A PARALLEL ALGORITHM FOR COMPUTING THE SINGULAR VALUE DECOMPOSITION OF A MATRIX

E.R. JESSUP* AND D.C. SORENSEN†

Abstract.
A parallel algorithm for computing the singular value decomposition of a matrix is presented. The algorithm uses a divide and conquer procedure based on a rank one modification of a bidiagonal matrix. Numerical difficulties associated with forming the product of a matrix with its transpose are avoided, and numerically stable formulae for obtaining the left singular vectors after computing updated right singular vectors are derived. A deflation technique is described which together with a robust root finding method assures computation of the singular values to full accuracy in the residual and also assures orthogonality of the singular vectors.

Key words. bidiagonal matrix, singular value decomposition, divide and conquer algorithm.

AMS(MOS) subject classifications. 15A18, 65Y05.

1. Introduction

The singular value decomposition (SVD) of a real $m \times n$ matrix A can be written

$$A = U \Sigma V^T,$$

where U and V are both orthogonal matrices, and Σ is a diagonal matrix with non-negative diagonal elements. The columns of U and V are, respectively, the left and right singular vectors of A; the diagonal elements of Σ are its singular values. A standard algorithm for computing the singular value decomposition involves first reducing a matrix A to upper bidiagonal form B using elementary orthogonal transformations [12, 13] as follows

$$A = \hat{U} B \hat{V}^T$$

and then computing the SVD of $B = \hat{Y} \Sigma \hat{X}^T$. Combining the two results gives

$$A = \hat{U}(\hat{Y}\Sigma\hat{X}^T)\hat{V}^T = U \Sigma V^T ,$$

where $U = \hat{U}\hat{Y}$ and $V = \hat{V}\hat{X}$.

This paper focuses on the computation of the SVD of the bidiagonal matrix B by divide and conquer mechanisms based on rank one tearing of the bidiagonal matrix B. Algorithms founded on this technique have proven accurate and efficient for both serial and parallel computation of eigensystems of symmetric tridiagonal matrices [6, 10]. The notable speed and accuracy of the rank one updating process for that problem motivate application of rank one updating techniques to the singular value decomposition.

The presentation of the updating technique begins in Section 2 with a review of the rank one updating techniques used for the symmetric tridiagonal eigenproblem. Section 3 continues with a discussion of some difficulties arising in the design of an SVD algorithm that is both accurate and efficient. Section 4 describes a basic divide and conquer step for the SVD equivalent to a rank one tearing of a symmetric tridiagonal matrix.

Sections 5 and 6 are devoted to finite precision deflation rules and the orthogonality of the computed singular vectors. Section 7 covers implementation of the divide and conquer algorithm PSVD and the results of numerical experiments.

* Department of Computer Science, University of Colorado, Boulder, CO 80309-0430 (jessupcs.colorado.edu). This author was funded by DOE contracts W-31-109-Eng-38 (at Argonne National Laboratory), DE-AC05-84OR21400 (at Oak Ridge National Laboratory), and DE-FG02-92ER25122 and by NSF grant CCR-9109785.

† Department of Mathematical Sciences, Rice University, Houston, TX 77251-1892 (sorensen-rice.edu). This author was funded by DOE contracts W-31-109-Eng-38 (at Argonne National Laboratory) and DE-FG0f-91ER25103 and by NSF cooperative agreement CCR-9120008.

In all sections, we consider only the case $m = n$. If $m > n$, the initial reduction may be preceded by computing a QR factorization of A and using the $m \times n$ triangular matrix R in place of A . A similar procedure is appropriate for the case $m < n$.

Throughout this paper, unless otherwise specified, capital Roman letters represent matrices, lower case Roman letters represent column vectors, and lower case Greek letters represent scalars. A superscript T denotes transpose. All matrices and vectors are real.

2. Divide and Conquer for the Symmetric Tridiagonal Eigenproblem . In [6], Cuppen presents a divide and conquer technique for finding the eigenvalues and eigenvectors of a symmetric tridiagonal matrix. Rank one tearing is applied to divide the tridiagonal matrix T of order n into

$$T = \begin{pmatrix} \tilde{T}_1 & \beta e_k e_1^T \\ \beta e_1 e_k^T & \tilde{T}_2 \end{pmatrix} = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + \beta \begin{pmatrix} e_k \\ e_1 \end{pmatrix} \left(e_k^T, \quad e_1^T \right)$$

where $1 \le k \le n$, and e_j represents the j-th canonical vector of appropriate length. If the eigensystems of the two submatrices are $T_1 = Q_1 D_1 Q_1^T$ and $T_2 = Q_2 D_2 Q_2^T$, then

$$T \quad = \quad Q \left[D + \beta \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \left(z_1^T, \quad z_2^T \right) \right] Q^T = Q \left[D + \rho z z^T \right] Q^T,$$

where

$$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix}, \quad D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix},$$

z_1^T is the last row of Q_1, z_2^T is the first row of Q_2, and ρ is chosen so that $\| z \|_2 = 1$. The problem is then reduced to finding the eigensystem of a diagonal matrix plus a rank one change. The eigenvalues of T are equal to the eigenvalues of $D + \rho z z^T$; the eigenvectors of T are the eigenvectors of $D + \rho z z^T$ premultiplied by the matrix Q.

An updating technique described in [5, 6, 11] is employed to determine the eigensystem of $D + \rho z z^T$. When the diagonal elements of D are distinct and the elements of z are nonzero, the eigenvalues of $D + \rho z z^T$ are equal to the roots of

$$w(\lambda) = 1 + \rho z^T (D - \lambda I)^{-1} z^T$$

and can be determined efficiently by a rational interpolation scheme developed in [5]. The eigenvector corresponding to the i-th eigenvalue λ_i is found directly from

$$u_i = (D - \lambda_i I)^{-1} z.$$

When the diagonal elements of $D = \mathrm{diag}(\delta_1, \ldots, \delta_n)$ are not distinct (*i.e.*, $\delta_k = \delta_{k+1} = \ldots = \delta_{k+l}$), the eigenvector basis is rotated to zero out the components $\zeta_{k+1}, \ldots, \zeta_{k+l}$ of z corresponding to the repeated diagonal elements of D [5]. When the j-th element of z is zero, the element δ_j is an eigenvalue of $D + \rho z z^T$, and the j-th unit vector e_j is its corresponding eigenvector.

Multiple diagonal elements of D and zero elements of z result in significant reduction in the work required to compute the eigensystem of $D + \rho z z^T$. This phenomenon called *deflation* has been refined for use in finite precision arithmetic where *nearly equal* diagonal elements of D and *small* elements of z are deflated [10]. As shown in [6, 10, 15], substantial deflation and resulting savings in computation time occurs for a wide variety of symmetric tridiagonal eigenproblems. The computed eigensystem is obtained to high accuracy, and the computed eigenvectors are orthogonal [10, 14, 20].

An experimental comparison in [16] finds the implementation TREEQL [10] of the divide and conquer method and bisection with inverse iteration to be the fastest serial techniques for solving the symmetric tridiagonal eigenproblem. TREEQL is generally fastest when deflation is significant. The QL method (implemented as EISPACK's TQL2 [18]) is generally slowest. All three demonstrate comparable high accuracy in practice, although only TREEQL, TQL2, and bisection can be proven backward stable [3, 7]. TREEQL is also fast on shared-memory multiprocessors [10] but less efficient on statically-scheduled distributed-memory multiprocessors [14].

3. Background . The bidiagonal SVD is closely related to the symmetric tridiagonal eigenproblem. For example, the matrix products $\hat{T}_1 = B^T B$ and $\hat{T}_2 = B B^T$ are symmetric tridiagonal matrices of order n having as eigenvalues the squares of the singular values of B and having as eigenvectors, respectively, the left and right singular vectors of B. Thus, one way to determine the SVD of B is to compute the eigendecompositions $\hat{T}_1 = X \Sigma^2 X^T$ and $\hat{T}_2 = Y \Sigma^2 Y^T$. This approach, however, can be both inefficient and inaccurate. In this section, we review the drawbacks of using eigensolvers to compute $B = Y\Sigma X^T$.

