THE CSD, GSVD, THEIR APPLICATIONS AND COMPUTATIONS

By

Zhaojun Bai

IMA Preprint Series # 958

April 1992

The CSD, GSVD, their Applications and Computations

Zhaojun Bai*

April 6, 1992

Abstract

Since the CS decomposition (CSD) and the generalized singular value decomposition (GSVD) emerged as the generalization of the singular value decomposition about fifteen years ago, they have been proved to be very useful tools in numerical linear algebra. In this paper, we review the theoretical and numerical development of the decompositions, discuss some of their applications and present some new results and observations. We also point out some open problems. A Fortran 77 code has been written that computes the CSD and the GSVD.

Keywords: singular value decomposition, CS decomposition, generalized singular value decomposition.

Subject Classifications: AMS(MOS): 65F30; CR:G1.3

1 Introduction

The singular value decomposition (SVD) of a matrix is one of the most important tools in numerical linear algebra. It has been widely used in scientific computing. Recently, Stewart [52] gave an excellent survey on the early history of the SVD back to the contributions of E. Beltrami and C. Jordan in 1873 and 1874.

In contrast to the SVD, the CS decomposition (CSD) of a partitioned orthonormal matrix and the generalized singular value decomposition (GSVD) of two matrices having the same number of columns have been around only about fifteen years. In the early years, the two decompositions were developed separately. Stewart in 1977 [48] first put forward an explicit CSD form, which he called a general decomposition of a unitary matrix, although it is implicit in the works of Davis and Kahan in 1970 [16] regarding the perturbation of a linear operator, and Björck and Golub in 1973 [11] concerning the canonical angles between subspaces. The motivation of the CSD stems from the question of how much we can simplify the representation of an orthogonal projector and how to measure separation between subspaces. The CSD allows us to define canonical angles between pairs of subspaces in such a way that as the largest canonical angle approaches zero the subspaces approach one another. This in turn leads to some useful results on the singular values of products and differences of projections.

The GSVD was first introduced by Van Loan in his PhD dissertation in 1976 [56], which then was called the B-singular value decomposition to refer the standard SVD of one

*Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA, e-mail: bai@ms.uky.edu

matrix A associated with another given matrix B having the same number of columns of A. Specifically, it is well known that the singular values $\sigma(A)$ of a matrix A are the square roots of the eigenvalues of positive semi-definite symmetric matrix A^TA, i.e.,

$$\sigma(A) = \{\sigma \geq 0 \,|\, \det(A^TA - \sigma^2 I) = 0\},$$

where $\det(A)$ stands for the determinant of the matrix A. It is a natural generalization that if given another matrix B, the generalized singular values $\sigma(A, B)$ are defined as the square roots of the eigenvalues of positive semi-definite symmetric matrix pencil $A^TA - \lambda B^TB$, i.e.,

$$\sigma(A, B) = \{\sigma \geq 0 \,|\, \det(A^TA - \sigma^2 B^TB) = 0\}.$$

Van Loan shows that $\sigma(A, B)$ can obtained by factorizing A and B into the products of an orthogonal matrix, a diagonal matrix and a nonsingular matrix, respectively. In particular, if B is the identity matrix, the decomposition gives the SVD of A.

In 1981, Paige and Saunders [40] described a more general formulation of Van Loan's B-SVD, which they called the *generalized singular value decomposition* (GSVD). Under their formulation, the CSD can be regarded as a special case of the GSVD.

The assessment of the conditioning of the decompositions comes from the perturbation analyses by Sun [53], Paige [41], Demmel and Verselić [19] and Li [36]. From their analyses, we know that the CSD is perfectly conditioned. However, the GSVD is not necessarily well conditioned under perturbation, unlike the standard SVD.

The CSD and the GSVD have been found to be very useful tools in numerical linear algebra. Their applications in many generalized problems are in the same spirit as the SVD in corresponding standard problems, such as in finding the intersection of the null spaces of two matrices [28], in the generalized eigenvalue problem arising from signal processing [47], in computing the Kronecker form of matrix pencil $A - \lambda B$ [32], in the constrained least squares problems [28], in the least squares problem with Tikhonov regularization [29], and so on.

The development of numerical methods for computing the CSD and the GSVD started with the paper by Stewart [49] based on the SVD and the Jacobi method for the symmetric eigenvalue problem. Later, Van Loan [57] put forward another method based on the SVD and the QR decomposition. Both Stewart's and Van Loan's algorithms primarily aim at computing the CSD, but they can be used to compute the GSVD by incorporating other standard decompositions. With the development of the Kogbetliantz algorithm for computing the SVD of a product of two matrices by Heath, Laub, Paige and Ward [31] and Hari and Veselić [30], Paige proposed a generalization of the Kogbetliantz algorithm to compute the GSVD directly. The most recent work by Bai and Demmel [7], and Adams, Bojanczyk, Ewerbring, Luk and Van Dooren [13,2], focuses on the stabilization and accuracy improvement of the existing three algorithms.

In this paper, we shall review theoretical aspects of the CSD and the GSVD in a unified way, and present some new results and insights for the decompositions. It will be emphasized that although the different formulations of the decompositions are mathematically equivalent, they have different numerical properties. A number of applications of the CSD and the GSVD will be discussed in this paper. We shall also review the three existing numerical algorithms and their variations for computing the decompositions, and compare their performance. A modified version of Stewart's algorithm is presented. Some open problems are discussed at the end of this paper.

A portable Fortran 77 code SGGSVD has been developed for computing the GSVD by Bai, Demmel and Zha. It can be obtained from the author.

Recently, motivated by different origins of application problems, the SVD has been further generalized for matrix triplets by Ewerbring and Luk [24] and Zha [63]. A proposal has been put forward by De Moor, Golub and Zha [20,21] for standardizing nomenclature of the generalization of the SVD of any number of matrices with conforming dimensions. Because of the length of this paper, we shall not discuss these generalizations in detail.

Throughout this paper we shall use the notational conventions of Golub and Van Loan's book *Matrix Computations* (second edition) [28]. Specifically, matrices are denoted by upper case italic and Greek letters, vectors by lower case italic letters, and scalars by lower case Greek letters or lower case italic if there is no confusion. The (i,j) entry of a matrix A is denoted by a_{ij}. The symbol $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ the set of real n-vectors, and $\mathbb{R}^{m\times n}$ the set of real $m \times n$ matrices. The matrix A^T is the transpose of A. The matrix $|A| = (|a_{ij}|)$. rank(A) is the rank of matrix A; null(A) is the null space of A, namely, null$(A) = \{x \,|\, Ax = 0\}$; span$\{Q\}$ is the subspace spanned by the columns of the matrix Q. $\|\cdot\|_p$ is the p-norm of a vector or matrix; usually, $p = 1, 2, \infty, F$. $\kappa_p(A)$ denotes the condition number of the matrix A: $\kappa_p(A) = \|A\|_p\|A^\dagger\|_p$, where $A^\dagger$ is the pseudo-inverse of A. Sometimes, $\kappa(A)$ is used if it is not necessary to specify the norm.

The presentation in this paper is based on the set of real numbers. All results have their analogues for the set of complex numbers.

The rest of this paper is organized as follows: §2 is on the theory of the decompositions and definitions of generalized singular values. §3 presents the perturbation theory of the CSD and the GSVD, and assesses of the conditioning of the decompositions. §4 discusses applications of the CSD and the GSVD. §5 is concerned with the numerical computations of the decompositions. §6 lists related generalizations of the SVD. The paper ends with a summary and some open problems.

2 The CSD and the GSVD

In this section, we first present the CSD, then the GSVD and their variants. It should be emphasised that the GSVD presented here is based on the CSD. However, historically, the existence of two decompositions was proved independently. Some details of existence proofs of the decompositions are presented since they reveal the possible different formulations of the decompositions for different dimensions, and shed light on their numerical computation. Finally, we shall define the generalized singular values (pairs), and their connection with the standard SVD.

Theorem 2.1 (CS Decomposition [16,11,48]) *Let $Q \in \mathbb{R}^{(m+p)\times k}$ have orthonormal columns. Partition Q in the form*

$$Q = \begin{array}{c c} & k \\ m & \left(\begin{array}{c} Q_1 \\ Q_2 \end{array}\right. \\ p & \end{array}\!\!\left.\begin{array}{c} \\ \\ \end{array}\right). \tag{2.1}$$

Then there are orthogonal matrices $U_1 \in \mathbb{R}^{m\times m}$, $U_2 \in \mathbb{R}^{p\times p}$ and $V \in \mathbb{R}^{k\times k}$ such that

$$\begin{pmatrix} U_1^{\mathrm{T}} & 0 \\ 0 & U_2^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} V = \begin{pmatrix} U_1^{\mathrm{T}} Q_1 V \\ U_2^{\mathrm{T}} Q_2 V \end{pmatrix} = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix}$$

assumes one of the following forms:
If $m \geq k, \ \ p \geq k$,

$$\begin{array}{cc} & k \\ \begin{array}{c} m \\ p \end{array} & \left(\begin{array}{c} \Sigma_1 \\ \Sigma_2 \end{array}\right) \end{array} = \begin{array}{cc} & k \\ \begin{array}{c} k \\ m-k \\ k \\ p-k \end{array} & \left(\begin{array}{c} C \\ 0 \\ S \\ 0 \end{array}\right) \end{array};$$

if $m \geq k, \ \ p < k$,

$$\begin{array}{cc} & k \\ \begin{array}{c} m \\ p \end{array} & \left(\begin{array}{c} \Sigma_1 \\ \Sigma_2 \end{array}\right) \end{array} = \begin{array}{cc} & \begin{array}{cc} p & k-p \end{array} \\ \begin{array}{c} p \\ k-p \\ m-k \\ p \end{array} & \left(\begin{array}{cc} C & 0 \\ 0 & I \\ 0 & 0 \\ S & 0 \end{array}\right) \end{array};$$

if $m > k, \ \ p \geq k$,

$$\begin{array}{cc} & k \\ \begin{array}{c} m \\ p \end{array} & \left(\begin{array}{c} \Sigma_1 \\ \Sigma_2 \end{array}\right) \end{array} = \begin{array}{cc} & \begin{array}{cc} m & k-m \end{array} \\ \begin{array}{c} m \\ m \\ k-m \\ p-k \end{array} & \left(\begin{array}{cc} C & 0 \\ S & 0 \\ 0 & I \\ 0 & 0 \end{array}\right) \end{array};$$

and if $m < k, \ \ p < k$,

$$\begin{array}{cc} & k \\ \begin{array}{c} m \\ p \end{array} & \left(\begin{array}{c} \Sigma_1 \\ \Sigma_2 \end{array}\right) \end{array} = \begin{array}{cc} & \begin{array}{ccc} n-k & k-p & k-m \end{array} \\ \begin{array}{c} n-k \\ k-p \\ n-k \\ k-m \end{array} & \left(\begin{array}{ccc} C & 0 & 0 \\ 0 & I & 0 \\ S & 0 & 0 \\ 0 & 0 & I \end{array}\right) \end{array}, \qquad m+p=n.$$

Here C *and* S *are nonnegative diagonal matrices satisfying*

$$C^2 + S^2 = I. \tag{2.2}$$

Proof. We carry out the proof in detail because it sheds light on the numerical computation of the decomposition. The proof is constructive and is essentially due to Stewart [49]. We first assume that matrices Q_1 and Q_2 are square, then we show that how the rest of the cases can be reduced to that of computing a CSD when Q_1 and Q_2 are square.

