions are given in Section 6.

2 Basic Idea

Before discussing the basic idea of our block method, we first sketch a proof of the following lemma. Here $\|A\|$ denotes the Frobenius norm, and ϵ is a small parameter.

Lemma 1 *Let a normal matrix A be divided into blocks as follows:*

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} \end{pmatrix}. \tag{1}$$

where the size of each block A_{ij} is $p \times p$.

If A is nearly block lower triangular, that is,

$$\sum_{k>i} \|A_{ki}\|^2 = O(\epsilon), \quad \textit{for all } i,$$

then A will be nearly block diagonal. Similarly if A is nearly block upper triangular.

Proof. Since the matrix A is normal, and $(AA^T)_{ii} = \sum_k A_{ik}A_{ik}^T$ and $(A^TA)_{ii} = \sum_k A_{ki}^TA_{ki}$, where $(X)_{ij}$ denotes the $(i,j)^{th}$ block of matrix X, we have

$$(AA^T - A^TA)_{ii} = \sum_k A_{ik}A_{ik}^T - \sum_k A_{ki}^TA_{ki} = 0. \tag{2}$$

Taking the sum on both sides of the above equation, we obtain the following relation for all i:

$$\sum_{k \neq i} (\|A_{ki}\|^2 - \|A_{ik}\|^2) = 0. \tag{3}$$

When $i = 1$, this implies that

$$\sum_{k>1} \|A_{1k}\|^2 = \sum_{k>1} \|A_{k1}\|^2 = O(\epsilon).$$

When $i = 2$, we have

$$\sum_{k>2} \|A_{2k}\|^2 + \|A_{21}\|^2 = \sum_{k>2} \|A_{k2}\|^2 + \|A_{12}\|^2,$$

or

$$\begin{aligned} \sum_{k>2} \|A_{2k}\|^2 &= \sum_{k>2} \|A_{k2}\|^2 + \|A_{12}\|^2 - \|A_{21}\|^2 \\ &\leq \sum_{k>2} \|A_{k2}\|^2 + \sum_{k>1} \|A_{1k}\|^2 \\ &= O(\epsilon). \end{aligned}$$

Proceeding in this way, we can easily prove by induction that

$$\sum_{k>i} \|A_{ik}\|^2 = O(\epsilon).$$

□

From the above lemma we see that, if we can reduce a real normal matrix to a block upper triangular form using orthogonal similarity transformations, the matrix will become block diagonal. The initial idea of our method is as follows. To avoid using complex arithmetic, we choose the size of each block to be 2×2 (set $p = 2$ in (1)) and each block A_{ij} will have the form

$$A_{ij} = \begin{pmatrix} a_{2i-1,2j-1} & a_{2i-1,2j} \\ a_{2i,2j-1} & a_{2i,2j} \end{pmatrix} \tag{4}$$

where $a_{2i-x,2j-y}$ (for $i \geq 1$, $j \geq 1$, $0 \leq x \leq 1$ and $0 \leq y \leq 1$) are scalar elements in the original matrix A. We apply a sequence of orthogonal similarity transformations (purely real arithmetic) to annihilate the off-diagonal blocks only in the lower triangle. If the procedure converges, a real normal matrix will become block diagonal with each pair of conjugate complex eigenvalues being grouped in the same block. A key problem is how to choose the orthogonal transformation matrices during the computation. If the orthogonal transformation matrices are not chosen properly, quadratic convergence may not be guaranteed. This is the main topic that we discuss in the next two sections.

It should be noted that the convergence problem is directly related to ordering schemes for reducing a matrix to a triangular form. The key to design a good parallel Jacobi (or Jacobi-like) algorithm is to design a parallel Jacobi ordering. Fortunately, many existing schemes for parallel Jacobi ordering, for example those described in [2, 3, 16, 22, 26, 27], have been proved to be equivalent to the well-known sequential cyclic ordering [16, 26]. If we can prove that our method works well with the cyclic Jacobi ordering, it will then work well with equivalent parallel Jacobi ordering schemes. To simplify our discussion, therefore, in the next two sections we only use the cyclic Jacobi ordering. In Section 5 we show how to implement our method on a parallel machine using a parallel ring Jacobi ordering.

The basic structure of our block Jacobi-like method using cyclic Jacobi ordering is sketched in Fig. 1. In this figure the value of totalBlocks is equal to the number of lower triangular off-diagonal blocks, or number of iterations in a sweep, which is equal to $n(n-2)/8$ for n the size of the original matrix A. (We assume that padding is used if necessary to ensure that n is even.) Because the size of each block is 2×2, the submatrix B is then of size 4×4. A counter zeroBlocks is used to check how many off-diagonal blocks in the lower triangle are zero to working accuracy. The process stops if the value of this counter equals totalBlocks in a complete sweep.

totalBlocks = $n(n-2)/8$
zeroBlocks = 0
REPEAT UNTIL zeroBlocks = totalBlocks
  DO $i = 1, n/2$
    DO $j = i+1, n/2$
      $B = \begin{pmatrix} A_{ii} & A_{ij} \\ A_{ji} & A_{jj} \end{pmatrix}$
      IF ($\|A_{ji}\|$ is not zero to working accuracy) **THEN**
        1. Find an orthogonal matrix Q such that B is reduced to a block triangular form through a similarity transformation.
        2. Update the corresponding rows and columns of A using Q.
      ELSE
        zeroBlocks = zeroBlocks + 1
      END IF
    END DO
  END DO
  IF (zeroBlocks $\neq$ totalBlocks) zeroBlocks = 0
END REPEAT

Figure 1: The basic structure of our block Jacobi-like method (with orthogonal transformation matrices Q to be determined).

3 Theoretical Analysis

We show in the next three lemmas that, if the orthogonal transformation matrices are chosen properly and if the process converges for a normal matrix with distinct eigenvalues, then the convergence rate will ultimately be quadratic.