First, finding the eigendecomposition of $\hat{T}_1$ gives only the singular values and the left singular vectors of B. Computing the eigendecomposition of $\hat{T}_2$ gives the left vectors Y but requires redundant computation of the singular value matrix Σ. Because X and Y are computed independently, it is also generally impossible to correctly pair the left and right singular vectors associated with equal or nearly equal singular values. It is preferable to compute each right singular vector using suitable relationships to its corresponding left singular vector.

The vector pairing problem can be overcome by computing the matrix of right singular vectors directly from $X = B^T Y \Sigma^{-1}$. This approach fails, however, when Σ has a zero diagonal element. Moreover, numerical experiments have shown an increased residual and degraded orthogonality of right singular vectors computed this way for matrices with large condition numbers. This is particularly disturbing as the SVD is often called upon when a matrix has a large condition number. Attempts to avoid conditioning problems through a combination of the two equations such as

$$(B + \sigma I)x = (B + \sigma I)^T y \tag{1}$$

can fail when there are a significant number of small singular values. However, Arbenz and Golub [2] suggest an iterative procedure using a modified Lanczos process which essentially corrects initial numerical errors made in equation (1).

Inaccuracies in the small singular values can also result from multiplication of B and its transpose in finite precision arithmetic [13]. For example, suppose that $\mathrm{fl}(1+\epsilon^2) = 1$ in finite precision arithmetic. If

$$B = \begin{pmatrix} 1 & 0 \\ 1 & \epsilon \end{pmatrix},$$

then the computed product $T^T = BB^T$ is

$$\mathrm{fl}\left[\begin{pmatrix} 1 & 0 \\ 1 & \epsilon \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & \epsilon \end{pmatrix}\right] = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

with exact eigenvalues 0 and 2. The computed singular values of B are then 0 and $\sqrt{2}$, while the true singular values are $\left(\frac{2\epsilon^2}{2+\epsilon^2+\sqrt{4+\epsilon^4}}\right)^{\frac{1}{2}}$ and $\left(\frac{2+\epsilon^2+\sqrt{4+\epsilon^4}}{2}\right)^{\frac{1}{2}}$. For this matrix, the relative error in the smallest computed singular value is one.

Some existing techniques for the solution of the singular value problem bypass these numerical problems by operating on the matrix B and implicitly forming the product BB^T. The Golub-Reinsch QL method [12, 13] for computing the SVD, for example, has been implemented as the LINPACK routine DSVDC. When using rank one updating techniques, however, it is not convenient to represent the torn matrix as the product of bidiagonal matrices in this way; it is necessary to devise a different way to work with the product matrix implicitly.

A final alternative that permits computation of a correct SVD is to embed the order n bidiagonal matrix in an order $2n$ symmetric banded matrix: the eigenvalues of the $2n \times 2n$ matrix

$$M_1 = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$$

are the singular values of B and their negatives. To compute the SVD of B, the columns and rows of M_1 are permuted to the order $1, n+1, 2, n+2, \ldots, n, 2n$ to form the $2n \times 2n$ tridiagonal matrix M_2 with a zero diagonal. The eigenvector u_i of M_2 corresponding to eigenvalue $\lambda_i = \sigma_i$ has as its odd-numbered components the components of the ith left singular vector $y_i = (\nu_{1i}, \ldots, \nu_{ni})^T$ and as its even-numbered components the components of the ith right singular vector $x_i = (\mu_{1i}, \ldots, \mu_{ni})^T$ [12]:

$$M_2 \begin{pmatrix} \nu_{1i} \\ \mu_{1i} \\ \vdots \\ \nu_{ni} \\ \mu_{ni} \end{pmatrix} = \sigma_i \begin{pmatrix} \nu_{1i} \\ \mu_{1i} \\ \vdots \\ \nu_{ni} \\ \mu_{ni} \end{pmatrix}.$$

Methods for the symmetric tridiagonal eigenproblem are then applied directly to the matrix M_2. This approach is efficient for methods that can take advantage of the zero structure of M_2 and that can compute the first n eigenpairs independently of the second n eigenpairs. Bisection with inverse iteration, for example, falls into this category the divide and conquer method of [6, 10] described in Section 2 does not [15]. A divide and conquer strategy that maintains the zero diagonal in the torn submatrices is described in [1].

The remainder of this paper discusses a method for computing the SVD that is both efficient and stable. It implicitly formulates the matrix product BB^T in a way that avoids cancellation in finite precision arithmetic, and it computes each right singular vector from its corresponding left singular vector. The formulation presented here is the most accurate of several alternatives considered in [17].

4. Divide and Conquer for the Bidiagonal SVD . This section presents a divide and conquer technique designed for use with the matrix B. It is an efficient alternative to the divide and conquer eigensolver applied to a $2n \times 2n$ tridiagonal matrix. It avoids the numerical difficulties associated with explicit formation of BB^T or B^TB by reformulating the product to prevent cancellation. The algorithm relies on rank one tearing. Specifically, the rank one modification of the matrix B

$$B = \begin{pmatrix} B_1 & \beta e_k e_1^T \\ 0 & B_2 \end{pmatrix} = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} + \beta \begin{pmatrix} e_k \\ 0 \end{pmatrix} (0, e_1^T), \tag{2}$$

where $\beta = \beta_k$, allows implicit formation of BB^T as follows:

$$\begin{aligned} BB^T &= \begin{pmatrix} B_1 & \beta e_k e_1^T \\ 0 & B_2 \end{pmatrix} \begin{pmatrix} B_1^T & 0 \\ \beta e_1 e_k^T & B_2^T \end{pmatrix} \\ &= \begin{pmatrix} B_1 B_1^T & 0 \\ 0 & B_2(I - e_1 e_1^T)B_2^T \end{pmatrix} + \begin{pmatrix} \beta e_k \\ B_2 e_1 \end{pmatrix} (\beta e_k^T, e_1^T B_2^T) \\ &= \begin{pmatrix} B_1 B_1^T & 0 \\ 0 & \hat{B}_2 \hat{B}_2^T \end{pmatrix} + \begin{pmatrix} \beta e_k \\ \alpha e_1 \end{pmatrix} (\beta e_k^T, \alpha e_1^T), \end{aligned} \tag{3}$$

where the matrix

$$\hat{B}_2 = B_2 \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$$

is the bidiagonal matrix B_2 with its first column replaced by the zero vector, and $\alpha e_1 = B_2 e_1$. This splitting may be considered a special case of the general rank one updates to the SVD described in [4].

The singular value decompositions $B_1 = U_1 \Sigma_1 V_1^T$ and $\hat{B}_2 = \hat{U}_2 \hat{\Sigma}_2 \hat{V}_2^T$ can be computed independently and used with equation (3) to produce

$$\begin{aligned} BB^T &= \begin{pmatrix} U_1 \Sigma_1^2 U_1^T & 0 \\ 0 & \hat{U}_2 \hat{\Sigma}_2^2 \hat{U}_2^T \end{pmatrix} + \begin{pmatrix} \beta e_k \\ \alpha e_1 \end{pmatrix} (\beta e_k^T, \alpha e_1^T), \\ &= \begin{pmatrix} U_1 & 0 \\ 0 & \hat{U}_2 \end{pmatrix} \left[\begin{pmatrix} \Sigma_1^2 & 0 \\ 0 & \hat{\Sigma}_2^2 \end{pmatrix} + \begin{pmatrix} u_1 \\ \hat{u}_2 \end{pmatrix} (u_1^T, \hat{u}_2^T) \right] \begin{pmatrix} U_1^T & 0 \\ 0 & \hat{U}_2^T \end{pmatrix}, \end{aligned} \tag{4}$$

where $u_1 = \beta U_1^T e_k$ and $\hat{u}_2 = \alpha \hat{U}_2^T e_1$. The eigendecomposition of the diagonal plus rank one matrix can be found via the updating techniques derived in [5, 10, 11] and summarized in Section 2.