First, by the SVD of Q_1:

$$U_1^{\mathrm{T}} Q_1 V = C,$$

we know that the matrix

$$\left(\begin{array}{c} C \\ \bar{Q}_2 \end{array}\right) = \left(\begin{array}{cc} U_1^{\mathrm{T}} & 0 \\ 0 & I \end{array}\right) \left(\begin{array}{c} Q_1 \\ Q_2 \end{array}\right) V$$

has orthonormal columns. Hence

$$C^2 + \bar{Q}_2^{\mathrm{T}} \bar{Q}_2 = I,$$

which implies that

$$\bar{Q}_2^{\mathrm{T}} \bar{Q}_2 = I - C^2$$

is diagonal. This means that the columns of $\bar{Q}_2$ are orthogonal. Let $\bar{q}_j^{(2)}$ denote the j-th column of $\bar{Q}_2$, Then the desired matrix U_2 can be constructed as follows:

1. For each j such that $\bar{q}_j^{(2)} \neq 0$, set $u_j^{(2)} = \bar{q}_j^{(2)}/\|\bar{q}_j^{(2)}\|$.
2. Fill in the remaining columns of U_2 with an orthonormal basis for the orthogonal complement of the column space of $\bar{Q}_2$.

From the orthogonality of the columns of $\bar{Q}_2$, we have

1. U_2 is orthogonal,
2. $U_2^{\mathrm{T}}\bar{Q}_2 = S$, where $S = \mathrm{diag}(s_1, s_2, \dots, s_k)$ with $s_j = \begin{cases} \|\bar{q}_j^{(2)}\|, & \text{if } \bar{q}_j^{(2)} \neq 0, \\ 0, & \text{if } \bar{q}_j^{(2)} = 0. \end{cases}$

This completes the proof of the theorem for $m = p = k$.

There are four cases, corresponding to the four forms in the theorem. We only need to treat the first and second cases; these illustrate the techniques sufficiently well for the other two cases.

For the case $m \geq k$, $p \geq k$, let

$$\bar{U}_1^{\mathrm{T}} Q_1 = \begin{pmatrix} \bar{Q}_1 \\ 0 \end{pmatrix}, \quad \bar{U}_2^{\mathrm{T}} Q_2 = \begin{pmatrix} \bar{Q}_2 \\ 0 \end{pmatrix},$$

be the QR factorizations of Q_1 and Q_2. Then the problem is reduced to that of computing the CSD of the $k \times k$ square matrices $\bar{Q}_1$ and $\bar{Q}_2$.

Turing to the case $m \geq k$, $p \leq k$, let

$$Q_2 V_1 = \begin{array}{cc} & \begin{array}{cc} p & k-p \end{array} \\ p & \begin{pmatrix} \bar{Q}_{21} & 0 \end{pmatrix} \end{array}$$

be the RQ factorization of Q_2. Then

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} V_1 = \begin{array}{cc} & \begin{array}{cc} p & k-p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{pmatrix} \hat{Q}_{11} & \hat{Q}_{12} \\ \bar{Q}_{21} & 0 \end{pmatrix} \end{array}.$$

By the orthonormality $\hat{Q}_{11}^{\mathrm{T}}\hat{Q}_{12} = 0$, $\hat{Q}_{12}^{\mathrm{T}}\hat{Q}_{12} = I$. Hence if

$$\hat{U}_1 = \begin{pmatrix} \hat{U}_{11} & \hat{Q}_{12} \end{pmatrix}$$

is orthogonal, then

$$\begin{pmatrix} \hat{U}_1^{\mathrm{T}} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} V_1 = \begin{array}{cc} & \begin{array}{cc} p & k-p \end{array} \\ \begin{array}{c} n-k \\ k-p \\ p \end{array} & \begin{pmatrix} \hat{U}_{11}^{\mathrm{T}}\hat{Q}_{11} & 0 \\ 0 & I \\ \bar{Q}_{21} & 0 \end{pmatrix} \end{array}.$$

Since $n - k \geq p$, we may therefore apply QR factorization to $\hat{U}_{11}^{\mathrm{T}}\hat{Q}_{11}$ to obtain

$$\bar{U}_1^{\mathrm{T}}(\hat{U}_{11}^{\mathrm{T}}\hat{Q}_{11}) = \begin{array}{cc} & p \\ \begin{array}{c} p \\ m-k \end{array} & \begin{pmatrix} \bar{Q}_{11} \\ 0 \end{pmatrix} \end{array}$$

Then it is easy to see that the problem is reduced to that of computing the CSD of $p \times p$ square matrices $\bar{Q}_{11}$ and $\bar{Q}_{21}$. ∎

Following Stewart [49], since the diagonal elements c_i and s_i of C and S satisfy

$$c_i^2 + s_i^2 = 1,$$

for some angle θ_i we have $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$. This accounts for the choice of letters C (cosine) and S (sine), and for the name "CS decomposition", in short CSD. Clearly the CSD also simultaneously gives the SVD of both matrices Q_1 and Q_2.

In some literature, the CSD is given in a different form, but they are essentially the same. In particular, the CSD can be given with a decomposition of a partitioned orthogonal matrix as the following result.

Theorem 2.2 (CS Decomposition [48]**)** *Let the orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ be partitioned in the form*

$$Q = \begin{array}{c|cc} & l & n-l \\ \hline l & Q_{11} & Q_{12} \\ n-l & Q_{21} & Q_{22} \end{array},$$

where $2l \le n$. Then there are orthogonal matrices $U = diag(U_{11}, U_{22})$ and $V = diag(V_{11}, V_{22})$ such that

$$U^{\mathrm{T}}QV = \begin{array}{c|ccc} & l & l & n-2l \\ \hline l & C & -S & 0 \\ l & S & C & 0 \\ n-2l & 0 & 0 & I \end{array}$$

where the diagonal matrices C and S satisfy (2.2) in Theorem 2.1.

Armed with the CSD of a partitioned orthonormal matrix, we may now have the following GSVD for any two matrices A and B having the same number of columns:

Theorem 2.3 (GSVD – Triangular Form [40]**)** *If $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, with $rank(A^T,\ B^T)^T = k$, then there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{p\times p}$ and $Q \in \mathbb{R}^{n\times n}$ such that*

$$\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} Q = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix} (\, R,\ 0\,), \tag{2.3}$$

where R is a $k \times k$ nonsingular upper triangular matrix, and the $m \times k$ matrix Σ_1 and $p \times k$ matrix Σ_2 assume one of the forms in Theorem 2.1 depending on the dimensions m, p, k.

Proof. Let

$$\begin{pmatrix} A \\ B \end{pmatrix} = U_1 \begin{pmatrix} R_1 & 0 \\ 0 & 0 \end{pmatrix} V_1^{\mathrm{T}}$$

be the URV decomposition, also called the complete orthogonal factorization, of $(\, A^T,\ B^T\,)^T$ [28], where U_1 and V_1 are $(m+p) \times (m+p)$ and $n \times n$ orthogonal matrices, respectively, R_1 is a nonsingular upper triangular matrix of order k. By partitioning U into the form

$$U_1 = \begin{array}{c|cc} & k & n-k \\ \hline m & U_{11} & U_{12} \\ p & U_{21} & U_{22} \end{array},$$

the result of the theorem follows from the CSD of the partitioned orthonormal matrix $(\, U_{11}^T,\ U_{21}^T\,)^T$. ∎

The advantage of the triangular form of the GSVD is that only orthogonal transformation are required, and it is likely to be more amenable to reliable numerical computation; for more details, see §5. The price to be paid is that the resulting matrices are only upper triangular. If we are also willing to use a nonsingular transformation, we can reduce an arbitrary matrix pair (A, B) having the same number of columns to diagonal forms:

Theorem 2.4 (GSVD – Diagonal Form [56]) *If $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, with rank$(A^T,\ B^T) = k$, then there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{p\times p}$ and a nonsingular matrix $X \in \mathbb{R}^{n\times n}$ such that*

$$\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} X = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix} (\, I_k,\ 0\,), \tag{2.4}$$

where the $m \times k$ diagonal matrix Σ_1 and the $p \times k$ diagonal matrix Σ_2 assume the one of the forms in Theorem 2.1 depending on the relationship of m, p and k.

Proof. Take

$$X = Q \begin{pmatrix} R^{-1} & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

and use Theorem 2.3. ∎

Three remarks are in order;

Remark 1. If B is square and nonsingular, then from the decomposition (2.4), it is easy to see that the GSVD of A and B yields the standard SVD of AB^{-1}: $U^TAB^{-1}V = \Sigma_1\Sigma_2^{-1}$. In particular, if $B = I$, then the GSVD of A and B is the SVD of A. Moreover, when $(A^T,\ B^T)^T$ has orthonormal columns, then the GSVD of A and B is just the CSD of $(A^T,\ B^T)^T$, i.e., $R = I$ in (2.3). In this sense, we may regard the CSD as a special case of the GSVD.

Remark 2. Although the formulation of the GSVD in Theorem 2.3 and 2.4 are mathematically equivalent, the numerical computations of the two decompositions are significantly different, because of the possibly ill-conditioned matrix R in Theorem 2.3 and the matrix X in Theorem 2.4. An algorithm which can numerically stably compute one decomposition can not necessarily compute the other stably because of the possibly ill-conditioned factors R or X with respect to inversion; more details are presented in §5.

Remark 3. It is easy to show that

$$\kappa_2(R) = \kappa_2(X) \leq \kappa_2(G),$$

where $G = (A^T,\ B^T)^T$. Hence we can say that the conditioning of the nonsingular matrix R or X in the GSVD is no worse than the original conditioning of the given matrix G.

Now we are in a position to define the generalized singular values of matrices A and B. The n GENERALIZED SINGULAR VALUE PAIRS (α_i, β_i) of A and B having the same number of columns are defined as follows:

1. $\min\{m, p, k\}$ generalized singular value pairs

$$\alpha_i = c_i, \quad \beta_i = s_i,$$

where c_i and s_i are the diagonal elements of C and S in (2.3) or (2.4);

2. then $k - \min\{m, p, k\}$ generalized singular value pairs
$$\begin{array}{lll} \alpha_i = 1, & \beta_i = 0, & \text{for zero column of } \Sigma_2 \text{ or} \\ \alpha_i = 0, & \beta_i = 1, & \text{for zero column of } \Sigma_1 \end{array}$$

3. finally, $n - k$ generalized singular value pairs
$$\alpha_i = 0, \quad \beta_i = 0$$
corresponding to the zero columns in (2.3) or (2.4).