In the following discussion submatrix B has a form as in Fig. 1. Submatrix B_{ij} is also a block matrix in the same form as B, that is, when the original matrix is divided into a block matrix with each block A_{ij} of size 2×2, B_{ij} will be a 2×2 block submatrix in the same form as B. The difference is that B in Fig. 1 represents the block submatrix on the main diagonal, but B_{ij} is more general.

Lemma 2 *If a normal matrix A is divided into blocks B_{ij} then the main diagonal blocks have the following property:*

$$\|B_{ii}B_{ii}^T - B_{ii}^T B_{ii}\| \leq \sum_{k \neq i} (\|B_{ki}\|^2 + \|B_{ik}\|^2). \tag{5}$$

Proof. Since the matrix A is normal, we have

$$(AA^T - A^T A)_{ii} = \sum_k B_{ik}B_{ik}^T - \sum_k B_{ki}^T B_{ki} = 0$$

or

$$B_{ii}B_{ii}^T - B_{ii}^T B_{ii} = \sum_{k \neq i} (B_{ki}^T B_{ki} - B_{ik}B_{ik}^T).$$

Then

$$\begin{aligned}\|B_{ii}B_{ii}^T - B_{ii}^T B_{ii}\| &\leq \sum_{k\neq i}\|B_{ki}^T B_{ki} - B_{ik}B_{ik}^T\| \\ &\leq \sum_{k\neq i}(\|B_{ki}^T B_{ki}\| + \|B_{ik}B_{ik}^T\|).\end{aligned}$$

Since $\|B_{ki}^T B_{ki}\| \leq \|B_{ki}^T\|\|B_{ki}\| = \|B_{ki}\|^2$, we thus have

$$\|B_{ii}B_{ii}^T - B_{ii}^T B_{ii}\| \leq \sum_{k\neq i}(\|B_{ki}\|^2 + \|B_{ik}\|^2).$$

□

It can be seen from above lemma that, if the norm of every off-diagonal block is small, each block matrix B_{ii} on the main diagonal will be close to normal, that is, $\|B_{ii}B_{ii}^T - B_{ii}^T B_{ii}\| = O(\epsilon^2)$ if $\max(\|B_{ij}\|) = \epsilon$ for $i \neq j$.

The next lemma shows that, when a matrix of size 4×4 is reduced to a block triangular form, with the size of each block being 2×2, through an orthogonal similarity transformation, the norm of its upper off-diagonal block will also be decreased if the matrix is close to normal.

Lemma 3 *Assume that matrix B of size 4×4, is close to normal and has the property*

$$\|BB^T - B^T B\| = O(\eta) \tag{6}$$

where η is a small positive number, and that B has four nonzero eigenvalues and can be reduced to a block triangular form through an orthogonal similarity transformation, that is,

$$Q^T B Q = D \tag{7}$$

where Q is a real orthogonal matrix and

$$D = \begin{pmatrix} D_{11} & D_{12} \\ 0 & D_{22} \end{pmatrix} \tag{8}$$

for each block D_{ij} being of size 2×2.

Let the eigenvalues of D_{11} and D_{22} be λ_{1i} and λ_{2i} for $i = 1, 2$, respectively. (These are also the eigenvalues of B.) If these eigenvalues are bounded and the eigenvalues of D_{11} are separated from those of D_{22}, that is,

$$\left|1 - \frac{\lambda_{1i}}{\lambda_{2j}}\right| > c \tag{9}$$

for c some positive constant, then we have

$$\|D_{12}\| = O(\eta).$$

Proof. Since $B = QDQ^T$ and $B^T = QD^TQ^T$, we have

$$\|BB^T - B^T B\| = \|Q(DD^T - D^T D)Q^T\|.$$

Thus

$$\|DD^T - D^T D\| = O(\eta). \tag{10}$$

It is easy to see that the second block of the first column in $DD^T - D^T D$ is $D_{22}D_{12}^T - D_{12}^T D_{11}$. We know that $\|D_{12}\| \neq 0$. Otherwise, B becomes a normal matrix. Since $\|D_{12}\| \neq 0$ and the eigenvalues are nonzero and distinct, $\|D_{22}D_{12}^T - D_{12}^T D_{11}\|$ may be nonzero. However, it is easy to see from (10) that it must, when being divided into blocks, be of order η at most, that is,

$$\|D_{22}D_{12}^T - D_{12}^T D_{11}\| = O(\eta). \tag{11}$$

Matrix B has four nonzero eigenvalues and both D_{11} and D_{22} have full rank. The following inequality holds:

$$\|D_{12}^T - D_{22}^{-1} D_{12}^T D_{11}\| \leq \|D_{22}^{-1}\| \|D_{22}D_{12}^T - D_{12}^T D_{11}\|. \tag{12}$$

Since the eigenvalues of matrix B are bounded and nonzero, the eigenvalues of $\|D_{22}^{-1}\|$ must also be bounded. We then obtain

$$\|D_{12}^T - D_{22}^{-1} D_{12}^T D_{11}\| = O(\eta). \tag{13}$$

Let

$$D_{11} = Q_1 R_1 Q_1^H$$

and

$$D_{22}^{-1} = Q_2 R_2 Q_2^H$$

be the eigenvalue decomposition of D_{11} and D_{22}^{-1} for Q_1 and Q_2 unitary and R_1 and R_2 upper triangular, and define

$$Q_2^H D_{12}^T Q_1 = E.$$

we then have

$$\begin{aligned} \|E - R_2 E R_1\| &= \|Q_2(E - R_2 E R_1) Q_1^H\| \\ &= \|D_{12}^T - D_{22}^{-1} D_{12}^T D_{11}\| \\ &= O(\eta). \end{aligned} \tag{14}$$