This computation requires that the diagonal elements of the matrix

$$\begin{pmatrix} \Sigma_1^2 & 0 \\ 0 & \hat{\Sigma}_2^2 \end{pmatrix}$$

be distinct and that the elements of $(u_1^T, \hat{u}_2^T)$ be nonzero. When these assumptions do not hold, the problem deflates. However, because the squares of the singular values less than one are not as well-separated as the singular values themselves, the deflation rules of [10] concerning nearly equal diagonal values are not appropriate. To develop deflation rules for the SVD, it is necessary to reformulate the basic step of the updating process and to provide rules based on the original data rather than on the squared data appearing in equation (4).

To this end, let the $(n-k) \times (n-k-1)$ matrix $\tilde{B}_2$ be defined by

$$(0, \tilde{B}_2) \equiv \hat{B}_2 \equiv B_2 \left(I - e_1 e_1^T\right).$$

Now consider the singular value decomposition of $\tilde{B}_2 = \tilde{U}_2 \tilde{\Sigma}_2 \tilde{V}_2^T$, and note that

$$\hat{B}_2 = (\,\tilde{U}_2 \quad \tilde{u}\,) \begin{pmatrix} \tilde{\Sigma}_2 \\ 0^T \end{pmatrix} \tilde{V}_2^T,$$

where $\tilde{u}$ is a unit vector orthogonal to the columns of $\tilde{U}_2$. When $B_1 = U_1 \Sigma_1 V_1^T$,

$$B = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} \Sigma_1 & u_1 & 0 \\ 0 & u_2 & \tilde{\Sigma}_2 \\ 0 & \mu & 0 \end{pmatrix} \begin{pmatrix} V_1^T & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{V}_2^T \end{pmatrix}. \tag{5}$$

For notational convenience, we permute equation (5) to obtain

$$B = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 & u_1 \\ 0 & \tilde{\Sigma}_2 & u_2 \\ 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} V_1^T & 0 & 0 \\ 0 & 0 & \tilde{V}_2^T \\ 0 & 1 & 0 \end{pmatrix}.$$

Deflation rules are then needed for the interior matrix

$$\bar{M} \equiv \begin{pmatrix} \bar{\Sigma} & \bar{u} \\ 0 & \mu \end{pmatrix} \equiv \begin{pmatrix} \Sigma_1 & 0 & u_1 \\ 0 & \tilde{\Sigma}_2 & u_2 \\ 0 & 0 & \mu \end{pmatrix} \tag{6}$$

where $\bar{\Sigma} = \mathrm{diag}(\bar{\sigma}_1, \ldots, \bar{\sigma}_{n-1})$ and $\bar{u} = (\bar{\mu}_1, \ldots, \bar{\mu}_{n-1})$. (The matrices of equation (5) are not explicitly permuted in the implementation of PSVD described in Section 7.)

The deflation procedure for $\bar{M}$ resembles that for tridiagonal matrices. In exact arithmetic, the problem deflates whenever any of the following occurs:

1. an element of $\bar{u}$ is zero: $\bar{\mu}_j = 0$.
2. diagonal elements of $\bar{\Sigma}$ are equal: $\bar{\sigma}_i = \bar{\sigma}_j$, $i \neq j$.
3. a diagonal element of $\bar{\Sigma}$ is zero: $\bar{\sigma}_j = 0$

It is easily verified that if $\bar{\mu}_j = 0$, $\bar{\sigma}_j$ is a singular value of $\bar{M}$ with left and right singular vectors equal to the j-th canonical vector e_j providing the deflation for case 1. The other two cases may be reduced to case 1 using appropriate plane rotations to introduce a zero component in the vector $\bar{u}$.

When $\bar{\sigma}_i = \bar{\sigma}_j$, two-sided rotations are applied as in the tridiagonal case. A plane rotation G_1 in the (i, j)-plane is constructed and applied to $\bar{M}$ (and to the other matrix factors in equation (6) as well) so that

$$\bar{M} \leftarrow \begin{pmatrix} G_1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} G_1 \bar{\Sigma} G_1^T & G_1 \bar{u} \\ 0 & \mu \end{pmatrix} \begin{pmatrix} G_1^T & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G_1 \bar{\Sigma} G_1^T & G_1 \bar{u} \\ 0 & \mu \end{pmatrix}.$$

A 2×3 submatrix of $\bar{M}$ is affected as follows:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \sigma & 0 & \bar{\mu}_i \\ 0 & \sigma & \bar{\mu}_j \end{pmatrix} \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \sigma & 0 & 0 \\ 0 & \sigma & \tau \end{pmatrix},$$

where $\sigma = \bar{\sigma}_i = \bar{\sigma}_j$. with $\tau^2 = (\bar{\mu}_i)^2 + (\bar{\mu}_j)^2$, $c = \bar{\mu}_j/\tau$ and $s = \bar{\mu}_i/\tau$.

When $\bar{\sigma}_i = 0$, a one-sided rotation G_2 in the (i, n) - plane is applied from the left using μ to zero out $\bar{\mu}_i$: In this case

$$\bar{M} \leftarrow G_2 \bar{M}$$

and a 2×2 submatrix is affected as follows:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \bar{\sigma}_i & \bar{\mu}_i \\ 0 & \mu \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & \tau \end{pmatrix}$$

because $\bar{\sigma}_i = 0$.

In practice, these deflation rules must be modified to accommodate the limitations of finite precision arithmetic. Finite precision rules that apply when $\bar{\Sigma}$ or $\bar{u}$ has small elements or when $\bar{\Sigma}$ has close elements are given in Section 7.

Permuting so that all zero elements in the last column are grouped together, the result of deflation is a matrix of the form

$$\bar{M}' = PH\bar{M}G^T P^T = \begin{pmatrix} \hat{\Sigma}_1 & 0 & 0 \\ 0 & \hat{\Sigma} & u \\ 0 & 0 & \mu \end{pmatrix},$$

where $\tilde{\Sigma}$ has distinct, positive elements, and the vector u has only nonzero elements. P is the appropriate permutation matrix, and G and H are matrices consisting of accumulated products of the rotations constructed at each of the deflation steps.

After deflation, one need only compute the SVD of

$$M \equiv \begin{pmatrix} \tilde{\Sigma} & u \\ 0 & \mu \end{pmatrix} \equiv Y \Sigma X^T. \tag{7}$$

The diagonal elements of $\hat{\Sigma}_1$ are taken as singular values of $\bar{M}$ with appropriate canonical vectors as singular vectors. The squares of the singular values of M and its left singular vectors are given by the eigendecomposition

$$Y \Sigma^2 Y^T \equiv M M^T \equiv \begin{pmatrix} \tilde{\Sigma}^2 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} u \\ \mu \end{pmatrix} \begin{pmatrix} u^T, & \mu \end{pmatrix}.$$

An eigenvalue σ^2 of MM^T is a root of the secular equation

$$f(\sigma^2) = 1 + u^T (\tilde{\Sigma}^2 - \sigma^2)^{-1} u - \left(\frac{\mu}{\sigma}\right)^2 \tag{8}$$

and can be computed using the root-finder from [5]. If the sorted diagonal elements of $\mathrm{diag}(\tilde{\Sigma}^2, 0)$ are $0 = \tilde{\sigma}_1^2 < \tilde{\sigma}_2^2 < \ldots < \tilde{\sigma}_n^2$, the jth eigenvalue σ_j^2 of MM^T lies in the interval $(\tilde{\sigma}_j^2, \tilde{\sigma}_{j+1}^2)$ [5], and all eigenvalues are positive. The j-th singular value of B is σ_j, and the left singular vector of B associated with σ_j for $j = 1, \ldots, n$ is

$$y_j = \begin{pmatrix} (\tilde{\Sigma}^2 - \sigma_j^2)^{-1} u \\ -\mu / \sigma_j^2 \end{pmatrix} \theta,$$

where θ is a normalization factor. The corresponding right singular vector is

$$x_j = \frac{M^T y_j}{\| M^T y_j \|_2}.$$

That is, a vector in the direction of the right singular vector x_j is given by

$$\begin{aligned} M^T y_j &= \begin{pmatrix} \tilde{\Sigma}^2 & 0 \\ u^T & \mu \end{pmatrix} \begin{pmatrix} (\tilde{\Sigma}^2 - \sigma_j^2)^{-1} u \\ -\mu / \sigma_j^2 \end{pmatrix} \theta \\ &= \begin{pmatrix} \tilde{\Sigma}^2 (\tilde{\Sigma}^2 - \sigma_j^2)^{-1} u \\ \\ u^T (\tilde{\Sigma}^2 - \sigma_j^2)^{-1} u - \mu / \sigma_j^2 \end{pmatrix} \theta. \end{aligned}$$

Recall from equation (8) that σ_j satisfies

$$u^T (\tilde{\Sigma}^2 - \sigma_j^2)^{-1} u + \left(\frac{\mu}{\sigma_j}\right)^2 = -1.$$

Thus, the quantities σ_j, x_j, and y_j can be computed as follows

PROCEDURE 4.1 (SOLUTION OF THE DEFLATED UPDATING PROBLEM).