With loss of generality, we assume that the n generalized singular value pairs (α_i, β_i) of A and B are ordered such that

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n, \qquad \beta_1 \leq \beta_2 \leq \cdots \leq \beta_n. \tag{2.5}$$

The ratios $\sigma_i = \alpha_i/\beta_i$, $i = 1, \ldots, n$, are called the GENERALIZED SINGULAR VALUES. Note that we may have infinite $(1/0)$ or undefined $(0/0)$ types of generalized singular values, which are related to the possible cases of an indefinite generalized symmetric eigenvalue problem [45]. If a given pair of matrices A and B having the same number of columns have no infinite and undefined generalized singular values, the pair is called a REGULAR matrix pair, otherwise, it is called an IRREGULAR matrix pair.

It is customary to call a generalized singular value pair (α_i, β_i) a NONTRIVIAL generalized singular value pair if at least one of α_i and β_i is nonzero, otherwise, it is called a TRIVIAL generalized singular value pair. For a nontrivial generalized singular value pair (α_i, β_i), since $\alpha_i^2 + \beta_i^2 = 1$, for some angle θ_i we have

$$(\alpha_i, \beta_i) = (\cos\theta_i, \sin\theta_i), \quad \sigma_i = \cot\theta_i.$$

Hence, in some situations, we need use only one angle to represent either a generalized singular value pair or a generalized singular value.

We note that the generalized singular value pairs of A and B are independent of column scaling of A and B, i.e., if A and B have the same column scaling: $A_1 = AD$, $B_1 = BD$, then the generalized singular value pairs of A_1 and B_1 are the same as those of A and B since from the GSVD (Theorem 2.4) of A and B, we have

$$\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix} \begin{pmatrix} AD \\ BD \end{pmatrix} (D^{-1}X) = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix} (I,\ 0),$$

which is just the GSVD of A_1 and B_1. We shall refer this property as column scaling independence of the generalized singular value pairs.

To conclude this section, we note that, as above, the GSVD of A and B can be obtained from the spectral decomposition of $A^TA - \lambda B^TB$, which is a generalization of the well known relation between the SVD of A and the spectral decomposition of A^TA. Another well known fact is that the SVD of A can be obtained from the spectral decomposition of Jordan-Wielandt matrix

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}.$$

This fact is generalized in the following theorem, which establishes another way to link the GSVD with the spectral decomposition of a generalized Jordan-Wielandt matrix. For the simplicity of presentation, here we assume that both matrices A and B are square, and B is nonsingular. More general cases can be established in a similar way.

Theorem 2.5 *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$ have the GSVD*

$$U^T A X = \Sigma_1 = diag(\alpha_i), \quad V^T B X = \Sigma_2 = diag(\beta_i),$$

and assume that B is nonsingular. Then the matrix pencil

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} - \lambda \begin{pmatrix} I & 0 \\ 0 & B^T B \end{pmatrix} \tag{2.6}$$

has eigenvalues $\pm\alpha_i/\beta_i, i = 1, \ldots, n$ corresponding to the eigenvectors

$$\begin{pmatrix} u_i \\ \pm\frac{1}{\beta_i} x_i \end{pmatrix}, \quad i = 1, \ldots, n$$

where u_i is the ith column of U and x_i is the ith column of X.

Proof. The proof is a simple computation. Let

$$Z = \frac{1}{\sqrt{2}} \begin{pmatrix} U & U \\ X\Sigma_2^{-1} & -X\Sigma_2^{-1} \end{pmatrix}.$$

Then Z diagonalizes the matrix pencil (2.6) via the congruence transformation $Z^T \times (2.6) \times Z$. ∎

3 Perturbation Theory

In this section, we review the perturbation theory for the CSD and the GSVD. Thus, we discuss the question of how the generalized singular values (pairs) change if the matrices A and B are perturbed by E and F, respectively. As the central question in many matrix perturbation problems, the answer will reflect the conditioning of the decomposition we are concerned with. Sun [53] first gave perturbation bounds for the GSVD. Using a different derivation, Paige [41] gave a perturbation bound for the CSD, and an improved bound for the GSVD. Li [36] discussed the perturbation of the associated subspaces in the GSVD. We shall also present a new perturbation bound which reflects the column scaling invariance property of the generalized singular value pairs. All these analysis give absolute perturbation bounds. A relative perturbation bound for the generalized singular values of a special matrix pair was recently given by Demmel and Veselić [19]. The following is a summary of these results and some new observations.

Theorem 3.1 (The CSD Perturbation Bound [41]) *Let both*

$$Q = \begin{array}{c c} & k \\ \begin{array}{c} m \\ p \end{array} & \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \end{array} \quad and \quad \tilde{Q} = Q + \delta Q = \begin{array}{c c} & k \\ \begin{array}{c} m \\ p \end{array} & \begin{pmatrix} \tilde{Q}_1 \\ \tilde{Q}_2 \end{pmatrix} \end{array}$$

have orthonormal columns, and let the generalized singular value pairs (α_i, β_i) of Q_1 and Q_2, and the generalized singular value pairs $(\tilde{\alpha}_i, \tilde{\beta}_i)$ of $\tilde{Q}_1$ and $\tilde{Q}_2$ be ordered as in (2.5). Then we have

$$\sqrt{\sum_{i=1}^{k} [(\alpha_i - \tilde{\alpha}_i)^2 + (\beta_i - \tilde{\beta}_i)^2]} \le \|\delta Q\|_F. \tag{3.1}$$

The above theorem states the perfect conditioning of the CSD of a partitioned orthonormal matrix. It says that the perturbation to the generalized singular value pairs will be no larger than the perturbation to the original matrix as long as the perturbed matrix still has orthonormal columns. This fact is an immediate consequence of the well conditioning of the SVD since the CSD simultaneously gives the SVD of two subblocks of an orthonormal matrix. However, the following perturbation bound states that the generalized singular value pairs of arbitrary matrices A and B having the same number of columns are not necessarily well conditioned like the CSD; this is essentially because the nonsingular matrix X in Theorem 2.4 or R in Theorem 2.3 can be ill-conditioned. It should be mentioned that the following bound is slightly weaker than the ones presented in [53,41], but the way it is presented here is more useful from the applications point of view.

Theorem 3.2 (The GSVD Perturbation Bound [53,41]) *Let*

$$G = \begin{pmatrix} A \\ B \end{pmatrix}, \quad \tilde{G} = G + \delta G = \begin{pmatrix} \tilde{A} \\ \tilde{B} \end{pmatrix}.$$

Assume rank(G) = *rank*$(\tilde{G}) = n$, *let the generalized singular value pairs* (α_i, β_i) *of* A *and* B *and those* $(\tilde{\alpha}_i, \tilde{\beta}_i)$ *of* $\tilde{A}$ *and* $\tilde{B}$ *be ordered as in (2.5), and assume*

$$\|\delta G\|_F \leq \epsilon \|G\|_2. \tag{3.2}$$

Then

$$\sqrt{\sum_{i=1}^{n}[(\alpha_i - \tilde{\alpha}_i)^2 + (\beta_i - \tilde{\beta}_i)^2]} \leq \sqrt{2}\kappa_2(G)\epsilon, \tag{3.3}$$

where

$$\kappa_2(G) = \|G\|_2 \|G^\dagger\|_2.$$

From (3.3), it is natural to define the number $\kappa_2(G)$ as the *condition number* of the generalized singular value pairs. As we pointed out in §2, the generalized singular value pairs are column scaling invariant. However, the above condition number $\kappa_2(G)$ does not reflect the desired property. One way of dealing with possible artificial ill-conditioning due to bad scaling is to examine the effect of perturbations on individual components, namely componentwise perturbation theory. The results are in the following theorem.

Theorem 3.3 (The First Order GSVD Perturbation Bound [9]) *With the same notation as in Theorem 3.2, if*

$$|\delta G| \leq \epsilon |G|, \tag{3.4}$$

then up to the first order, we have

$$\sqrt{\sum_{i=1}^{n}[(\alpha_i - \tilde{\alpha}_i)^2 + (\beta_i - \tilde{\beta}_i)^2]} \lesssim \sqrt{2}\eta\kappa_{\mathrm{BS}}(G)\epsilon, \tag{3.5}$$

where $\eta = \sqrt{(m+p-n)n}$ *and*

$$\kappa_{\mathrm{BS}}(G) = \|\, |G||G^\dagger| \,\|_2.$$

The number $\kappa_{\rm BS}(G)$ is called the *Bauer-Skeel condition number* of the generalized singular value pairs. It has the property that it is independent of column scaling, which to some extent circumvents the problems of artificial ill-conditioning of the generalized singular value pairs.

For the perturbation of the associated subspaces in the GSVD see [53,36]. We shall not go into the details in this paper.

Since the generalized singular values of A and B are the square roots of eigenvalues of the symmetric matrix pencil $A^TA - \lambda B^TB$, Demmel and Veselić [19] give the following perturbation bound for a special pair of matrices A and B by using the Courant-Fischer minimax theorem. The new bound is a *relative* perturbation bound and reveals the effect of scaling on the generalized singular values[1].

Theorem 3.4 **([19])** *Let A and B be matrices with the same number of columns, and B of full column rank. Let $A = A_1D_1$ and $B = B_1D_2$ where D_1 and D_2 have full rank, and let $E = E_1D_1$ and $F = F_1D_2$ be perturbations of A and B, respectively, such that*

$$||E_1x||_2 \leq \eta_1||A_1x||_2, \quad ||F_1x||_2 \leq \eta_2||B_1x||_2, \tag{3.6}$$

for all x and some $\eta_1, \eta_2 < 1$. Let σ_i and $\tilde{\sigma}_i$ be the i-th generalized singular values of (A, B) and $(A + E, B + F)$, respectively, with the same ordering. Then either $\sigma_i = \tilde{\sigma}_i = 0$ or

$$\frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \frac{\eta_1 + \eta_2}{1 - \eta_2}. \tag{3.7}$$

Note that if both A and B have full column rank, then we have

$$||E_1x||_2 \leq ||E_1||_2||x||_2 \leq \frac{||E_1||_2}{\sigma_{\min}(A_1)}||A_1x||_2,$$

and

$$||F_1x||_2 \leq ||F_1||_2||x||_2 \leq \frac{||F_1||_2}{\sigma_{\min}(B_1)}||B_1x||_2,$$

hence we can choose

$$\eta_1 = \frac{||E_1||_2}{\sigma_{\min}(A_1)}, \quad \eta_2 = \frac{||F_1||_2}{\sigma_{\min}(B_1)}$$

in (3.6). Therefore, substituting η_1 and η_2 into Theorem 3.4 yields the following corollary.