Let

$$E = \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{pmatrix},$$

$$R_1 = \begin{pmatrix} r_{11}^{(1)} & r_{12}^{(1)} \\ 0 & r_{22}^{(1)} \end{pmatrix}$$

and

$$R_2 = \begin{pmatrix} r_{11}^{(2)} & r_{12}^{(2)} \\ 0 & r_{22}^{(2)} \end{pmatrix}.$$

Expanding $G = E - R_2 E R_1$, we obtain

$$G = \begin{pmatrix} s_{11}e_{11} - r_{12}^{(2)} r_{11}^{(1)} e_{21} & s_{12}e_{12} - r_{11}^{(2)} r_{12}^{(1)} e_{11} - r_{12}^{(2)} r_{12}^{(1)} e_{21} - r_{12}^{(2)} r_{22}^{(1)} e_{22} \\ s_{21}e_{21} & s_{22}e_{22} - r_{22}^{(2)} r_{12}^{(1)} e_{21} \end{pmatrix}$$

where $s_{ji} = 1 - r_{jj}^{(2)} r_{ii}^{(1)}$.

However, $r_{ii}^{(1)}$ is an eigenvalue of D_{11} and $1/r_{jj}^{(2)}$ is an eigenvalue of D_{22}. From (9), we have

$$\begin{aligned} |s_{ji}| &= \left|1 - r_{jj}^{(2)} r_{ii}^{(1)}\right| \\ &= \left|1 - \frac{\lambda_{1i}}{\lambda_{2j}}\right| \\ &> c. \end{aligned}$$

According to equation (14) all the elements in G must be of order η, that is,

$$\begin{aligned}
|s_{21}e_{21}| &= O(\eta); \\
\left|s_{11}e_{11} - r_{12}^{(2)}r_{11}^{(1)}e_{21}\right| &= O(\eta); \\
\left|s_{22}e_{22} - r_{22}^{(2)}r_{12}^{(1)}e_{21}\right| &= O(\eta); \\
\left|s_{12}e_{12} - r_{11}^{(2)}r_{12}^{(1)}e_{11} - r_{12}^{(2)}r_{12}^{(1)}e_{21} - r_{12}^{(2)}r_{22}^{(1)}e_{22}\right| &= O(\eta).
\end{aligned}$$

From the first equation we can obtain $|e_{21}| = O(\eta)$ because $|s_{21}|$ is greater than a constant c. After we know that $|e_{21}|$ is of order η, we can easily verify that $|e_{11}|$ (and $|e_{22}|$) is of order η from the second equation (and the third equation). Because $|e_{11}|$, $|e_{21}|$ and $|e_{22}|$ are all of order η and $|s_{12}| > c$, we obtain $|e_{12}| = O(\eta)$ from the fourth equation. Therefore, $\|E\|$ must be of order η and we have

$$\begin{aligned}
\|D_{12}\| &= \|Q_2 E Q_1^H\| \\
&= \|E\| \\
&= O(\eta).
\end{aligned}$$

□

Assume that a normal matrix is divided into blocks of size 2×2 and that the norm of each off-diagonal block is of order ϵ. According to Lemma 2 the norm of submatrix B in Fig. 1 must satisfy the equation

$$\|BB^T - B^T B\| = O(\epsilon^2).$$

After an orthogonal similarity transformation which reduces B to a block upper triangular matrix, the norm of the upper off-diagonal block in B should have the same order as $\|BB^T - B^T B\|$ according to the above lemma. Therefore, it will be of order ϵ^2. In the following we show that, if the orthogonal transformation matrices are chosen properly, ultimate quadratic convergence can be achieved.

Explicitly write matrices B and Q in (7) as 2×2 block matrices, that is,

$$B = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

and

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}$$

and assume that $\|A_{12}\| = O(\epsilon)$ and $\|A_{21}\| = O(\epsilon)$. Then we have the following lemma.

Lemma 4 *Assume that an orthogonal matrix is found such that*

$$\begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}^T \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} \\ 0 & D_{22} \end{pmatrix}. \tag{15}$$

Further assume that the eigenvalues of A_{11} and A_{22} are γ_{1k}, γ_{2k} for $k = 1, 2$, respectively, and the eigenvalues of D_{11} and D_{22} are λ_{1l} and λ_{2l} for $l = 1, 2$. If the eigenvalues γ_{ik} of A_{ii} and the eigenvalues λ_{jl} of D_{jj} satisfy the following inequality

$$\left|1 - \frac{\gamma_{ik}}{\lambda_{jl}}\right| > c \tag{16}$$

for $i \neq j$, and c being a constant greater than zero, then the norms of both Q_{12} and Q_{21} in the generated orthogonal matrix must be of order ϵ, that is,

$$\|Q_{12}\| = O(\epsilon)$$

and

$$\|Q_{21}\| = O(\epsilon).$$

Proof. From (15) we have

$$A_{21}Q_{11} + A_{22}Q_{21} = Q_{21}D_{11},$$

or

$$A_{21}Q_{11} = Q_{21}D_{11} - A_{22}Q_{21},$$

or

$$A_{21}Q_{11}D_{11}^{-1} = Q_{21} - A_{22}Q_{21}D_{11}^{-1} \tag{17}$$

and

$$A_{11}Q_{12} + A_{12}Q_{22} = Q_{11}D_{12} + Q_{12}D_{22},$$

or

$$(A_{12}Q_{22} - Q_{11}D_{12})D_{22}^{-1} = Q_{12} - A_{11}Q_{12}D_{22}^{-1}. \tag{18}$$