1. *Solve equation (8) for* σ_j.
2. $y = \begin{pmatrix} v_j \\ \eta_j \end{pmatrix} = \begin{pmatrix} (\tilde{\Sigma}^2 - \sigma_j^2)^{-1} u \\ -\mu / \sigma_j^2 \end{pmatrix}$.
3. $x = \begin{pmatrix} \tilde{\Sigma}^2 v_j \\ -1 \end{pmatrix}$.
4. $y_j = \frac{y}{\|y\|_2}$, $x_j = \frac{x}{\|x\|_2}$.

The orthogonality of the singular vectors computed according to this procedure is examined in Section 6.

The singular values of $B = \hat{Y} \Sigma \hat{X}^T$ are those of $\bar{M}$ and its singular vectors are derived from those of $\bar{M}$. Specifically,

$$\hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \Sigma \end{pmatrix},$$

$$\hat{Y} = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & Y \end{pmatrix},$$

and

$$\begin{pmatrix} V_1^T & 0 & 0 \\ 0 & 0 & \tilde{V}_2^T \\ 0 & 1 & 0 \end{pmatrix}^T \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix},$$

where I is the identity matrix of the same order as $\hat{\Sigma}_1$.

5. Deflation Rules for Finite Precision Arithmetic . Section 4 gives rules for deflating the problem when the matrix $\bar{M} = \begin{pmatrix} \bar{\Sigma} & \bar{u} \\ 0 & \mu \end{pmatrix}$ from equation (7) when $(\bar{\Sigma} \quad \bar{u})$ has equal diagonal elements, $\bar{\sigma}_i = \bar{\sigma}_j$, $i \neq j$, or zeros in the last column, $\bar{\mu}_j = 0$. As in the tridiagonal case, these rules can be extended to deflate the problem when $\bar{M}$ has *close* diagonal elements, $\bar{\sigma}_i \approx \bar{\sigma}_j$, or *small* elements, $|\bar{\mu}_j| < \epsilon$ for some small positive value of ϵ. In this section, we present deflation rules for finite precision arithmetic followed by an analysis to show that the errors imposed by deflation are small.

The finite precision rules are summarized in Procedure 5.1 below. They follow the same three basic steps as in exact arithmetic, but small elements of $\bar{u}$ or $\bar{\Sigma}$ are approximated as zero and close diagonal elements of $\bar{\Sigma}$ are approximated as equal. To keep track of deflation, two lists are used: *deflate_list* holds the indices of all small elements in the last column of the matrix, and *solve_list* holds the remaining indices. The list *solve_list* is initialized with the indices $1, \ldots, n-1$ of all diagonal elements in $\bar{\Sigma}$. When deflation is complete, the list *deflate_list* holds the indices of the diagonal elements of the transformed matrix to be accepted as singular values of $\bar{M}$, and *solve_list* holds the indices of the rows of $\bar{M}$ retained in the deflated matrix.

PROCEDURE 5.1 (DEFLATION IN FINITE PRECISION ARITHMETIC).

1. *For all $k \in$ solve_list,*
   if $|\bar{\mu}_k| < \epsilon$, move k from solve_list to deflate_list.
2. *Permute the indices so the elements of solve_list are increasing adjacent integers and $\bar{\sigma}_k \leq \bar{\sigma}_{k+1}$ (That is, replace M by $P'MP$ such that $Pe_n = e_n$ with $k' \in$ deflate_list for $1 \leq k' \leq |$deflate_list$|$ and $k' \in$ solve_list for $|$deflate_list$| + 1 \leq k' \leq n - 1$.)*
3. *For all but the last $k \in$ solve_list,*
   if $|\bar{\sigma}_k - \bar{\sigma}_{k+1}|$ is small, apply a two-sided plane rotation so that $\mu_k \leftarrow 0$, and move k from solve_list to deflate_list.
   { In PSVD, the two-sided rotation is applied as follows:
   $\tau^2 = (\bar{\mu}_k^2 + \bar{\mu}_{k+1}^2)$
   if $|(\bar{\sigma}_k - \bar{\sigma}_{k+1})\bar{\mu}_k\bar{\mu}_{k+1}| < \epsilon\tau^2$ *then*
   $c = \bar{\mu}_{k+1}/\tau$ *and* $s = \bar{\mu}_k/\tau$
   $\bar{\sigma}_k \leftarrow c^2\bar{\sigma}_k + s^2\bar{\sigma}_{k+1}$
   $\bar{\sigma}_{k+1} \leftarrow s^2\bar{\sigma}_k + c^2\bar{\sigma}_{k+1}$
   $\bar{\mu}_{k+1} \leftarrow \tau$
   $\bar{\mu}_k \leftarrow 0$ }
4. *For $k \in$ solve_list,*
   if $|\bar{\sigma}_k|$ is small, apply a one-sided plane rotation so that $\mu_k \leftarrow 0$, and move k from solve_list to deflate_list.
   { In PSVD, the one-sided rotation is applied as follows:
   $\tau^2 = (\bar{\mu}_k^2 + \mu^2)$
   if $|\bar{\sigma}_k\mu_k| < \epsilon\tau^2$ *then*
   $c = \bar{\mu}_k/\tau$ *and* $s = \mu/\tau$
   $\bar{\sigma}_k \leftarrow c\bar{\sigma}_k$
   $\mu \leftarrow \tau$
   $\bar{\mu}_k \leftarrow 0$ }

We now examine the errors introduced by Procedure 5.1. The first source of error is the transformation of the matrix at steps 2 and 3. The exact result of one of the two-sided rotations would be

$$
\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \bar\sigma_k & 0 & \bar\mu_k \\ 0 & \bar\sigma_{k+1} & \bar\mu_{k+1} \end{pmatrix} \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} =
$$

$$
\begin{pmatrix} \hat\sigma_k & 0 & 0 \\ 0 & \hat\sigma_{k+1} & \tau \end{pmatrix} + cs(\hat\sigma_k - \hat\sigma_{k+1}) \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \tag{9}
$$

where $\hat\sigma_k = c^2\bar\sigma_k^2 + s^2\sigma_{k+1}$ and $\hat\sigma_{k+1} = s^2\bar\sigma_k^2 + c^2\sigma_{k+1}$. In step 2, the second term in the matrix sum in equation (9) is set to zero. Thus, with each rotation an error is imposed of the form

$$
E_k^{(2)} = \epsilon_k \left(e_{k+1}{e_k}^T + e_k{e_{k+1}}^T \right),
$$

where $\epsilon_k = (\bar\sigma_k - \bar\sigma_{k+1})\bar\mu_k\bar\mu_{k+1}/(\bar\mu_k^2 + \bar\mu_{k+1}^2)$. The test to decide when rotations should be performed guarantees that $|\epsilon_k| < \epsilon$ for all k. Similarly, the exact result of one of the one-sided rotations would be

$$
\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \bar\sigma_k & \bar\mu_k \\ 0 & \mu \end{pmatrix} = \begin{pmatrix} \hat\sigma_k & 0 \\ 0 & \tau \end{pmatrix} - s\bar\sigma_k \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},
$$

so that each rotation in step 3 causes an error of the form

$$
E_k^{(3)} = \epsilon_k \left(e_n{e_k}^T \right),
$$

with $|\epsilon_k| < \epsilon$ for all k.