Theorem 3.5 *With the same notation as in Theorem 3.4. If both A and B are of full column rank, then we have*

$$\frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \leq \frac{1}{1 - \eta_2}\left[\kappa_2(A_1)\frac{||E_1||_2}{||A_1||_2} + \kappa_2(B_1)\frac{||F_1||_2}{||B_1||_2}\right]. \tag{3.8}$$

We make two remarks.

Remark 1. Suppose D_1 and D_2 are chosen so that the columns of A_1 and B_1 have unit two-norm. Then $\kappa_2(A_1) \leq n^{1/2}\min_D \kappa_2(AD)$, i.e., this scaling nearly minimizes the condition number of A over all possible column scalings. It is possible that $\kappa_2(A_1) \ll \kappa_2(A)$, and $\kappa_2(B_1) \ll \kappa_2(B)$ [60].

[1] The presentation here is slightly different with original one, but it is easy to show they are equivalent.

Remark 2. Note that in the absolute perturbation bound (3.3), $\kappa_2(G)$ or $\kappa_{\rm BS}(G)$ is used as the condition number of the generalized singular value pairs. However, in the relative perturbation bound (3.8), $\kappa(A_1)$ and $\kappa(B_1)$ are used to measure the conditioning of the generalized singular values. Ignoring the effect of scaling, $\kappa_2(G)$ or $\kappa_{\rm BS}(G)$ could be much smaller than $\kappa_2(A)$ and/or $\kappa_2(B)$, and vice versa.

4 Applications

The CSD and the GSVD are useful in many applications in matrix computations that involve a pair of matrices. In this section, we shall survey some of their applications.

4.1 Intersection of null spaces and the canonical angles between a pair of subspaces

Given $A \in {\rm I\!R}^{m\times n}$ and $B \in {\rm I\!R}^{p\times n}$, consider the problem of finding an orthonormal basis for the common null space ${\rm null}(A) \cap {\rm null}(B)$. The problem has been treated in [28, p.583] by using the SVD twice. Now if we write the GSVD of A and B (2.3) as

$$A(Q_1,\, Q_2) = U\Sigma_1(R,\, 0), \;\; B(Q_1,\, Q_2) = V\Sigma_2(R,\, 0),$$

where $Q_2 \in {\rm I\!R}^{n\times(n-k)}$, then we have

$${\rm null}(A) \cap {\rm null}(B) = {\rm span}\{Q_2\}.$$

Q_2 gives the desired orthonormal basis of the common null space of A and B.

In [51, p.37], the CSD is used as a tool for defining canonical angles between pairs of subspaces, and for computing the singular values of products and differences of two projections.

4.2 Generalized Eigenvalue problem $A^TAx = \lambda B^TBx$

The generalized eigenvalue problem

$$A^TAx = \lambda B^TBx \tag{4.1}$$

arises in MUSIC and ESPRIT direction-of-arrival estimation algorithms in signal processing computations [47,59], where $A \in {\rm I\!R}^{m\times n}$ and $B \in {\rm I\!R}^{p\times n}$ are signal data matrices, and usually $m \geq n$, $p \geq n$. It is certainly unwise to form the cross-products A^TA and B^TB explicitly because of unnecessary roundoff error. As we discussed in §1, the generalized eigenvalue problem (4.1) is equivalent to the GSVD of A and B. Using the diagonal form of the GSVD (2.4), we have

$$X^TA^TAX = \begin{pmatrix} \Sigma_1^T\Sigma_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad X^TB^TBX = \begin{pmatrix} \Sigma_2^T\Sigma_2 & 0 \\ 0 & 0 \end{pmatrix}.$$

Then we know that the nontrivial eigenpairs of the generalized eigenvalue problem (4.1) are given by

$$A^TAx_i = \lambda_iB^TBx_i, \quad i = 1, \ldots, k,$$

where λ_i are the quotients of the diagonal entries of $\Sigma_1^T\Sigma_1$ and $\Sigma_2^T\Sigma_2$, and x_i is the i-th column of X, $k = {\rm rank}(\, A^T, B^T\,)^T$.

4.3 Positive definite generalized eigenvalue problem $Ax = \lambda Bx$

Given symmetric positive definite matrices $A, B \in \mathbb{R}^{n\times n}$, there are two traditional methods for solving the generalized eigenvalue problem

$$Ax = \lambda Bx. \tag{4.2}$$

One is suggested by Martin and Wilkinson [28, p.469] which reduces (4.2) to standard symmetric eigenvalue problems. The other is the QZ algorithm by Moler and Stewart [28, p.394]. Recently, Shougen and Shuqin [46] and Kaufman [34] showed that by reducing the problem (4.2) to the computation of the GSVD, we may be able to compute small eigenvalues more accurately than the traditional methods when the elements are graded or B is ill conditioned. The formulation of using the GSVD to solve the problem (4.2) is as follows.

Consider the Cholesky factorization of A and B given by

$$A = L_1 L_1^T, \quad B = L_2 L_2^T. \tag{4.3}$$

From the discussion of §4.2, we see that the GSVD of L_1 and L_2 gives the eigenvalues and eigenvectors of the generalized eigenvalue problem (4.2). The high accuracy of the Cholesky factorization of a positive definite matrix is discussed in [17].

In particular, if A and B have band structure, then the factors L_1 and L_2 in (4.3) will inherit the structure, and we may take the advantage of the structure in computations, and in particular in parallel processing. For more details, see [34].

4.4 Generalized Total least squares problem

The total least square (TLS) method is a global fitting technique for solving an overdetermined linear system of equations when errors occur in all the data. The technique is also called error-in-variable regression in the statistical literature. The SVD is a fundamental tool in solving the TLS problem. To see how the GSVD is introduced for solving the generalized TLS (GTLS) problem, let us start with the standard TLS problem. Consider a system of m linear equations

$$Ax \approx b, \quad A \in \mathbb{R}^{m\times n}, \quad b \in \mathbb{R}^m, \quad m \geq n+1, \tag{4.4}$$

or

$$\tilde{A}\begin{pmatrix} x \\ -1 \end{pmatrix} \approx 0,$$

where $\tilde{A} = (\,A\ \ b\,)$. If $\tilde{A}$ is of full column rank, then the system (4.4) is inconsistent. Therefore the TLS problem is first to seek a perturbation E of minimal Frobenius norm to $\tilde{A}$, such that the rank of $\tilde{A} + E$ is reduced to n; then a TLS solution x is any solution of

$$(\tilde{A} + E)\begin{pmatrix} x \\ -1 \end{pmatrix} = 0.$$

To find such a perturbation E, from the Schmidt-Mirsky theorem [51], which is also called the Eckart-Young-Mirsky theorem in [28], we know that the desired minimal perturbation matrix E is given by

$$E = U\hat{\Sigma}V^T, \quad \hat{\Sigma} = \mathrm{diag}(0, \ldots, 0, -\sigma_{n+1}),$$

where U and V and σ_{n+1} are from the SVD of $\tilde{A}$: $\tilde{A} = U\Sigma V^T$, $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{n+1})$. Then it is clear that the following system is consistent:

$$(\tilde{A} + E)\begin{pmatrix} x \\ -1 \end{pmatrix} = U\bar{\Sigma}V^T\begin{pmatrix} x \\ -1 \end{pmatrix} = 0,$$

where $\bar{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_n, 0)$. The TLS solution is given by scaling the last column v_{n+1} of V:

$$\begin{pmatrix} x \\ -1 \end{pmatrix} = \frac{-1}{v_{n+1,n+1}} v_{n+1},$$

where we assume that $v_{n+1,n+1} \neq 0$, otherwise we have a nongeneric TLS problem. See [55] for more details.

The above TLS problem has been generalized in two aspects: first, we can allow for multiple right-hand sides in (4.4), i.e., the given system is

$$AX \approx B, \quad A \in \mathbb{R}^{m \times n}, \quad B \in \mathbb{R}^{m \times d}. \tag{4.5}$$

Second, we can allow for a more general form of the perturbation matrix $\tilde{E}$, say $\tilde{E} = ER_C$, where the entries of E are independently and identically distributed with a common variance σ^2, and R_C is the Cholesky factor of the covariance matrix $\mathcal{E}(\tilde{E}^T\tilde{E})$ of the error $\tilde{E}$, i.e., $\mathcal{E}(\tilde{E}^T\tilde{E}) = \sigma^2 R_C^T R_C$. Then the generalized TLS (GTLS) problem can be cast as seeking the minimal perturbation E to $\tilde{A} = (\ A\ B\ )$ such that the following system is consistent:

$$(\tilde{A} + ER_C)\begin{pmatrix} X \\ -I \end{pmatrix} = 0.$$

In [54], a GTLS solution of (4.5) is obtained from the SVD of $\tilde{A}R_C^{-1}$, which is the GSVD of $\tilde{A}$ and R_C. The details of the formula are tedious, and the reader is referred to the work of Van Huffel and Vandewalle [54].

4.5 Truncated GSVD

The GSVD has been used by Hansen [29] to analyze Tikhonov regularization in general form in the same spirit as the SVD is used to analyze standard regularization. The problem is to solve the linear least squares problem

$$\min \|Ax - b\|_2, \tag{4.6}$$

where A is very ill-conditioned. A regularization due to Tikhonov is often used. The idea is that instead of solving problem (4.6), we solve the following least squares problem:

$$\min\{\|Ax - b\|_2^2 + \lambda^2\|Lx\|_2^2\}, \tag{4.7}$$

where the matrix L is chosen to approximate some operator often as derivative. The term $\lambda^2\|Lx\|_2^2$ is used to control the smoothness or shape of the solution x. It is common to assume that

$$A \in \mathbb{R}^{m \times n}, \quad L \in \mathbb{R}^{p \times n}, \quad m \geq n \geq p, \quad \text{rank}(L) = p \tag{4.8}$$

and

$$\text{null}(A) \cap \text{null}(L) = \{0\} \Leftrightarrow \text{rank}(A^T, L^T)^T = n. \tag{4.9}$$

Under the assumptions of (4.8) and (4.9), there is a unique regularized solution, called x_λ, given by

$$x_\lambda = (A^TA + \lambda L^TL)^{-1}A^Tb.$$

Using the diagonal form of the GSVD (Theorem 2.4) of (A, L), it is easy to recast the regularized solution x_λ in the following form

$$x_\lambda = X \begin{pmatrix} C(C^2 - \lambda^2 S^2)^{-1} & 0 & 0 \\ 0 & I & 0 \end{pmatrix} U^T b.$$

Starting with this formula, Hansen [29] discusses the perturbation theory of Tikhonov regularization related to A and L, and the choice of the regularization and truncation parameters.

4.6 The Problems LSE and LSQI

Linear equality or inequality constrained least squares problems arise in constrained surface fitting, constrained optimization, geodetic least squares adjustment, signal processing and other applications [12]. The linear equality constrained least squares (LSE) problem can be stated as follows: find an n-vector x that solves

$$\min_{Bx=d} \|Ax - b\|_2, \tag{4.10}$$

where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times n}$, and $m \geq n \geq p$, $b \in \mathbb{R}^n$, and $d \in \mathbb{R}^p$. Clearly, the Problem LSE has a solution if and only if the equation $Bx = d$ is consistent. For simplicity, it is often assumed that $\text{rank}(B) = p$.