Since $\|A_{21}\| = O(\epsilon)$, $\|A_{12}\| = O(\epsilon)$ and $\|D_{12}\| = O(\epsilon^2)$ which can be obtained by combining the results of Lemma 1 and Lemma 3, the norms of the left-side of the equations in (17) and (18) must be of order ϵ. We thus have

$$\|Q_{21} - A_{22}Q_{21}D_{11}^{-1}\| = O(\epsilon) \tag{19}$$

and

$$\|Q_{12} - A_{11}Q_{12}D_{22}^{-1}\| = O(\epsilon). \tag{20}$$

The above two equations have the same form as in (13). Adopting the same technique that we used in Lemma 3 to prove $\|D_{12}\| = O(\eta)$, we can easily obtain $\|Q_{21}\| = O(\epsilon)$ and $\|Q_{12}\| = O(\epsilon)$. (Note that as the inequality (9) played a key role in proving $\|D_{12}\| = O(\eta)$, so does the corresponding inequality (16) in proving this lemma.) $\square$

Using an orthogonal transformation matrix with $\|Q_{21}\| = O(\epsilon)$ and $\|Q_{12}\| = O(\epsilon)$ to update (premultiply) a vector $(0\ 0\ v_3\ v_4)^T$ (or $(v_1\ v_2\ v_3\ v_4)^T$ for v_1 and v_2 of order ϵ^2), the zero elements in the vector will become $O(\epsilon^2)$ (or v_1 and v_2 will retain their original order) if both v_3 and v_4 are of order ϵ. This is a key factor for obtaining ultimate quadratic convergence.

Consider a 4×4 block matrix A for each block being of size 2×2 and that after k sweeps the norm of each off-diagonal block becomes the order of ϵ, that is,

$$A^{(k)} = \begin{pmatrix} A_{11} & \epsilon_{12} & \epsilon_{13} & \epsilon_{14} \\ \epsilon_{21} & A_{22} & \epsilon_{23} & \epsilon_{24} \\ \epsilon_{31} & \epsilon_{32} & A_{33} & \epsilon_{34} \\ \epsilon_{41} & \epsilon_{42} & \epsilon_{43} & A_{44} \end{pmatrix}$$

where the off-diagonal block A_{ij} is represented by ϵ_{ij} which means $\|A_{ij}\|$ is of order ϵ.

Assume that the cyclic ordering is adopted in the computation. At the $(k+1)^{th}$ sweep A_{21} will be annihilated first. According to the structure depicted in Fig. 1, a submatrix B is formed, that is,

$$B = \begin{pmatrix} A_{11} & \epsilon_{12} \\ \epsilon_{21} & A_{22} \end{pmatrix}.$$

Since all off-diagonal blocks A_{ij} are of order ϵ, from Lemma 2 we have

$$\|BB^T - B^T B\| = O(\epsilon^2).$$

After an orthogonal similarity transformation on B to annihilate A_{21}, the norm of A_{12} will become $O(\epsilon^2)$ according to Lemma 3.

Next A_{31} is annihilated. For the same reason we have $\|A_{13}\| = O(\epsilon^2)$ after the orthogonal similarity transformation. However, A_{21} will become nonzero again. If the norms of the off-diagonal blocks of the generated orthogonal matrix are of order ϵ, we can guarantee $\|A_{21}\| = O(\epsilon^2)$ and $\|A_{12}\| = O(\epsilon^2)$ after the updating.

It can be seen from above discussion that $\|A_{ji}\|$ will become $O(\epsilon^2)$ after A_{ij} is annihilated for $i > j$ and during the successive transformations both $\|A_{ji}\|$ and $\|A_{ij}\|$ can remain of order ϵ^2 if at each step the orthogonal transformation matrix is chosen properly such that the norms of its off-diagonal blocks (of size 2×2) are of order ϵ. After the $(k+1)^{th}$ sweep the matrix will become

$$A^{(k+1)} = \begin{pmatrix} A_{11} & O(\epsilon^2) & O(\epsilon^2) & O(\epsilon^2) \\ O(\epsilon^2) & A_{22} & O(\epsilon^2) & O(\epsilon^2) \\ O(\epsilon^2) & O(\epsilon^2) & A_{33} & O(\epsilon^2) \\ O(\epsilon^2) & O(\epsilon^2) & O(\epsilon^2) & A_{44} \end{pmatrix}.$$

Therefore, asymptotic quadratic convergence is obtained.

Now the problem is how to choose the orthogonal transformation matrix so that the norms of its off-diagonal blocks are of order ϵ. The inequality (16) plays a crucial role in proving Lemma 4. When B is close to normal, the eigenvalues of A_{11} and A_{22}, the diagonal blocks of B, will be very close to the true eigenvalues of B. If an orthogonal transformation also involves (either implicitly or explicitly) a permutation on rows and columns between blocks, the eigenvalues of D_{22} can be very close to those of A_{11}. In that case the inequality (16) will not be satisfied. We then do not know if the generated orthogonal matrix still has the desired form.

A natural way to alleviate this problem is to incorporate sorting with Jacobi ordering. With the use of sorting, diagonal blocks settle down more quickly. After the diagonal blocks settle down, there will be no permutation of rows and columns between blocks and thus the inequality (16) is guaranteed. Experiments (e.g. those presented in [19, 26]) have shown that to incorporate sorting with Jacobi ordering can improve performance for symmetric matrices. In the next section we give some experimental results to show that ultimate quadratic convergence can be obtained for general normal matrices by incorporating a sorting procedure with the cyclic ordering when the QR algorithm with double implicit shift is used in the block Jacobi-like method for the local block submatrix reduction.

4 Experimental Results

In our experiments the basic algorithm for reducing B in Fig. 1 to a block triangular matrix at each step is the QR algorithm with double implicit shift and relevant subroutines in EISPACK [21] are used. (Note diagonal eigenvalues sometimes have to be swapped to ensure the norm of the lower off-diagonal block $\|A_{ji}\| = 0$.)