As in the exact case, the resulting matrix may then be permuted to form

$$
\begin{pmatrix} \hat\Sigma_1 & & \bar u_1 \\ 0 & \tilde\Sigma & u \\ 0 & 0 & \mu \end{pmatrix} + E_1 = \begin{pmatrix} \hat\Sigma_1 & & 0 \\ 0 & \tilde\Sigma & u \\ 0 & 0 & \mu \end{pmatrix} + E, \tag{10}
$$

where

$$
E = \begin{pmatrix} 0 & 0 & \bar u_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + E_1 .
$$

In these equations, E_1 is a matrix with elements bounded by ϵ, $\bar u_1 = \{\bar\mu_k | k \in \textit{deflate\_list}\}$, and $u = \{\bar\mu_k | k \in \textit{solve\_list}\}$. Thus, $|e_i^T \bar u_1| < \epsilon$ for all i and $|e_i^T u| \geq \epsilon$ for all i.

We approximate the vector $\bar u_1$ as zero, accept the diagonal elements of $\hat\Sigma$ as singular values of $\bar M$, and compute the SVD of the deflated submatrix

$$
\begin{pmatrix} \tilde\Sigma & u \\ 0 & \mu \end{pmatrix}
$$

by the procedure outlined in Section 4. Because the constituent errors are small, the SVD of $\bar M$ computed in this way is the exact SVD of a matrix close to $\bar M$.

The magnitude of the total error E depends on the deflation tolerance ϵ. A reasonable choice for ϵ is *macheps* $\times$ $\tilde\sigma_{max}$ where *macheps* is machine precision and $\tilde\sigma_{max}$ is the largest diagonal element of $\bar\Sigma$ in the undeflated matrix $\bar M$. Because the accurate determination of the small singular values may be important, however, it is also possible to vary the tolerance at each step of deflation according to the size of the singular value at that deflation step.

6. Orthogonality of the Singular Vectors . Let us now consider the possible limitations on orthogonality of singular vectors due to nearly equal singular values. Many of the results concerning this issue for the symmetric eigenvalue problem apply directly. In particular the recent results of Sorensen and Tang [20] concerning the numerical orthogonality of the computed eigenvectors will apply. Our first result in this section is a perturbation lemma that demonstrates the inherent difficulty with nearly equal roots.

For purposes of this discussion, we denote the diagonal elements of $\tilde\Sigma$ of the deflated matrix M by $\tilde\sigma_j$ and the corresponding components of the vector u by μ_j for $j = 1, 2, \ldots, n-1$ so that the secular equation (8) becomes

$$
f(\sigma^2) = 1 + \sum_{j=1}^{n-1} \frac{\mu_j^2}{\tilde\sigma_j^2 - \sigma^2} - \frac{\mu^2}{\sigma^2}. \tag{11}
$$

Let

$$(12) \qquad y_\sigma^T = \left(\frac{\mu_1}{\tilde\sigma_1^2-\sigma^2}, \frac{\mu_2}{\tilde\sigma_2^2-\sigma^2}, \dots \frac{\mu_n-1}{\tilde\sigma_{n-1}^2-\sigma^2}, \frac{\mu}{-\sigma^2}\right)\left[\frac{1}{f'(\sigma^2)}\right]^{\frac12},$$

and let

$$(13) \qquad x_\sigma^T = \left(\frac{\tilde\sigma_1\mu_1}{\tilde\sigma_1^2-\sigma^2}, \frac{\tilde\sigma_1\mu_2}{\tilde\sigma_2^2-\sigma^2}, \dots \frac{\tilde\sigma_1\mu_{n-1}}{\tilde\sigma_{n-1}^2-\sigma^2}, -1\right)\left[\frac{1}{1+\sum_{j=1}^{n-1}\frac{(\tilde\sigma_j\mu_j)^2}{(\tilde\sigma_j^2-\sigma^2)^2}}\right]^{\frac12}.$$

Note that the squares of the singular values of M are roots of f and that the unit vectors x_σ and y_σ are just permuted left and right singular vectors of M corresponding to σ when σ^2 is a root of f.

Lemma 6.1. *Let y_σ and x_σ be given by equations (12) and (13) respectively. Then for any $\sigma,\gamma \notin \{\tilde\sigma_i : i=1,\dots,n-1\}\cup\{0\}$,*

$$(14) \qquad |y_\sigma^T y_\gamma| = \frac{1}{|\sigma^2-\gamma^2|}\frac{|f(\sigma^2)-f(\gamma^2)|}{(f'(\sigma^2)f'(\gamma^2))^{\frac12}},$$

and

$$(15) \qquad |x_\sigma^T x_\gamma| = \left|\frac{\sigma^2(f(\sigma^2)-f(\gamma^2))}{(\sigma^2-\gamma^2)} + f(\gamma^2)\right|\left[\left(1+\sum_{j=1}^{n-1}\frac{(\tilde\sigma_j\mu_j)^2}{(\tilde\sigma_j^2-\sigma^2)^2}\right)\left(1+\sum_{j=1}^{n-1}\frac{(\tilde\sigma_j\mu_j)^2}{(\tilde\sigma_j^2-\gamma^2)^2}\right)\right]^{-\frac12}.$$

Proof: Equation (14) is just the result implied by Lemma 4.2 in [10]. To derive equation (15) we note that

$$\begin{aligned}
1 \;+\; & \sum_{j=1}^{n-1}\frac{\tilde\sigma_j^2\mu_j^2}{(\tilde\sigma_j^2-\sigma^2)(\tilde\sigma_j^2-\gamma^2)} \\
= \;& 1+\sum_{j=1}^{n-1}\frac{(\tilde\sigma_j^2-\sigma^2)\mu_j^2}{(\tilde\sigma_j^2-\sigma^2)(\tilde\sigma_j^2-\gamma^2)} - \frac{\mu^2}{\gamma^2} + \frac{\sigma^2\mu^2}{\sigma^2\gamma^2} + \sigma^2\sum_{j=1}^{n-1}\frac{\mu_j^2}{(\tilde\sigma_j^2-\sigma^2)(\tilde\sigma_j^2-\gamma^2)} \\
= \;& \left(\frac{1+\sum_{j=1}^{n-1}\mu_j^2}{\tilde\sigma_j^2-\gamma^2-\frac{\mu^2}{\gamma^2}}\right) + \sigma^2\left(\frac{\mu^2}{\sigma^2\gamma^2}+\sum_{j=1}^{n-1}\frac{\mu_j^2}{(\tilde\sigma_j^2-\sigma^2)(\tilde\sigma_j^2-\gamma^2)}\right) \\
(16) \qquad = \;& f(\gamma^2) + y_\sigma^T y_\gamma\left(f'(\sigma^2)f'(\gamma^2)\right)^{\frac12}.
\end{aligned}$$

Equation (15) follows from equation (14) applied to the second term in equation (16). ■

Note that in equations (12) and (13), y_σ and x_σ are always vectors of unit length and that the set of n vectors selected by setting σ^2 equal to the roots of equation (11) provides the set of left and right singular vectors for the deflated matrix M in equation (7). Moreover, equation (14) shows that the set of left vectors are mutually orthogonal, and equation (15) shows that the set of right vectors are mutually orthogonal whenever σ^2 and γ^2 are set to distinct roots of f. Finally, the term $|\sigma^2-\gamma^2|$ appearing in the denominator of (14) warns that it may be difficult to attain orthogonal singular vectors when the roots σ and γ are close.

We now wish to examine the situation of close roots. With an argument similar to that given in Lemma 4.6 of [10], one can show that $\tilde\sigma_i-\sigma$ must be bounded away from zero due to the deflation process. Because of this, we can expect to compute the differences $\tilde\sigma_j-\sigma$ to high relative accuracy. This is quite important with regards to orthogonality of the computed singular vectors as the following lemma shows.