The linear inequality constrained least squares (LSQI) problem is

$$\min \|Ax - b\|_2 \quad \text{subject to} \quad \|Bx - d\|_2 \leq \alpha, \tag{4.11}$$

where $A \in \mathbb{R}^{m\times n} (m \geq n)$, $B \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^n$, $d \in \mathbb{R}^p$ and $\alpha \geq 0$.

It has been shown [58,10,28,12] that the GSVD of A and B in question provides a useful tool to analyze and understand the existing methods (such as weighting techniques) to solve these problems.

4.7 Other applications

Besides what we have discussed, the GSVD has also been used by Kågström [32] in the study of the canonical structure of the matrix pencil $A - \lambda B$. It is known that the column and row nullities of A and B, and the possible common null spaces, give information about the Kronecker structure of $A - \lambda B$. The GSVD can be used to extract the significant information concerning these null spaces. Paige [42] and Bai [5] have used the GSVD for analyzing and solving generalized linear model regression problem. In the block Kaczmarz and Cimmino algorithms for solving large sparse nonsymmetric linear system, the CSD has found to be a useful tool to analyze the conditioning of the orthogonal row projectors used in the procedure [14].

5 Algorithms

Competitive numerical algorithms have been proposed for computing the CSD and the GSVD. In this section, we shall review the existing serial and parallel algorithms for computing these decompositions, and present some modifications to the algorithms. As we have seen in §2, if the CSD is computed, then the GSVD of A and B can be directly obtained from the SVD or the URV decompositions and the CSD. Hence Stewart's [49] and Van Loan's [57] algorithms focus on how to compute the CSD . On the other hand, since the CSD can be regarded as a special case of the GSVD, we should be able to compute the GSVD without going via the CSD. Paige's Jacobi-Kogbetliantz-like algorithm [43] is proposed for directly computing the GSVD.

5.1 Backward Stability

Before we proceed, it is appropriate to state what we mean by the *backward stability* of an algorithm for computing the CSD or the GSVD. As we noted in §2, the different formulations of the CSD and the GSVD may result in different backward stability criteria for the computed decompositions because of the potential ill conditioning of the nonsingular matrix R in Theorem 2.3 and the matrix X in Theorem 2.4. A stable method for one decomposition may not be stable for the other, although the decompositions are mathematically equivalent.

Backward stability for the triangular form of the GSVD (Theorem 2.3) is defined as follows: Let the computed orthogonal matrices be $\bar{U}$, $\bar{V}$ and $\bar{Q}$, the diagonal matrices be $\bar{\Sigma}_1$ and $\bar{\Sigma}_2$, the upper triangular matrix be $\bar{R}$, and let ϵ_M be the machine precision. Then the following conditions should be satisfied:

$$||\bar{U}^T\bar{U} - I|| \approx \epsilon_M, \quad ||\bar{V}^T\bar{V} - I|| \approx \epsilon_M, \quad ||\bar{Q}^T\bar{Q} - I|| \approx \epsilon_M, \tag{5.1}$$

and

$$||\bar{U}^T A\bar{Q} - \bar{\Sigma}_1\bar{R}|| \approx \epsilon_M||A||, \quad ||\bar{V}^T B\bar{Q} - \bar{\Sigma}_2\bar{R}|| \approx \epsilon_M||B||. \tag{5.2}$$

From (5.1) and (5.2), we know that there exist exactly orthogonal matrices $\tilde{U}$, $\tilde{V}$ and $\tilde{Q}$ within ϵ_M of $\bar{U}$, $\bar{V}$ and $\bar{Q}$, and E and F such that

$$||E|| \approx \epsilon_M||A||, \quad ||F|| \approx \epsilon_M||B||, \tag{5.3}$$

and

$$\tilde{U}^T(A+E)\tilde{Q} = \bar{\Sigma}_1\bar{R}, \quad \tilde{V}^T(B+F)\tilde{Q} = \bar{\Sigma}_2\bar{R}. \tag{5.4}$$

These assertions also say that to within roundoff error, the rows of $\bar{U}^T A\bar{Q}$ and $\bar{V}^T B\bar{Q}$ are parallel.

For defining the backward stability of the computed CSD, we just need to change the $\bar{R}$ in (5.2) to be an identity matrix, since if $(A^T, B^T)^T$ has orthonormal columns, the GSVD of A and B is just the CSD of $(A^T, B^T)^T$.

In some applications, we may prefer the diagonal form of the GSVD (Theorem 2.4). In this case, besides criteria (5.1) for the orthogonality of the computed orthogonal matrices $\bar{U}$ and $\bar{V}$, the backward stability of the computed decomposition is defined as

$$\tilde{U}^T(A+E)\bar{X} = \bar{\Sigma}_1, \quad \tilde{V}^T(B+F)\bar{X} = \bar{\Sigma}_2, \tag{5.5}$$

where E and F are as in (5.3).

We should point out that (5.4) and (5.5) are not numerically equivalent. A numerical stable algorithm for computing the decomposition (5.4) may not be stable for the decomposition (5.5) because of the possible numerical unstability in the inverse of the computed $\bar{R}$, and vice versa.

All above the measurements of the backward error are in terms of the norms of the original matrices, so they define *normwise backward stability.*

5.2 Stewart's and Van Loan's Algorithms for computing the CSD

The general problem of computing the CSD of a partitioned orthonormal matrix has four cases, corresponding to the four forms in Theorem 2.1. As shown in the existence of proof for the decomposition, using the QR or RQ factorization all cases can be reduced to computing the CSD of two square matrices. We refer to this reduction as the preprocessing step. Hence in this section, we shall focus on Stewart's and Van Loan's algorithms for computing the CSD of a partitioned orthonormal matrix $Q = (Q_1^T, Q_2^T)^T$, where both Q_1 and Q_2 are square matrices.

5.2.1 Stewart's algorithm for computing the CSD

Although the existence proof the CSD shows that we only need to compute one SVD, and one step of column normalization, it turns out not to be so trivial to compute them in practice with guaranteed numerical accuracy and stability in floating point arithmetic. Stewart [49] put forward the first method for computing the CSD of a partitioned orthonormal matrix $(Q_1^T, Q_2^T)^T$. In Stewart's algorithm, we first compute the SVD of Q_1: $U_1^T Q_1 V = C$, then determine an orthogonal matrix J such that $(Q_2 V)J$ can be column normalized to give a matrix U_2 that is orthogonal to working accuracy. This is accomplished by computing the eigenvalue decomposition of symmetric cross-product matrix $H = V^T Q_2^T Q_2 V$ by Jacobi's method. Finally an orthogonal matrix K is determined such that $K^T CJ$ is diagonal. Stewart's analysis shows a surprising fact, which is that we may take $K = J$, provided we do not perform certain unnecessary Jacobi rotations. Stewart's algorithm is summarized as follows.

Stewart's Algorithm [49]

```
  Compute the SVD $U_1^T Q_1 V = C = \mathrm{diag}(c_i)$;
  $Q_2 := Q_2 V$;
  $H := Q_2^T Q_2$;
  loop until $|h_{ij}| \le n\epsilon_M \sqrt{h_{ii}h_{jj}}$ for all $i, j$, $i \neq j$.
      select pivot indices $i$ and $j$ (say column-ordering);
      if $|h_{ij}| > n\epsilon_M \sqrt{h_{ii}h_{jj}}$ then
         if $c_i + c_j \ge 0.7$ then
           form the Jacobi rotation $J$;
           $V := VJ$;
           $Q_2 := Q_2 J$;
           $U_1 := U_1 J$;
           $H := J^T H J$;
           recompute $h_{ii}$ and $h_{jj}$ from the current $Q_2$ if necessary;
```

```
            end if
        end if
    end loop
    normalize $Q_2$ to give $U_2$ and $S$;
```

Stewart's original stopping criterion $\epsilon_M\sqrt{h_{ii}h_{jj}}$ is modified to $n\epsilon_M\sqrt{h_{ii}h_{jj}}$ in order to take into account that the input partitioned matrix is orthonormal only to working accuracy $n\epsilon_M$. The *ad hoc* quantity 0.7, chosen to decide whether to suppress the rotation, is to guarantee that the diagonal form of C effectively is unchanged in the passage to J^TCJ.

As pointed out by Stewart [49], it is important to note that in the loop of the algorithm, although we are diagonalizing H, it is the object of the process to orthogonalize the columns of Q_2. It might be possible that some of the diagonal elements of H may be degraded because of cancellation during the process. It is recommended to recomputed some of columns of matrix H from the current Q_2. To simplify the implementation of Stewart's algorithm, we propose to use the one-sided Jacobi algorithm to orthogonalize the columns of Q_2 in order to avoid forming the cross-product matrix H explicitly. It also means that we always use the current columns of Q_2. The modification of Stewart's algorithm is as follows, where $q_i^{(2)}$ denotes the i-th column of the current Q_2:

Modified Stewart's Algorithm

```
  Compute the SVD $U_1^TQ_1V = C = \text{diag}(c_i)$;
  $Q_2 := Q_2V$;
  loop until $k = n(n-1)/2$ (all columns of $Q_2$ are numerically effective orthogonal )
      $k = 0$;
      select pivot indices $i$ and $j$ (say row-ordering);
      if $c_i + c_j \geq 0.7$, then
          compute $h_{ii} = q_i^{(2)^T} q_i^{(2)}$;  $h_{ij} = q_i^{(2)^T} q_j^{(2)}$;  $h_{jj} = q_j^{(2)^T} q_j^{(2)}$;
          if $|h_{ij}| > n\epsilon_M\sqrt{h_{ii}h_{jj}}$ then
              form the Jacobi rotation $J$;
              $Q_2 := Q_2J$;
              $V := VJ$;
              $U_1 := U_1J$;
          else
              $k = k + 1$;
          end if
      else
          $k = k + 1$;
      end if
  end loop
  normalize $Q_2$ to give $U_2$ and $S$;
```

Using the one-sided Jacobi procedure for orthogonalizing columns of Q_2 costs $24n$ flops in the inner loop versus $30n$ flops in the Stewart's algorithm. Moreover, one-sided Jacobi rotation simplifies the implementation of the algorithm, and has better numerical stability and accuracy. Empirically, it also reduces the overall number of sweeps. We note that in modified Stewart's algorithm, we also put the test of whether to suppress the (i,j) rotation (i.e., testing $c_i + c_j \geq 0.7$) before computing the quantities h_{ii}, h_{ij} and h_{jj} in order to save the cost of computing these quantities if we need to suppress the (i,j) rotation.

Extensive numerical experiments show that in practice the total number of sweeps of one-sided Jacobi needed to orthogonalize the columns of Q_2 usually do not exceed four. The modified Stewart's algorithm shows better numerical stability and accuracy than original Stewart's algorithm.