The stopping criteria used in our experiment is the same as that in EISPACK, that is, an off-diagonal element a_{ij} is considered as zero if $|a_{ij}| \leq (|a_{ii}| + |a_{jj}|) * \epsilon_{mach}$ for ϵ_{mach} the machine precision. The norm of a lower triangular off-diagonal block $\|A_{ji}\|$ is considered as zero if all the elements in it are considered as zero. The computation stops if all the lower triangular off-diagonal blocks are considered as zero.

Three different matrices are used in our experiments. The first matrix has distinct real eigenvalues, the second one has half of its eigenvalues real and the other half complex, and the third contains distinct complex eigenvalues.

Sweep	Lower Block Triangular Norm		
	Matrix 1	Matrix 2	Matrix 3
0	1.216919467218377	1.942474075382786	2.496954343408182
1	0.8439242118214305	1.444070030260214	1.849869669358216
2	0.6715976208966092	0.9099848524494509	1.085259770981389
3	0.5694434773575591	0.5781929865448607	0.4617255166054944
4	0.4741511123441828	0.3448206046877936	0.1168896666930765
5	0.4158622482882098	0.2100276271297201	2.394464637809306e-02
6	0.3396277313047820	0.1199726773598173	1.947540032820058e-04
7	0.2952900301463802	9.980169372632480e-02	9.808465071791773e-09
8	0.2670511113785617	9.139422480738207e-02	1.177560482223073e-15
9	0.1976341619796229	6.635400959295680e-02	3.205392919010865e-16
23	2.982459710120542e-02	2.170532650662510e-09	
24	1.472832625703867e-02	2.183222832088858e-09	
25	1.366728263813406e-02	9.386963954187879e-11	
26	1.125831279655028e-02	3.310682565962637e-16	
27	4.490752073166923e-03		
28	4.213903185722646e-03		
29	3.239742746922371e-03		
30	2.037809355066387e-03		

Table 1: Sweeps and lower block triangular norms for 40×40 matrices obtained by using the block algorithm without sorting.

In our first experiment we did not adopt any special sorting procedure. The procedures used to find an orthogonal matrix (the first step in the **IF** statement in Fig. 1) are as follows:

1.1. Reduce B to a block triangular form using standard QR with double implicit shift;
1.2. If $\|A_{ji}\|$ is nonzero, shift eigenvalues so that $\|A_{ji}\| = 0$.

The second procedure is used to ensure that the lower block triangular norm is zero.

Some experimental results are depicted in Table 1. It can be seen from the table that the algorithm performs well for the matrix which has only complex eigenvalues, but not so well for matrices which contain real eigenvalues. Note that the standard QR with double implicit shift does not sort real eigenvalues. Though it does not order complex eigenvalues either, the lower 2×2 block will converge to the true eigenvalues rapidly when it is close to a pair of conjugate complex eigenvalues because of the double shift. Therefore, the rows and columns of the block submatrix will not be exchanged during the orthogonal similarity transformation. The effect is similar to sorting although the eigenvalues are not sorted in the usual way. This may be the reason why a matrix with only complex eigenvalues can converge quadratically to block diagonal form.

In our second experiment the same QR algorithm was used for block submatrix reduction. However, we added a sorting procedure to sort real eigenvalues, that is,

1.1. Reduce B to a block triangular form using standard QR with double implicit shift;
1.2. If $\|A_{ji}\|$ is nonzero, shift eigenvalues so that $\|A_{ji}\| = 0$
1.3. Sort real eigenvalues in B in a nonincreasing order.

Sweep	Lower Block Triangular Norm		
	Matrix 1	Matrix 2	Matrix 3
0	1.216919467218377	1.942474075382786	2.496954343408182
1	0.6750246824159247	1.461835195736259	1.839162340923474
2	0.2199706803702323	0.9192624818874845	1.094029846622824
3	4.559208067945802e-02	0.4212417065916218	0.3932978785936760
4	2.000134475041065e-03	7.792134902151260e-02	7.739108958000541e-02
5	5.069396887509214e-06	4.672120103780485e-03	2.036895300210892e-03
6	1.708986452402760e-11	8.243231692930333e-06	8.707250423915960e-07
7	3.322903255620598e-16	1.606329703756845e-11	1.568197532065648e-13
8		1.764324938059070e-16	1.246328515799187e-16

Table 2: Sweeps and lower block triangular norms for 40×40 matrices obtained by the block algorithm with the use of sorting.

Matrix size	40	80	120	160	200
Matrix 1	7	8	9	9	10
Matrix 2	8	10	11	12	13
Matrix 3	8	10	11	12	13

Table 3: Sweeps taken for matrices of various sizes.

When the real eigenvalues in each block submatrix are sorted in a nonincreasing order, the cyclic ordering can guarantee that all the real eigenvalues will be sorted in nonincreasing order. Table 2 shows the experimental results when the same matrices are used as those in our first experiment. It can be seen that quadratic convergence is obtained for all the matrices after the sorting procedure is incorporated. This is a evidence for the importance of sorting in block Jacobi-like methods.

Some experimental results for matrices of different sizes are given in Table 3. We can see from this table that it takes a few more sweeps to converge for a matrix with complex eigenvalues than for a matrix with all real eigenvalues. We tried sorting the complex eigenvalues to improve the performance, but no significant improvement was achieved. It seems that the problem is harder to solve by the block Jacobi-like method when a matrix has complex eigenvalues, or when a normal matrix is unsymmetric.