Lemma 6.2. *Suppose that $\hat\sigma^2$ and $\hat\gamma^2$ are numerical approximations to exact roots σ^2 and γ^2 of f. Assume that these roots are distinct and let the relative errors for the quantities $\tilde\sigma_i-\sigma$ and $\tilde\sigma_i-\gamma$ be denoted by θ_i and η_i respectively. That is, the computed differences are*

$$(17) \qquad \tilde\sigma_i^2-\hat\sigma^2 = (\tilde\sigma_i^2-\sigma^2)(1+\theta_i) \quad \text{and} \quad \tilde\sigma_i^2-\hat\gamma^2 = ({\tilde\sigma_i}^2-\gamma^2)(1+\eta_i),$$

for $i=1,2,\dots,n$. Let $y_{\hat\sigma}$ and $y_{\hat\gamma}$ be defined according to equation (12), and let $x_{\hat\sigma}$ and $x_{\hat\gamma}$ be defined according to equation (13) using the computed quantities given in equation (17). If $|\theta_i|,|\eta_i|\le\epsilon << 1$, then

$$|y_{\hat\sigma}^T y_{\hat\gamma}| \le \epsilon(2+\epsilon)\left(\frac{1+\epsilon}{1-\epsilon}\right)^2,$$

and

$$|x_{\hat{\sigma}}^T x_{\hat{\gamma}}| \le \frac{\epsilon(2+\epsilon)}{(1-\epsilon)^2}.$$

Proof: A proof of the bound on the inner product of the left vectors is given in Lemma 4.7 of [10]. For the right vectors, note that

$$\begin{aligned} 1 \quad + & \sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(1+\theta_j)(\tilde{\sigma}_j^2 - \gamma^2)(1+\eta_j)} \\ = & -\sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} + \sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(1+\theta_j)(\tilde{\sigma}_j^2 - \gamma^2)(1+\eta_j)} \\ = & -\sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \left(\frac{\theta_j + \eta_j + \theta_j \eta_j}{(1+\theta_j)(1+\eta_j)} \right). \end{aligned}$$

Thus,

$$\begin{aligned} |x_{\hat{\sigma}}^T x_{\hat{\gamma}}| \quad = & \quad |\sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \left(\frac{\theta_j + \eta_j + \theta_j \eta_j}{(1+\theta_j)(1+\eta_j)} \right) \cdot \\ & \left[\left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \sigma^2)^2} \right) \left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \gamma^2)^2} \right) \right]^{-\frac{1}{2}} | \\ \le & \quad | \left(\sum_{j=1}^{n-1} | \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} | \right) \left(\frac{\epsilon(2+\epsilon)}{(1-\epsilon)^2} \right) \cdot \\ & \left[\left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \sigma^2)^2} \right) \left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \gamma^2)^2} \right) \right]^{-\frac{1}{2}} | \\ \le & \quad \left(\frac{\epsilon(2+\epsilon)}{(1-\epsilon)^2} \right), \end{aligned}$$

and the lemma is proved ■

This lemma shows that orthogonality can be assured whenever it is possible to provide small relative errors in the computed differences $\tilde{\sigma}_j - \sigma$. The results of [20] indicate that this condition may be achieved in practice.

We have applied the root-finder used in [10] directly to equation (11) and taken the square root of σ^2 to get a singular value without any apparent difficulty even when very difficult problems were solved. Nevertheless, it is certainly conceivable that problems could arise . Namely, it may be necessary to further refine the root finding process to prevent loss of accuracy in terms of the form

$$\frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2}$$

when $\tilde{\sigma}_j$ is small and σ is near to σ_j.

Both of the following two modifications to the root finder remove the dependence on σ^2 from all terms other than the term $\frac{\mu}{\sigma^2}$. First, one could use

$$\frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2} = \frac{\mu_j^2}{(\tilde{\sigma}_j + \sigma)(\tilde{\sigma}_j - \sigma)}$$

and update both $\tilde{\sigma}_j - \sigma$ and $\tilde{\sigma}_j + \sigma$ to avoid unnecessary cancellation caused by the squaring. Alternatively, one could use

$$\frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2} = \frac{\mu_j^2}{2\tilde{\sigma}_j} \left(\frac{1}{(\tilde{\sigma}_j + \sigma)} + \frac{1}{(\tilde{\sigma}_j - \sigma)} \right),$$

and the results of [20] would apply directly to the evaluation of f in this form.

If either of these last two schemes are employed then it is possible to show that the hypothesis (Eqn. 17) of Lemma 6.2 will be satisfied when the differences $\tilde{\sigma}_j - \sigma$ are computed to high relative accuracy.

7. Experimental Results . In this section, we present computational results from the implementation PSVD of the divide and conquer method developed in Sections 4–5. PSVD splits the matrix B recursively into submatrices of order 8 and solves the subproblems using DSVDC if B is of order 16 or greater and calls DSVDC without matrix splitting otherwise. DSVDC is the fastest serial method for solving the problems of very small order [15]. The results reported here concern only the computation of the SVD of a bidiagonal matrix B and not reduction of a general matrix to bidiagonal form. In all cases, the full sets of left and right singular vectors were computed along with the singular values.

The first set of experiments include serial timings and accuracy tests. These were carried out in double precision on a single Sequent Symmetry S81 processor using the Weitek floating point accelerator. On this machine, $macheps = 2.22 \times 10^{-16}$. We compare the results from PSVD to those from the LINPACK code DSVDC [8] and the implementation of bisection and inverse iteration B/III developed in [16].

The five bidiagonal matrices tested are introduced below. A pair of computed eigenvalues $\hat{\lambda}_i$, $\hat{\lambda}_{i+1}$ belong to a *cluster* if $\hat{\lambda}_i - \hat{\lambda}_{i+1} \leq 10^{-14}|\hat{\lambda}|_{max}$.

1. Matrix [2,1]: All diagonal elements are 2, and all off diagonal elements are 1. All singular values lie within the interval [1,3]. For all tested matrix orders, the singular values are computationally distinct.
2. Random: These matrices have uniformly distributed random entries between -1 and 1 generated by the uniform pseudorandom number generator RAND available from NETLIB on both diagonal and off-diagonal elements. The matrices tested turn out to have singular values with minimum magnitude $O(10^{-5})$.
3. B_W: Inspired by the Wilkinson matrix W^+ [21], this matrix of even order has diagonal elements $\frac{n}{2}, \ldots, 1, 1, \ldots, \frac{n}{2}$ and all off-diagonal elements equal to 1. Its smallest singular value is $O(10^{-3})$, and, in finite precision, its largest singular values have multiplicity two for matrix orders of about ten and larger.
4. Matrix $[2, u]/n$: The matrix $[2, u]/n$ of order n has the value $2/n$ in each off-diagonal position and the value i/n in the ith diagonal position. This matrix is ill-conditioned and has one singular value less than 10^{-14} for orders greater than eighty.
5. Modified matrix [2,1]: This matrix is formed from matrix [2,1] by setting the sixth through ninth diagonal elements $\alpha_6, \ldots, \alpha_9$ and fifth through eighth off-diagonal elements $\beta_5, \ldots, \beta_8$ equal to 10^{-14}. This matrix is severely ill-conditioned, having between four and eight singular values less than 10^{-8} and between two and four singular values less than 10^{-14} for all tested orders.

Let $\hat{X}\hat{\Sigma}\hat{Y}^T$ denote the computed SVD of B. To determine the accuracy of the result, we measure the residual error and the deviation from orthogonality of both sets of singular vectors:

$$\begin{aligned} \mathcal{R} &= \frac{1}{\hat{\sigma}_1} \max_i \| B\hat{x}_i - \hat{\sigma}_i \hat{y}_i \|_2 \\ \mathcal{O}_X &= \| \hat{X}^T \hat{X} - I \|_\infty \\ \mathcal{O}_Y &= \| \hat{Y}^T \hat{Y} - I \|_\infty \end{aligned}$$

When all of these quantities are small, the computed SVD of B is nearly the SVD of a matrix near B [15]. In other words, $(\hat{X} + \delta\hat{X})\hat{\Sigma}(\hat{Y} + \delta\hat{Y})^T$ is the exact SVD of a matrix $B + E$ with small $\delta\hat{X}$, $\delta\hat{Y}$, and E.