5.2.2 Van Loan's algorithm for computing the CSD

Van Loan's algorithm [57] for computing the CSD of a partitioned orthonormal matrix can be regarded as a modification of Stewart's scheme in order to avoid working with the cross-product matrix $H = Q_2^T Q_2$. Van Loan's algorithm is based on the observation that if a well-conditioned matrix has nearly orthogonal columns, then it can be safely diagonalized by the QR factorization. Strictly speaking, the observation is based on the following theorem.

Theorem 5.1 ([57]) *Assume that the $m \times k$ matrix $Y = (y_1, y_2, \ldots, y_k)$ satisfies*

$$Y^T Y = D^2 + E$$

where $D = diag(\|y_1\|, \|y_2\|, \ldots, \|y_k\|)$, and let

$$Y = QR$$

be the QR factorization of Y, where Q is an $m \times k$ orthogonal matrix, and R is $k \times k$ upper triangular. Let Y_i be the first i columns of Y. Then for all i and j ($j > i$) we have

$$|r_{ij}| \leq \min \left\{ \|y_j\|, \frac{\|E\|}{\sigma_{\min}(Y_i)} \right\}.$$

Van Loan's algorithm now can be described as searching for a well-conditioned block (submatrix) among Q_1 and Q_2 to do the diagonalization by the QR decomposition, and using the SVD to diagonalize an ill-conditioned block (submatrix). In more detail, let

$$U_2^T Q_2 V = S = \mathrm{diag}(s_1, s_2, \ldots, s_n)$$

be the SVD of Q_2, where $n \times n$ U_2 and $n \times n$ V are orthogonal matrices, and the diagonal elements of S are increasingly ordered

$$0 \leq s_1 \leq s_2 \leq \cdots \leq s_k \leq 1/\sqrt{2} < s_{k+1} \leq \cdots \leq s_n.$$

Then a QR factorization of the product $Q_1 V$ produces

$$U_1^T (Q_1 V) = R.$$

As we know, R would be diagonal in exact arithmetic. But because of roundoff error, we shall only have

$$R = \begin{array}{c c} & \begin{array}{cc} k & n-k \end{array} \\ \begin{array}{c} k \\ n-k \end{array} & \begin{pmatrix} \mathrm{diag}(c_1, \ldots, c_k) & 0 \\ 0 & R_1 \end{pmatrix} \end{array},$$

where by Theorem 5.1, the first k columns of R correspond to "large" singular values of Q_1. Now by the SVD of submatrix R_1,

$$\tilde{U}_1^T R_1 \tilde{V} = \text{diag}(c_{k+1}, \ldots, c_n),$$

and the QR factorization of the $(n-k)\times(n-k)$ matrix $W = D\tilde{V}$, where $D = \text{diag}(s_{k+1}, \ldots, s_n)$,

$$\tilde{U}_2^T W = R_2,$$

we have

$$R_2 = \text{diag}(s_{k+1}, \ldots, s_n),$$

since $s_{k+1}, \ldots, s_n$ are "large". Combining the previous steps, we have

$$\begin{pmatrix} I & 0 \\ 0 & \tilde{U}_1^T \end{pmatrix} U_1^T Q_1 V \begin{pmatrix} I & 0 \\ 0 & \tilde{V} \end{pmatrix} = \text{diag}(c_1, \ldots, c_n),$$

$$\begin{pmatrix} I & 0 \\ 0 & \tilde{U}_2^T \end{pmatrix} U_2^T Q_2 V \begin{pmatrix} I & 0 \\ 0 & \tilde{V} \end{pmatrix} = \text{diag}(s_1, \ldots, s_n),$$

which is the desired CSD of Q_1 and Q_2.

The *ad hoc* quantity $1/\sqrt{2}$, chosen as the dividing line between the large and small singular values, has the effect of minimizing the backward error bounds of (5.1) and (5.2). One may wish to play with this constant under certain circumstances since the overall amount of work depends on the size of the index k. Large k will result in smaller subproblems, and reduce the total amount of work, but may increase the backward error by a certain factor.

5.3 Stewart's and Van Loan's algorithms for computing the GSVD

As we have seen, the computation of the GSVD is closely related to the computation of the CSD. In this section, we show how to use Stewart's or Van Loan's algorithms to compute the GSVD of matrices A and B having the same number of columns.

The first step is a preprocessing step to compute the QR decomposition of $G = (A^T, B^T)^T$:

$$G = \begin{pmatrix} A \\ B \end{pmatrix} = Q_1 R_1 = \begin{pmatrix} Q_{11} \\ Q_{21} \end{pmatrix} R_1 \tag{5.6}$$

If G is rank deficient, instead of using the QR decomposition, we may use the URV decomposition, the rank-revealing QR decomposition or even the SVD of G [28].

The second step is to use Stewart's or Van Loan's algorithms to compute the CSD of Q_{11} and Q_{21}, denoted as

$$\begin{pmatrix} U_1^T & 0 \\ 0 & U_2^T \end{pmatrix} \begin{pmatrix} Q_{11} \\ Q_{21} \end{pmatrix} V = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix}, \tag{5.7}$$

where Σ_1 and Σ_2 assume one of the forms in Theorem 2.1.

The third step is a postprocessing step to form $W = V^T R_1$, and compute the RQ factorization of W:

$$W = RQ. \tag{5.8}$$

The desired triangular form GSVD of A and B immediately follows from combining (5.6), (5.7) and (5.8).

5.4 Paige's Algorithm for computing the GSVD

Paige proposed a Jacobi-Kogbetliantz approach to compute the GSVD directly, without going via computing the CSD. This algorithm has two phases: the first phase is a preprocessing step to reduce the given matrix pair to triangular forms; the second phase is to compute the GSVD of two triangular matrices by a generalized Kogbetliantz algorithm [35].

5.4.1 Preprocessing

By using QR decomposition, two slightly different preprocessing schemes have been proposed by Paige [43] and Bai and Demmel [7]. Both of the schemes may result in an irregular triangular pair. Most recently, using the URV decomposition, Bai and Zha [8] propose a new preprocessing step, which results in a regular matrix pair, and simplifies the second phase. The results of the preprocessing step are summarized as follows.

Theorem 5.2 ([8]) *Let A be an $m \times n$ matrix and B be a $p \times n$ matrix. Then there are orthogonal matrices U_1, V_1 and Q_1 such that*

$$U_1^T A Q_1 = \begin{array}{c|ccc} & n' & s & r \\ \hline s & 0 & A_{12} & A_{13} \\ r & 0 & 0 & A_{23} \\ m' & 0 & 0 & 0 \end{array}, \quad V_1^T B Q_1 = \begin{array}{c|ccc} & n' & s & r \\ \hline r & 0 & 0 & B_{13} \\ p-r & 0 & 0 & 0 \end{array}, \tag{5.9}$$

where $m' = m - r - s$, $n' = n - r - s$, and A_{12} and B_{13} are nonsingular upper triangular. If $m' \geq 0$, A_{23} is upper triangular, otherwise, A_{23} is upper trapezoidal.

Moreover, if

$$U_2^T A_{23} Q_2 = \Sigma_1 R_1, \quad V_2^T B_{13} Q_2 = \Sigma_2 R_1, \tag{5.10}$$

is the triangular form GSVD of A_{23} and B_{13}, then the triangular form GSVD of A and B is given by

$$U^T A Q = \Sigma_A(\ 0,\ R\ ), \quad V^T B Q = \Sigma_B(\ 0,\ R\ ),$$

where

$$U = U_1 \cdot diag(I, U_2, I), \quad V = V_1 \cdot diag(V_2, I), \quad Q = Q_1 \cdot diag(I, I, Q_2)$$

and

$$R = \begin{pmatrix} A_{12} & A_{13} Q_2^T \\ 0 & R_1 \end{pmatrix}, \quad \Sigma_A = \begin{pmatrix} I & 0 \\ 0 & \Sigma_1 \\ 0 & 0 \end{pmatrix}, \quad \Sigma_B = \begin{pmatrix} 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix},$$

where the block partitions are conformal with the blocks in (5.9).

Proof. The proof is based upon the recursive use of the standard URV and the QR decompositions; see [8] for more details. ∎

In numerical implementation of the preprocessing step, we need to supply a tolerance value, say the machine precision ϵ_M, to determine the effective numerical rank in the URV decomposition of B and a subblock of A.

From Theorem 5.2, we see that the computation of the GSVD of A and B is reduced to computation of the GSVD of two square triangular matrices A_{23} and B_{13} with B_{13}

nonsingular[2]. This is the focus of the next section. For simplicity of notation, we denote the triangular matrix pair (A_{23}, B_{13}) by (A, B) in the next section.

5.4.2 Computing the GSVD of two square upper triangular matrices

We first review the Kogbetliantz algorithm [35] for computing the SVD of an upper triangular matrix A. Then Paige's algorithm [43] for computing the GSVD can be regarded as a generalization of the Kogbetliantz algorithm. Suppose the kth transformation of Kogbetliantz algorithm operates on rows and columns i and j of A. Let

$$A_{ij} = \begin{pmatrix} a_{ii} & a_{ij} \\ 0 & a_{jj} \end{pmatrix}$$

be the 2×2 submatrix subtended by rows and columns i and j of A. Let the rotation matrices

$$U_k = \begin{pmatrix} \cos\phi_k & \sin\phi_k \\ -\sin\phi_k & \cos\phi_k \end{pmatrix}, \qquad V_k = \begin{pmatrix} \cos\psi_k & \sin\psi_k \\ -\sin\psi_k & \cos\psi_k \end{pmatrix}$$

be chosen so that

$$U_k^T A_{ij} V_k = \begin{pmatrix} \gamma_{ii} & 0 \\ 0 & \gamma_{jj} \end{pmatrix} \tag{5.11}$$

is the SVD of the 2×2 triangular matrix A_{ij}. Demmel and Kahan proposed a scheme, called SLASV2 in LAPACK [3], which computes the decomposition (5.11) to nearly full machine precision, barring over/underflow. Let $\hat{U}_k$ and $\hat{V}_k$ be identity matrices with (i,i), (i,j), (j,i) and (j,j) elements replaced by the (1,1), (1,2), (2,1) and (2,2) elements of U_k and V_k, respectively. Then let

$$A_{k+1} = \hat{U}_k^T A_k \hat{V}_k, \tag{5.12}$$

where $A_0 = A$. After the first sweep through all the superdiagonal (i,j) in row cyclic order, the upper triangular matrix A becomes lower triangular. Then the second sweep will restore upper triangular form, and so on [31,30]. In [27,26,44,15,4], it has been proved that if one of the rotation angles $\{\phi_k, \psi_k\}$ at each transformation lies in a closed interval in $(-\pi/2, \pi/2)$, then the sequence $\{A_k\}$ from (5.12) converges initially linearly, and ultimately quadratically, to a diagonal matrix, and gives the SVD of A. For literature on different sweep orders, see [39] and the references therein.