5 Parallel Ring Jacobi Ordering

Our aim is to design a method for parallel implementation of eigenvalue decomposition of real normal matrices. The key to design a parallel Jacobi algorithm is to find an effective parallel ordering for simultaneous Jacobi rotations. The parallel ordering should be equivalent to the sequential cyclic Jacobi ordering to ensure quadratic convergence. In this section we first review a parallel Jacobi ring ordering algorithm [26]. When the order of generating different Jacobi rotations in each sweep is considered, this algorithm is equivalent to a well-known parallel round robin ordering [3] (and thus equivalent to the classical sequential cyclic-by-row ordering). However, one of the distinct features of this ring ordering algorithm is that parallel sorting of eigenvalues can be done by using the same procedure as parallel Jacobi ordering, and then ordering and sorting can be performed simultaneously in each sweep. Therefore, this ordering algorithm is equivalent to the cyclic Jacobi ordering in terms of both ordering and sorting, and is well suited for parallel implementation of our block method for eigenvalue decomposition of a real normal matrix. We briefly describe how our block method can be implemented in a parallel system using this parallel ring ordering.

Jacobi's method for eigenvalue decomposition works by applying a series of 2×2 Jacobi rotations to the left and right of the matrix to drive it to a desired diagonal form. As in the traditional Jacobi algorithm, the rotations are performed in a fixed sequence called a sweep. Each sweep consists of $n(n-1)/2$ rotations which are used to annihilates every off-diagonal elements exactly once. The iterative procedure terminates if one complete sweep occurs in which all off-diagonal elements are zeros to working accuracy. If the rotations in a sweep are chosen in a reasonable, systematic order, the convergence rate is ultimately quadratic [8, 11]. Exceptional cases in which cycling occurs are easily avoided by the use of a threshold strategy [25]. Since each Jacobi rotation only affects two rows and two columns of the matrix, different rotations can easily be applied in parallel to disjoint pairs of rows and columns of the matrix. This makes it very attractive for parallel processing.

Many parallel Jacobi ordering algorithms have been introduced in the literature [1, 2, 3, 6, 7, 9, 13, 14, 15, 16]. Most of these orderings can be classified into two equivalent groups by the definition of ordering equivalence, that is, two orderings are equivalent if they can generate the same set of index pairs at the same step by a relabelling of the initial indices. The odd-even ordering [15] and the round robin ordering [3] are good representatives for each of these two groups. The issue of equivalence of orderings was originally discussed in [16]. However, our experimental results show that two orderings satisfying this definition may not share the same convergence properties even for symmetric eigenvalue decomposition since an ordering which can also sort the diagonal elements in each sweep will converge faster than one which does not [4, 12, 20, 26]. We then introduced a new ordering algorithm called ring Jacobi ordering, as illustrated in Fig. 2.

Our ring Jacobi ordering consists of two procedures, forward sweep and backward sweep, as shown in Fig. 2 (a) and (b) respectively. These two procedures are applied alternately during the computation. In either forward or backward sweep the n indices are organised into two rows. Any two indices in the same column at a step form one index pair. One index in each column is then shifted to another column as shown by the arrows so that different index pairs are generated at the next step. The up-and-down arrow in the figure indicates the exchange of two indices in the column before one is shifted. This arrow plays a crucial rule in both ordering and sorting. Without it the indices initially placed in the same row can never meet each other and the elements will not be sorted. Both forward and backward sweeps will take $n-1$ steps to generate $n(n-1)/2$ different index pairs in a complete sweep. Fig. 2 also shows that the ring ordering is equivalent to the well known round robin ordering, that is, it can generate the same index pairs in each step after a permutation of the initial index positions.

One distinct feature of the ring ordering is that it can be used to sort the values of n elements into non-decreasing order (forward sweep) or non-increasing order (backward sweep) [26]. The sorting procedure using a forward sweep is described as follows: The n elements to be sorted are initially placed in two rows. During sorting the data flow pattern will be the same as that for ordering. A compare-exchange operation is applied to each index pair. Each step now consists of two sub-steps. The first sub-step compares the two elements in each column and places the smaller one on the top and the larger one at the bottom except in even steps the larger element is placed on the top if the column has a up-and-down arrow in it. The second sub-step simply shifts the elements located at the bottom to the next column according to the arrow ring. The n elements can be sorted in a non-decreasing order after $n-1$ such steps.

The parallel ring Jacobi ordering is equivalent to the cyclic Jacobi ordering in terms of both ordering and sorting and it has been implemented for symmetric eigenvalue decomposition and