Table 7 shows the greatest residual and deviation from orthogonality measured for the five test problems solved by B/III, PSVD, and DSVDC for matrix orders 32, 100, and 200. Even for ill-conditioned matrices, PSVD attains full orthogonality of singular vectors and a small residual. Although only PSVD, DSVDC, and bisection (B) are provably stable methods, all three methods tested achieve similarly good results.

The runtimes for computing the full SVD's of the five matrices are compared in Figures 1–4 for problem [2,1] and B_W. (These are representative samples from the full set presented in [15].) For small order problems, PSVD is consistently the fastest method except when PSVD simply calls DSVDC, and B/III is the slowest. For larger order problems (greater than about 50), performance depends more strongly on the matrix characteristics. Namely, when significant deflation occurs, PSVD is the fastest of the three methods. Table 2 shows the fraction of singular values actually computed (as opposed to deflated) by PSVD for matrices [2,1] and B_W. The greater degree of deflation for B_W accounts for the lower runtime of PSVD relative to B/III for B_W.

Similar tests were performed on an Alliant FX/8. The operating system was Concentrix 3.0 and the optimization level for the subroutines comprising the units of computation were options -Ogv.

matrix order	method	maximum residual $\mathcal{R}$	maximum orthogonality (left) $\mathcal{O}_Y$	maximum orthogonality (right) $\mathcal{O}_X$
n = 32				
	PSVD	1.66d-14	7.65d-15	7.54d-15
	DSVDC	1.79d-15	1.13d-14	1.13d-14
	B/III	9.77d-16	9.76d-15	1.02d-14
n = 100				
	PSVD	9.39d-14	2.56d-14	2.37d-14
	DSVDC	4.22d-15	2.68d-14	2.86d-14
	B/III	2.38d-15	1.90d-14	1.87d-14
n = 200				
	PSVD	4.09d-15	1.13d-14	1.64d-14
	DSVDC	7.60d-15	8.13d-14	8.14d-14
	B/III	5.99d-15	5.42d-13	5.53d-14

TABLE 1

Maximum residual and orthogonalities of singular value decompositions computed by B/III, PSVD, and DSVDC for the five test matrices.

matrix	order	PSVD: fraction of roots computed
$[2, 1]$	32	1.0
	100	0.9
	200	0.8
B_W	32	0.6
	100	0.6
	200	0.5

TABLE 2

The number of roots computed by PSVD for five bidiagonal matrices.

order	Ratio of Time DSVDC (1 proc) / PSVD (8 procs)	Ratio of Time PSVD (1 proc) / PSVD (8 procs)
100	13.4	5.0
150	14.8	5.3
300	15.7	4.7
350	13.9	4.6

TABLE 3

Speedups for PSVD on the Alliant FX/8

Parallelism was invoked and controlled explicitly through the use of the SCHEDULE package [19]. Implementation details are quite similar to the those for the symmetric eigenvalue routine TREEQL [8]. The recursive matrix splitting leads to a hierarchy of subproblems with a data dependency graph in the form of a binary tree of height h. The smallest subproblems are of order $n/2^h$ and lie at the leaves of the tree (tree level 0). At level 0, the subproblems are solved independently in parallel, one problem per processor. At level l, $1 \leq l \leq h$, each problem of order $n/2^{h-l}$ is solved by updating the solutions to a pair of order $n/2^{h-l+1}$ subproblems from level $l-1$. At these levels, parallelism is achieved by dynamically assigning root-finding and singular vector computation tasks to processors [17]. As in the symmetric case, this parallel algorithm can be pipelined with block reduction of a matrix to bidiagonal form [9, 10, 17]. (An implementation of PSVD for distributed-memory machines without dynamically scheduled processes is described in [15]. Experiments with a similar implementation of the symmetric eigensolver suggest that PSVD would not be efficient on statically-scheduled multiprocessors [14, 15].)

Table 3 shows the speedup of PSVD run on eight processors relative to the PSVD and DSVDC run on one processor for matrix [2,1]. As can be seen, the performance of the parallel algorithm compared to DSVDC on a single processor is quite impressive. The somewhat disappointing results in the second column of this table are not yet understood. We attribute it to two aspects of the implementation. First the deflation step within each SVD update step is done serially and must be completed before any dynamic allocation is done for root finding. This limits the expected speedup. Second, there is potential for cache conflict when explicit parallel processing is done on the Alliant FX/8. We have not quantified either of these phenomena, however. Speedup of PSVD on eight processors as compared to one processor is limited to about 5. When the same comparison is done with 4 processors, speedup is limited to about 3. In all cases, the accuracies of DSVDC and PSVD were comparable.

The results in this section suggest that, as for the symmetric tridiagonal eigenproblem, PSVD provides a fast and accurate serial alternative to DSVDC and B/III. More care is needed in the parallel implementation to increase the speedup observed when PSVD is compared to itself using one processor and 8 processors. Moreover, a careful study of the effects of deflation is in order. A cursory examination of the results seemed to indicate that deflation is not nearly as prevalent in this setting as it has been in the symmetric tridiagonal case.

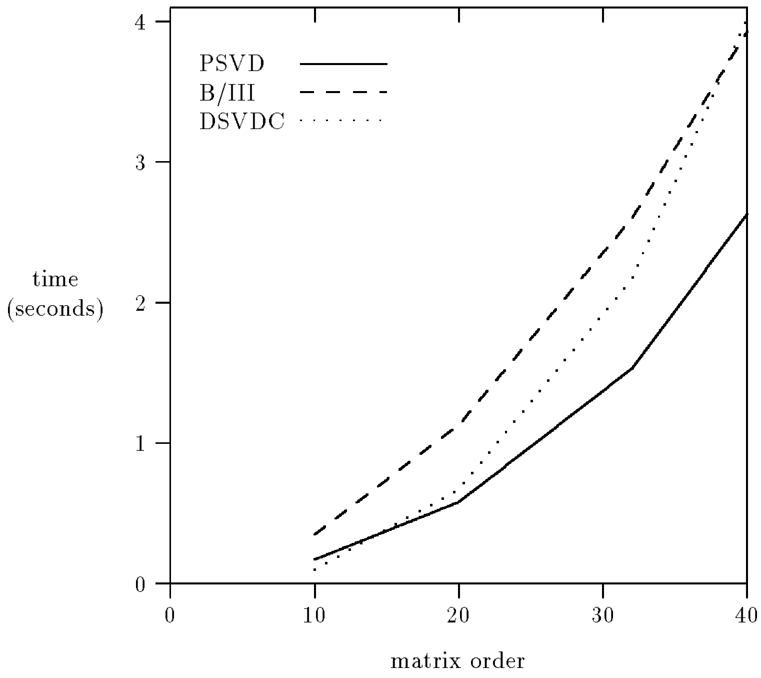

Fig. 1. *Times for computation of the SVD by B/III, PSVD, and DSVDC versus matrix order for matrix [2,1]*

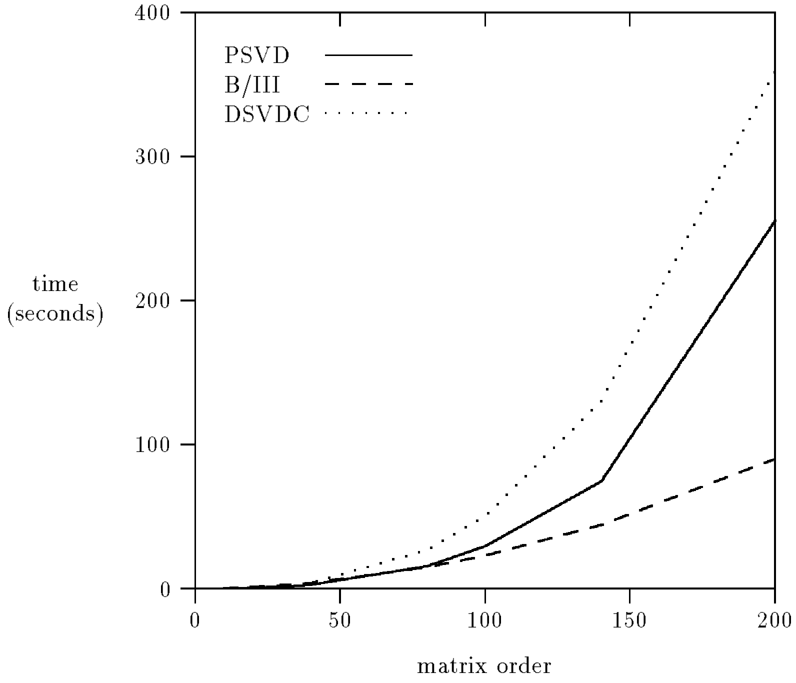

FIG. 2. *Times for computation of the SVD by B/III, PSVD, and DSVDC versus matrix order for matrix [2,1]*