To generalize the Kogbetliantz algorithm for computing the GSVD of upper triangular matrices A and B with B nonsingular, we observe that equivalently, we just need to compute the SVD of $C = AB^{-1}$. A sweep of the Kogbetliantz algorithm applied to the upper triangular matrix C will make it lower triangular. This implies that there are orthogonal matrices U_1 and V_1 such that

$$U_1^T C V_1 = C_1, \tag{5.13}$$

where C_1 is a lower triangular matrix. Writing (5.13) as

$$U_1^T A = C_1 V_1^T B,$$

we see that if we can determine an orthogonal matrix Q_1, which satisfies

$$U_1^T A Q_1 = A_1, \qquad V_1^T B Q_1 = B_1,$$

[2] Without loss of generality, if $m' \leq 0$, we may add some zero rows to A_{23} to get a triangular matrix.

where A_1 and B_1 are lower triangular, then equivalently, we have

$$C_1 = A_1 B_1^{-1}.$$

This reveals that using a sweep of the Kogbetliantz algorithm on an upper triangular C to get a lower triangular C_1 is equivalent to the problem of finding orthogonal matrices U_1, V_1 and Q_1 so that U_1 and V_1 transform the implicitly defined upper triangular matrix C to lower triangular form C_1, and meanwhile $U_1^T A Q_1$ and $V_1^T B Q_1$ are lower triangular. Heath *et al* [31], Paige [43] and Hari and Veselić [30] have shown that we may take advantages of the triangular structure of A and B and the ordering of sweeps to get the desired orthogonal transformations U_1, V_1 and Q_1 without forming B^{-1} and AB^{-1} explicitly. Specifically, at the (i,j) transformation of the Kogbetliantz algorithm, the needed 2×2 submatrix C_{ij} of C can be obtained from

$$C_{ij} = A_{ij} B_{ij}^{-1} = \begin{pmatrix} a_{ii} & a_{ij} \\ & a_{jj} \end{pmatrix} \begin{pmatrix} b_{ii} & b_{ij} \\ & b_{jj} \end{pmatrix}^{-1},$$

where a_{ij} and b_{ij} are the elements subtended by the rows and columns i and j of the updated A and B, respectively. If B_{ij} is numerically ill-conditioned with respect to inversion, then Paige [43] suggests that we may use

$$C_{ij} = A_{ij} \cdot \mathrm{adj}(B_{ij}) = \begin{pmatrix} a_{ii} & a_{ij} \\ & a_{jj} \end{pmatrix} \begin{pmatrix} b_{jj} & -b_{ij} \\ & b_{ii} \end{pmatrix}, \tag{5.14}$$

since $B_{ij} \cdot \mathrm{adj}(B_{ij}) = \det(B_{ij}) I$, where $\mathrm{adj}(B_{ij})$ and $\det(B_{ij})$ stand for the adjugate and determinant of B_{ij}, respectively.

By using the SVD of C_{ij}, $U_{ij}^T C_{ij} V_{ij} = \Sigma_{ij} = \mathrm{diag}(\sigma_{ii}, \sigma_{jj})$, we have

$$U_{ij}^T A_{ij} = \Sigma_{ij} \cdot V_{ij}^T B_{ij}.$$

This shows that the corresponding row vectors of $U_{ij}^T A_{ij}$ and $V_{ij}^T B_{ij}$ are parallel. Hence if we choose a rotation Q_{ij} so that $U_{ij}^T A_{ij} Q_{ij}$ is lower triangular, then $V_{ij}^T B_{ij} Q_{ij}$ must also be lower triangular, and

$$U_{ij}^T A_{ij} Q_{ij} = \Sigma_{ij} \cdot V_{ij}^T B_{ij} Q_{ij}, \tag{5.15}$$

which is just the GSVD of the 2×2 triangular matrices A_{ij} and B_{ij}. With this observation, we see that after completing a sweep in row order, the desired U_1, V_1 and Q_1 are the products $U_{12} U_{13} \cdots U_{n-1,n}$, $V_{12} V_{13}, \cdots V_{n-1,n}$ and $Q_{12} Q_{13} \cdots Q_{n-1,n}$, respectively.

Overall, we are actually carrying out the Kogbetliantz algorithm to diagonalize the implicitly defined matrix $C = AB^{-1}$. At convergence, this gives $U^T(AB^{-1})V = \Sigma$, a diagonal matrix. That is

$$U^T A Q = \Sigma \cdot V^T B Q \tag{5.16}$$

with the i-th row of $U^T AQ$ parallel to the i-th row of $V^T BQ$, which is the desired GSVD of A and B.

A lot of effort has gone into computing the GSVD of the 2×2 matrices A_{ij} and B_{ij} (5.15) accurately and stably in floating point arithmetic, since it is the kernel of the whole algorithm. Careless implementation of the 2×2 GSVD will yield a procedure that is nonconvergent or backward unstable. Many people have contributed to this interesting subproblem; see [43,13,7,2]. Below we state Bai and Demmel's 2×2 GSVD scheme, which has been proved to compute rotations $\bar{U}_{ij}$, $\bar{V}_{ij}$ and $\bar{Q}_{ij}$, such that $\bar{U}_{ij}^T C_{ij} \bar{V}_{ij}$ is diagonal within $\epsilon_M \|C_{ij}\|$, and the rows of $\bar{U}_{ij}^T A_{ij} \bar{Q}_{ij}$ and $\bar{V}_{ij}^T B_{ij} \bar{Q}_{ij}$ are within $\epsilon_M \|A_{ij}\|$ and $\epsilon_M \|B_{ij}\|$, respectively, of being parallel. Recently, Adams *et al* [2] presented another scheme with the same kind of properties. (Note that the ij subscript is dropped for 2×2 matrices for brevity of presentation.)

2×2 GSVD Algorithm[3] ([7])

```
compute the SVD of 2 x 2 $C = A\,\mathrm{adj}(B)$:  $U^T C V = \Sigma$;
compute $G = U^T A$;  $H = V^T B$;  $\hat{G} = |U|^T |A|$;  $\hat{H} = |V|^T |B|$;
/* The angles of $U$ and $V$ are chosen to satisfy the convergence condition. */
if $|c_u| \geq |s_u|$ or $|c_v| \geq |s_v|$ then      /* zero out (1,2) entries of $G$ and $H$ */
    if $\hat{g}_{12}/(|g_{11}| + |g_{12}|) \leq \hat{h}_{12}/(|h_{11}| + |h_{12}|)$ then
        call slartg($-g_{11}, g_{12}, c_q, s_q, r$);
    else
        call slartg($-h_{11}, h_{12}, c_q, s_q, r$);
    end if
else    /* zero out (2,2) entries of $G$ and $H$, then swap rows. */
    if $\hat{g}_{22}/(|g_{21}| + |g_{22}|) \leq \hat{h}_{22}/(|h_{21}| + |h_{22}|)$ then
        call slartg($-g_{21}, g_{22}, c_q, s_q, r$);
    else
        call slartg($-h_{21}, h_{22}, c_q, s_q, r$);
    end if
    $U \leftarrow U\Pi$;  $V \leftarrow V\Pi$;
end if
```

Here, $|A| = (|a_{ij}|)$, Π stands for a 2×2 rotation matrix with angle $\pi/2$. The function of slartg(f, g, c, s, r) is to generate a rotation from f and g to zero out g, i.e., $c = f/\sqrt{f^2 + g^2}$ and $s = g/\sqrt{f^2 + g^2}$. The rotation matrix Q is defined by c_q and s_q.

Summarizing the previous discussion, Paige's direct GSVD algorithm for computing the GSVD of two upper triangular matrices A and B with B nonsingular is presented as the follows. Let k be a user chosen parameter specifying the maximum number of sweeps the algorithm may perform (say, $k = 20$). A_{ij} (B_{ij}) denotes the 2×2 submatrix subtended by rows and columns i and j of A (B).

Paige's direct GSVD Algorithm

```
$sweep := 0$;
if $not\text{-}converged$ and $sweep \leq k$ do
    $sweep := sweep + 1$;
    do $(i,j)$-loop
        compute the GSVD of 2 x 2 matrices $A_{ij}$ and $B_{ij}$;
        $A := U_{ij}^T A Q_{ij}$;
        $B := V_{ij}^T B Q_{ij}$;
        $U := U U_{ij}$; $V := V V_{ij}$; $Q := Q Q_{ij}$;
    end of $(i,j)$-loop
```

[3] In the interest of brevity, we omit the part for the 2 by 2 lower triangular matrices, which can be described similarly.

```
        convergence test;
    end if
    compute the generalized singular value pairs $\alpha_i$ and $\beta_i$;
    order the generalized singular value pairs $\alpha_i$ and $\beta_i$ if desired.
```

The (i,j)-loop in the above algorithm can be simply chosen as the standard row cyclic pivot sequence. From (5.16), it is natural to use the parallelism (linear dependency) of the corresponding rows of $U_k^T AQ_k$ and $V_k^T BQ_k$ as the stopping criterion of the iteration. To measure the parallelism of two n-vectors a and b in high accuracy and against possible over/underflow, in [7] it is proposed that we first compute the QR factorization of the $n \times 2$ matrix $\left(\frac{a}{\|a\|}, \frac{b}{\|b\|}\right)$:

$$\left(\frac{a}{\|a\|}, \frac{b}{\|b\|}\right) = Q\begin{pmatrix} \mu_{11} & \mu_{12} \\ 0 & \mu_{22} \end{pmatrix},$$

and then compute the singular values $\gamma_1 \geq \gamma_2 \geq 0$ of the 2×2 upper triangular matrix $\{\mu_{ij}\}$. Therefore,

$$\mathrm{par}\left(\frac{a}{\|a\|}, \frac{b}{\|b\|}\right) \equiv \gamma_2$$

measures the parallelism (linear dependency) of these two vectors,

For the Paige's direct GSVD algorithm, let a_i and b_i be the i-th row vectors of $U_k^T AQ_k$ and $V_k^T BQ_k$ at the end of a sweep. If

$$\mathrm{error} = \sum_{i=1}^{n} \mathrm{par}\left(\frac{a_i}{\|a_i\|}, \frac{b_i}{\|b_i\|}\right) \leq n\tau$$

for a given tolerance value τ, then the corresponding row vectors of A and B are τ-parallel. This means that there are perturbations of size at most $n\tau\|a_i\|$ in row a_i and $n\tau\|b_i\|$ in row b_i that make them exactly parallel.

5.5 Comparison of the algorithms

The Stewart, Van Loan and Paige's algorithms for computing the CSD or the GSVD are based on different approaches. This section compares the advantages and disadvantages of the three algorithms. Some of the conclusions made here are empirical. With the further development and applications of the algorithms, new observations will be added to this discussion.