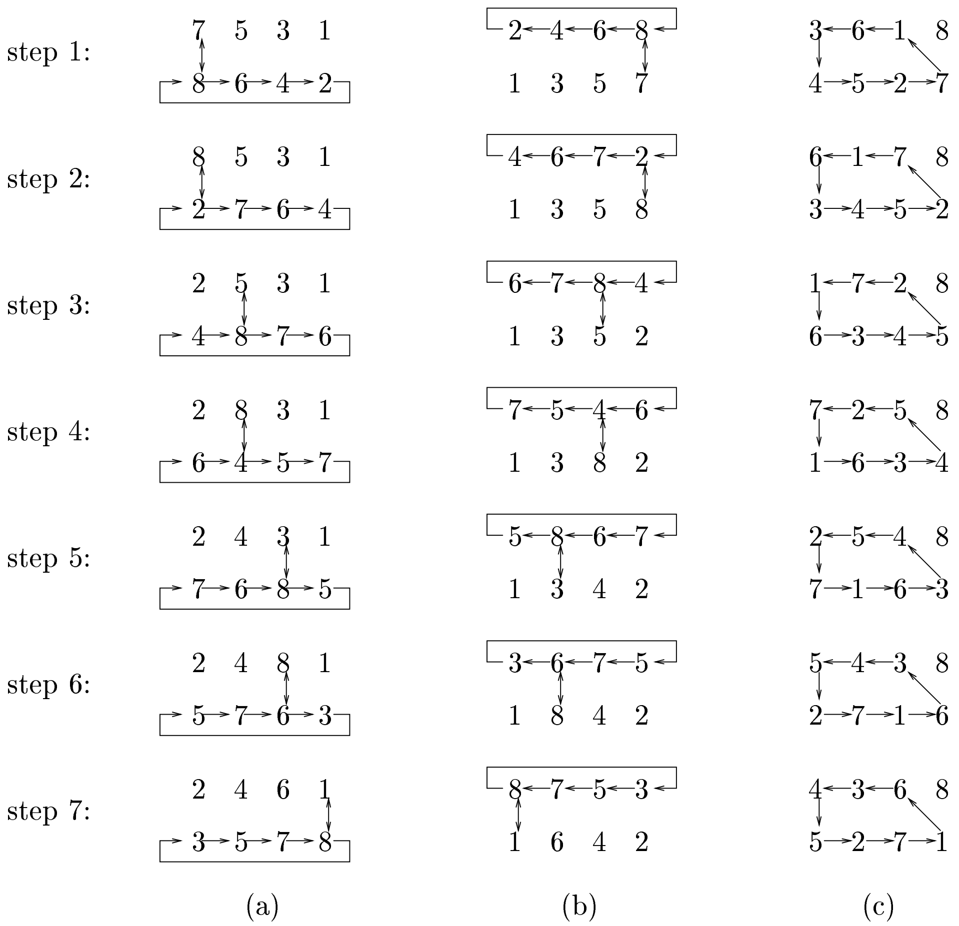

Figure 2: The ring Jacobi ordering. (a) forward sweep, (b) backward sweep, and (c) the round robin ordering with a new initialisation of indices.

SVD computations. Therefore, the parallel implementation of our block method using this ring Jacobi ordering is straightforward. In the following we briefly describe how this can be done in a parallel system in which the processors are configured as a one-dimensional ring (though the method can also be implemented on parallel machines of different configurations).

It should be noted that an algorithm without partitioning is not very useful in practice for general-purpose parallel computation because the system configuration is fixed, but the size of the user's problem may vary. We shall adopt a very efficient partitioning strategy for Jacobi's method which is described in [22]. For simplicity, we assume that the given system has p processors and $4p$ divides n for n the original matrix size. (If n is not divisible by $4p$, we may simply pad extra zero columns and rows.) We first divide the original matrix of size n into a block matrix of size m for $m = n/2$ and thus each block being of size 2×2. Next we further divide the m block columns into $2p$ groups with each group containing $m/2p$ block columns (or $n/4p$ columns). These groups are distributed evenly to p processors with each processor holding exactly two groups. Each group will have $m/2p$ main diagonal blocks of size 2×2. These groups are numbered and each is associated with one index as depicted in Fig. 2. They will move among the processors according to our parallel ring ordering in each sweep. At the beginning of a sweep, an initial step is required to annihilate the lower triangular off-diagonal blocks corresponding to the main diagonal blocks in the same group exactly once. This can be done by using the cyclic ordering. After that, in each step only those lower triangular off-diagonal blocks corresponding to two main diagonals from different groups are annihilated once, but the off-diagonal blocks associated with the main diagonals from the same group will not be annihilated again. Therefore, it is guaranteed that every lower triangular off-diagonal blocks is annihilated once and only once in one complete sweep.

When a lower triangular off-diagonal block is annihilated, the corresponding main diagonal elements are sorted at the same time. To sort the main diagonal elements of the entire matrix, we first sort the diagonal elements in each individual group using the cyclic ordering on each individual processor. (It is actually the bubble sort.) We then use our parallel ring ordering to sort diagonal elements in different groups. On each processor we can also adopt the cyclic ordering to sort diagonal elements in two groups (which can be considered as two super-indices in our ring ordering) in each step. The main diagonal elements of the entire matrix are then sorted after a complete sweep. Note that this sorting procedure is incorporated in the Jacobi ordering procedure and they are done simultaneously.

Before one group is moved to another processor at the end of each step, the rows and columns of the matrix need to be updated. Because the system is configured as a one-dimensional ring, the updating of the columns can be performed locally on each processor. However, the orthogonal transformation matrices need to be accumulated and sent to other processors for updating blocks of the corresponding rows.

The computation stops if all the off-diagonal blocks in the lower triangle are zero to working accuracy in a complete sweep. This can be done by adding a counter to each processor. At the end of each sweep the counter values are accumulated to see if the sum is equal to the number of the lower triangular off-diagonal blocks.

It can be seen that both computation and communication are quite balanced across the processors during the execution. This is one of the factors which make Jacobi's method attractive.

6 Conclusions

In this paper we discussed a block Jacobi-like method for the eigenvalue decomposition of a real normal matrix using real arithmetic. The basic idea of our method is first to divide the original matrix into blocks of size 2×2 and then treat each block as a single element and use the Jacobi method to reduce the block matrix to a block triangular form by applying a sequence of orthogonal similarity transformations (purely real arithmetic) to annihilate the off-diagonal blocks in the lower triangle. If the procedure converges, a real normal matrix will become block diagonal with each pair of conjugate complex eigenvalues grouped in the same block.

Our theoretical analysis and experimental results show that to achieve a quadratic convergence we have to incorporate a sorting procedure in each sweep for annihilating the off-diagonal blocks in the lower triangle. In our experiment the QR algorithm with double implicit shift is used for submatrix reduction. However, it is not the only candidate. For example, similar results were obtained when we used a scheme which combines the RTZ and the QR algorithms for local block submatrix reduction [28]. The key to success is to incorporate sorting with Jacobi ordering.

We also described a parallel ring Jacobi ordering. This parallel ordering can perform sorting and is thus equivalent to the well-known sequential cyclic Jacobi ordering in terms of both ordering and sorting. Therefore, it can be adopted in our method for parallel computation of eigenvalue decomposition of a real normal matrix. We showed an example of implementing the method on a parallel machine which is configured as a one-dimensional ring.