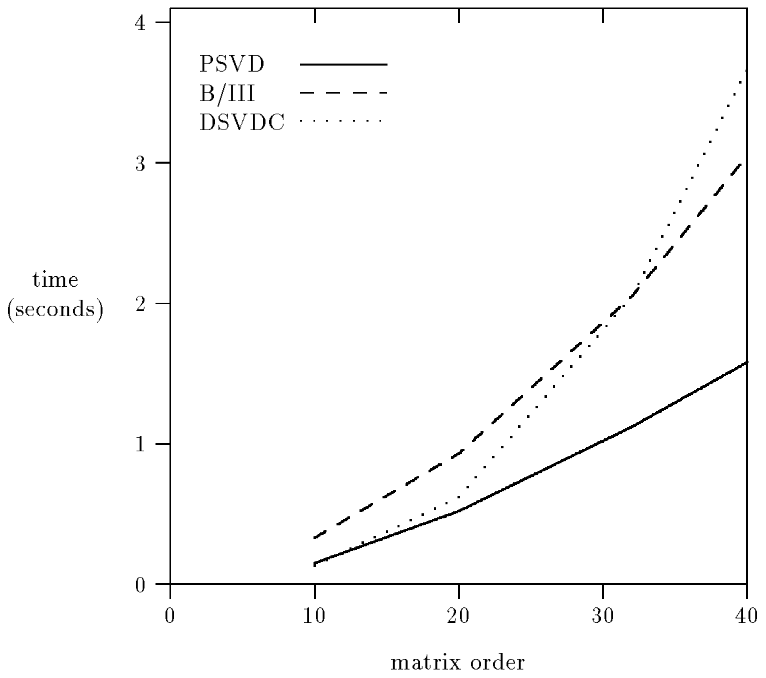

FIG. 3. *Times for computation of the SVD by B/III, PSVD, and DSVDC versus matrix order for matrix* B_W

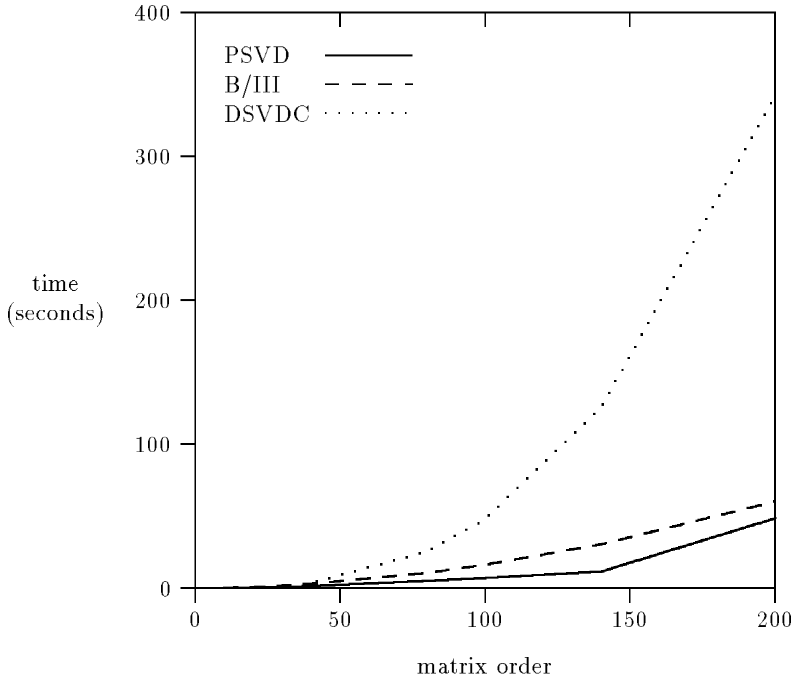

FIG. 4. *Times for computation of the SVD by B/III, PSVD, and DSVDC versus matrix order for matrix* B_W

8. Acknowledgements. We are grateful to Peter Arbenz and Gene Golub for several enlightening discussions. In particular, their paper [2] motivated us to consider the formulation developed in Section 3. We also wish to acknowledge a stimulating discussion during a special session on divide and conquer methods during the Gatlinburg X meeting held at Fairfield Glade in October 1987. A question from Gene Wachspress prompted us to reconsider the computation of right singular vectors so that we ultimately discovered the method described in Section 6.

REFERENCES

[1] P. Arbenz, *Divide-and-conquer algorithms for the computation of the SVD of bidiagonal matrices*, in Vector and Parallel Computing, J. Dongarra, I. Duff, P.Gaffney, and S. McKee, eds., Ellis Horwood, 1989, pp. 1–10.

[2] P. Arbenz and G. Golub, *On the spectral decomposition of Hermitian matrices subjected to indefinite low rank perturbations*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[3] J. Barlow, *Error analysis of update methods for the symmetric eigenvalue problem.* To appear in SIAM J. Matrix Anal. Appl.

[4] J. Bunch and C. Nielsen, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.

[5] J. Bunch, C. Nielsen, and D. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[6] J. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–95.

[7] J. Demmel and W. Kahan, *LAPACK working note #3: Computing small singular values of bidiagonal matrices with guaranteed relative accuracy*, Mathematics and Computer Science Division, Argonne National Laboratory, 1988.

[8] J. Dongarra, J. Bunch, C. Moler, and G. Stewart, *LINPACK Users' Guide*, SIAM Publications, 1979.

[9] J. Dongarra, S. Hammarling, and D. Sorensen, *Block reduction of matrices to condensed form for eigenvalue computations*, Journal of Computational and Applied Mathematics, 27 (1989), pp. 215–227.

[10] J. Dongarra and D. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s139–s154.

[11] G. Golub, *Some modified matrix eigenvalue problems*, SIAM Review, 15 (1973), pp. 318–34.

[12] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. SIAM Numer. Anal., Ser. B, Vol. 2 (1965), pp. 205–224.

[13] G. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, in Handbook for Automatic Computation: Linear Algebra, Springer Verlag, 1971, pp. 134–151.

[14] I. Ipsen and E. Jessup, *Solving the symmetric tridiagonal eigenvalue problem on the hypercube*, SIAM J. Sci. Stat. Comput., Vol. 11, No. 2, (1990), pp. 203–229.

[15] E. Jessup, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem*, PhD thesis, Dept of Computer Science, Yale University, 1989.

[16] E. Jessup and I. Ipsen, *Improving the accuracy of inverse iteration*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 550–571.

[17] E. Jessup and D. Sorensen, *A parallel algorithm for computing the singular value decomposition of a matrix*, Technical Report ANL/MCS-TM-102, Argonne National Laboratory, 1987.

[18] B. Smith, J. Boyle, J. Dongarra, B. Garbow, Y. Ikebe, V. Klema, and C. Moler, *Matrix Eigensystem Routines–EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, 2nd edition*, Springer-Verlag, 1976.

[19] D. Sorensen and J. Dongarra, *SCHEDULE: Tools for developing and analyzing parallel Fortran programs*, in The Characteristics of Parallel Algorithms, D. G. L.H. Jamieson and R. Douglass, eds., MIT Press, 1987.

[20] D. Sorensen and P. Tang, *On the orthogonality of eigenvectors computed by divide and conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[21] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.