In this paper we have introduced a new method that can be implemented in parallel for eigenvalue decomposition of real normal matrices. However, there are still many technical issues relating to efficient implementation of the method on parallel machines, for example, the problems of blocking and cache effects when implementing the method on a particular parallel machine; the issue of pipeline processing [10]; the performance issue (i.e., if our method is comparable to the more general parallel QR-based method [24] for eigenvalue decomposition of real normal matrices on high-performance parallel computers); and the issue of the use of our method for real near-normal matrices. These are interesting topics for future research.

References

[1] C. H. Bischof, "The two-sided block Jacobi method on a hypercube", in *Hypercube Multiprocessors*, M. T. Heath, ed., SIAM, 1988, pp. 612-618.

[2] R. P. Brent, "Parallel algorithms for digital signal processing", *Proceedings of NATO Advanced Study Institute on Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, Leuven, Belgium, August, 1988.

[3] R. P. Brent and F. T. Luk, "The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays", *SIAM J. Sci. and Statist. Comput.*, 6, 1985, pp. 69-84.

[4] J. Demmel and K. Veselić, "Jacobi's method is more accurate than QR", *SIAM J. Sci. Stat. Comput.*, 11, 1992, pp. 1204-1246.

[5] P. J. Eberlein, "On the Schur decomposition of a matrix for parallel computation", *IEEE Trans. on Computers*, Vol. C-36, No. 2, 1987, pp. 167-174.

[6] P. J. Eberlein and H. Park, "Efficient implementation of Jacobi algorithms and Jacobi sets on distributed memory architectures", *J. Par. Distrib. Comput.*, 8, 1990, pp. 358-366.

[7] L. M. Ewerbring and F. T. Luk, "Computing the singular value decomposition on the Connection Machine", *IEEE Trans. Computers*, 39, 1990, pp. 152-155.

[8] G. Forsythe and P. Henrici, "The cyclic Jacobi method for computing the principal values of a complex matrix", *Trans. Amer. Math. Soc.*, 94, 1960, pp. 1-23.

[9] G. R. Gao and S. J. Thomas, "An optimal parallel Jacobi-like solution method for the singular value decomposition", in *Proc. Internat. Conf. Parallel Proc.*, 1988, pp. 47-53.

[10] D. Gimenez, V. Hernandez, R. van de Geijn and A. M. Vidal, "A Jacobi method by blocks on a mesh of processors", *Concurrency: Practice and Experience*, 9(5), 1997, pp. 391-411.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[12] K. Hüper and U. Helmke, *Structure and convergence of Jacobi-type methods*, Tech. Report, TUM-LNS-TR-95-02, Institute of Network Theory and Circuit Design, Technical University of Munich, Germany, 1995.

[13] T. J. Lee, F. T. Luk and D. L. Boley, *Computing the SVD on a fat-tree architecture*, Report 92-33, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, November 1992.

[14] F. T. Luk, "Computing the singular-value decomposition on the *ILLIAC IV*", *ACM Trans. Math. Softw.*, 6, 1980, pp. 524-539.

[15] F. T. Luk, "A triangular processor array for computing singular values", *Lin. Alg. Applics.*, 77, 1986, pp. 259-273.

[16] F. T. Luk and H. Park, "On parallel Jacobi orderings", *SIAM J. Sci. and Statist. Comput.*, 10, 1989, pp. 18-26.

[17] N. Mackey, "Hamilton and Jacobi meet again: quaternions and the eigenvalue problem", *SIAM J. Matrix Anal. Appl.*, Vol. 16, No. 2, 1995, pp. 421-435.

[18] M. Mantharam and P. J. Eberlein, "The real two-zero algorithm: a parallel algorithm to reduce a real matrix to a real Schur form", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 6, No. 1, 1995, pp. 48-62.

[19] W. F. Mascarenhas, "On the convergence of the Jacobi method for arbitrary orderings", *SIAM J. Matrix Anal. Appl.*, Vol. 16, No. 4, pp. 1197-1209.

[20] P. P. M. De Rijk, "A one-sided Jacobi Algorithm for computing the singular value decomposition on a vector computer", *SIAM J. Sci. and Stat. Comput.*, 10, 1989, pp. 359-371.

[21] 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*, Vol. 6 of Lecture Notes in Computer Science, Springer-verlag, Berlin, 2 ed., 1976.

[22] R. Schreiber, "Solving eigenvalue and singular value problems on an undersized systolic array", *SIAM. J. Sci. Stat. Comput.*, 7, 1986, pp. 441-451.

[23] G. W. Stewart, "A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix", *SIAM J. Sci. and Statist. Comput.*, Vol. 6, No. 4, 1985, pp. 853-864.

[24] R. Suda, A. Nishida and Y. Oyanagi, "A High Performance Parallelization Scheme for Hessenberg Double Shift QR Algorithm", *Parallel Computing*, Vol. 25, No. 6, 1999, pp. 729-744.

[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England, 1965.

[26] B. B. Zhou and R. P. Brent, "A parallel ordering algorithm for efficient one-sided Jacobi SVD computations", *Journal of Parallel and Distributed Computing*, Vol. 42, 1997, pp. 1-10.

[27] B. B. Zhou and R. P. Brent, "On the parallel implementation of the one-sided Jacobi algorithm for singular value decompositions", *Proc. of 3rd Euromicro Workshop on Parallel and Distributed Processing*, San Remo, Italy, January 1995, pp. 401-408.

[28] B. B. Zhou and R. P. Brent, "Jacobi-like algorithms for eigenvalue decomposition of a real normal matrix using real arithmetic", *Proceedings of the 10th International Parallel Processing Symposium*, Honolulu, Hawaii, April, 1996, pp. 593-600.