From the design and discussion of each algorithm, we know that all algorithms are backward stable in the sense of (5.3) and (5.4). In each algorithm, there are certain parameters which we can use to control the numerical stability of the computed decomposition. The author is not aware of any stable algorithm for computing the diagonal form GSVD in the sense of (5.3) and (5.5).

To compare the costs of the three algorithms, for simplicity of exposition, we assume that the given matrices A and B are $n \times n$ square. Table 5.1 lists floating point operation (flop) counts of the three algorithms for computing generalized singular value pairs only or all factors in the triangular form GSVD. The flop counts are meant only as approximate

Table 5.1: Flop Count of Modified Stewart's, Van Loan's and Paige's Algorithms

	Required Σ_1, Σ_2	Required $U, V, Q, \Sigma_1, \Sigma_2, R$
Stewart		
Preprocessing	$8n^3$	$8n^3$
CSD	$(14 + 6j_1)n^3$	$(23 + 12j_1)n^3$
Postprocessing	0	$4.6n^3$
Van Loan		
Preprocessing	$8n^3$	$8n^3$
CSD	$15.3n^3 + 13.3(n-k)^3$	$26n^3 + 24(n-k)^3 + 6n(n-k)^2$
Postprocessing	0	$4.6n^3$
Paige		
Preprocessing	$2.7n^3$	$5.3n^3$
GSVD-R	$6j_2n^3$	$15j_2n^3$
Postprocessing	0	0

estimates of work; the lower order terms of costs have been omitted.[4] GSVD-R stands for the GSVD of two $n \times n$ triangular matrices.

From Table 5.1, we see that the cost of all three algorithms is of order of n^3. However, the actual cost of each algorithm strongly depends on the distribution of the generalized singular value pairs. j_1 and j_2 are the total number of sweeps required in Stewart's and Paige's algorithms. Empirically, they vary from 2 to 10. In Van Loan's algorithm, the cost of the second SVD depends on k, the number of small singular values from the first SVD.

Largely different scaling of A and B may cause numerical disadvantage to Stewart's and Van Loan's algorithms if we directly combine them together and start with the QR factorization of matrix $G = (A^T, B^T)^T$. This disadvantage can be circumvented if we scale one of the matrices in advance. Specifically, suppose $\|A\| \gg \|B\|$. Then we define $s = \|B\|/\|A\|$. Let

$$\begin{pmatrix} sA \\ B \end{pmatrix} = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix} (\, R,\ 0\,) Q^T$$

be the GSVD of sA and B. Then the GSVD of A and B is given by

$$\begin{aligned} \begin{pmatrix} A \\ B \end{pmatrix} &= \begin{pmatrix} \frac{1}{s}U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \end{pmatrix} (\, R,\ 0\,) Q^T \\ &= \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} \Sigma_1' \\ \Sigma_2' \end{pmatrix} (\, D^{-1}R, 0\,) Q^T \end{aligned}$$

where $\Sigma_1' = \frac{1}{s}\Sigma_1 D$, $\Sigma_2' = \Sigma_2 D$, the diagonal scaling matrix D being chosen so that Σ_1' and Σ_2' are normalized, i.e., $\Sigma_1'^T \Sigma_1' + \Sigma_2'^T \Sigma_2' = I$.

Practical implementation of the three algorithms, with the help of BLAS and the standard linear algebra packages, such as LINPACK [22] and LAPACK [3], is fairly easy, apart

[4] Here we adapted the flop count of basic standard decompositions from [28]: the QRD of $m \times n$ A: $2n^2(m - n/3) + 4(m^2n - mn^2 + n^3/3)$, the SVD of $m \times n$ A: $4m^2n + 8mn^2 + 9n^3$.

from the need to deal with the different cases for the dimensions m, p and n. It should be mentioned that because of restrictions in the existing sophisticated SVD subroutines, such as SSVDC from LINPACK or SGESVD from LAPACK, we cannot pass orthogonal matrices into such subroutines, and update them during the course of computing the SVD. Hence, Van Loan's algorithm has to use two extra $n \times n$ arrays as workspace to store the middle orthogonal matrices in the second SVD, and use them to do updating later.

Extensive numerical experiments on the numerical stability and accuracy of Paige's direct GSVD algorithm with Bai and Demmel's 2×2 GSVD kernel have been reported in [7]. A portable FORTRAN 77 code SGGSVD for computing the generalized singular value decomposition basing Paige's direct algorithm has been developed by Bai, Demmel and Zha. It can be obtained from the author.

5.6 Parallel algorithms

Parallel versions of the three algorithms have been discussed in the literature, although so far there have been no practical implementations.

As we have seen, Van Loan's algorithm is based on two standard decompositions: the QR decomposition and the SVD. Following the parallel algorithms for computing the QR decomposition and the SVD, Kaplan and Van Loan [33] proposed a scheme for computing the CSD in $O(n^2)$ times on an $O(n^2)$ systolic array. Luk and Qiao [38] modified Kaplan and Van Loan's scheme, and obtained an algorithm with $O(n \log n)$ time on $O(n^2)$ processors. As for Stewart's algorithm, it is clear that how a parallel version will depend on the use of a parallel Jacobi method.

In an appendix of Paige's paper [43], he notes that Gentleman has proposed a parallel implementation of his algorithm. Luk [37] discussed the details of Paige's algorithm for computing the GSVD of two $n \times n$ triangular matrices when one of them is nonsingular. A parallel implementation of Paige's algorithm in the general case is discussed by Bai [6], where the proposed scheme costs $O(n^2)$ time for parallel preprocessing, and $O(n^2/p)$ time for the GSVD of two upper trapezoidal matrices, where p is the dimension of the triangular array of processors.

6 Further Generalization of the Singular Value Decomposition

There are many possible further generalizations of the singular value decomposition. We cannot mention all of them in detail in this paper.

First, in some applications, we may be given two matrices A and B having the same number of rows. The problem is what the GSVD formula should be in this case. Such a problem is called the product singular value decomposition (PSVD). In particular, if B is nonsingular, the PSVD should give the SVD of $B^{-1}A$. This subject and its applications have been addressed in [31,25]. Most of the numerical techniques discussed in this paper can be extended to this situation.

The extension of the GSVD of two matrices (A, B) to a triple matrix (A, B, C) has been proposed by Zha [63] under the name of the restricted singular value decomposition (RSVD), and by Ewerbring and Luk [24] under the name of the HK singular value decomposition

(HK-SVD). Numerical experiments are in [13]. Some of them are still subject to debate and further development.

A proposal has been made by De Moor, Golub and Zha [20,21] for standardizing the nomenclature of the generalization of the SVD to any number of matrices with conforming dimensions.

7 Summary and Open Problems

We have seen that since Stewart and Van Loan proposed the CSD of a partitioned orthonormal matrix and the GSVD of two matrices having the same number of columns, exciting progress has been made. It has become an active area of research to study their theoretical background, perturbation analysis, applications and numerical computation. Some of these issues are well understood, and some of them are still subject to debate.

To end this paper, we propose some open problems.

The three existing algorithms are all designed for the stable computation of the GSVD in triangular form (Theorem 2.3). There is no stable algorithm known to compute the GSVD in diagonal form (Theorem 2.4). We suggest that we should try to use the GSVD in triangular form if at all possible. We should be cautious if we have to transform the triangular form GSVD to the diagonal form. The reader is referred to [23] for the discussion of numerical stability of matrix inversion.

Scaling often results in a better conditioned problem. Our new perturbation bound (Theorem 3.3) and Demmel and Veselić's perturbation bound (Theorem 3.4) reflect certain advantages if we first scale the problem appropriately. It is clear that the current three methods cannot take advantage of scaling in the same spirit as the Jacobi method for the eigenvalue and singular value problems [19]. No algorithm has been seen to satisfy the componentwise or columnwise backward stability conditions (3.4) and (3.6). Recent work of Deichmoller [61] is along this line, but further investigation is needed.

In some applications, the given matrices A and B may have certain structure, such as band structure. It is not well understood how one should modify the existing algorithms to take advantage of structure.

Finally, all existing parallel algorithms are as yet theoretical. The algorithm design, numerical implementation and performance evaluation on real parallel machines will be the subject of future study.

Acknowledgement

The author acknowledges Jim Demmel, Nick Higham, Pete Stewart and Hongyuan Zha for fruitful discussions on this work and for their comments on the manuscript.

This work was completed while the author was visitor at the Institute for Mathmatics and its Applications, University of Minnesota. This work was supported in part by NSF grant ASC-9102963 and by the Applied and Computational Mathematics Program, Defense Advanced Research Projects Agency, under contract DM28E04120.

References

[1] *IEEE Standard for Binary Floating Point Arithmetic.* ANSI/IEEE, New York, Std 754–1985 edition, 1985.

[2] G. E. Adams, A. W. Bojanczyk, and F. T. Luk, Computing the PSVD of two 2×2 triangular matrices, manuscript, 1991.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. Mckenney, S. Ostrouchov and D. Sorensen, *LAPACK Users' Guide*, to appear, SIAM Press, Philadelphia, 1992.

[4] Z. Bai, Note on the quadratic convergence of Kogbetliantz algorithm for computing the singular value decomposition, Lin. Alg. and Its Appl., 104, (1988), pp.131–140.

[5] Z. Bai, Numerical treatment of restricted Gauss-Markov linear model, Comm. in Statis. B17 No.2, (1988), pp.131-140.

[6] Z. Bai, A Parallel algorithm for computing the generalized singular value decomposition, Research report 90-04, Department of Mathematics, University of Kentucky, Lexington, Nov. 1990.

[7] Z. Bai and J. Demmel, Computing the generalized singular value decomposition. Report No. UCB/CSD 91/645, Computer Science Division, University of California, Berkeley, August 1991. Also as Research Report 91-09, Department of Mathematics, University of Kentucky, Lexington, August 1991.

[8] Z. Bai and H. Zha, A new preprocessing algorithm for the computation of the generalized singular value decomposition, manuscript, 1992.

[9] Z. Bai, A componentwise perturbation bound of the the generalized singular value decomposition, manuscript, 1992.

[10] J. L. Barlow, Error analysis and implementation aspects of deferred correction for equality constrained least squares problems. SIAM J. Numer. Anal., 25 (1988), pp. 1340–1358.

[11] Å. Björck and G. H. Golub, Numerical methods for computing angles between linear subspaces, Math. Comp. 27, (1973), pp.579–594.

[12] Å. Björck, Solution of equations in $\mathbb{R}^n$: Least Squares Methods, in P.G. Ciarlet and J.L. Lions ed., Handbook of Numerical Analysis, Volume I, Elsevier/North Holland, 1990.

[13] A. W. Bojanczyk, M. Ewerbring, F. T. Luk and P. van Dooren, An accurate product SVD algorithm, in SVD and Signal Processing, II, Algorithms, Analysis and Applications, R. J. Vaccaro ed. Elsevier Sci. Publishers, B.V. 1991.

[14] R. Bramley and A. Sameh, Row projection methods for large nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13, (1992), pp.168–193.

[15] J. P. Charlier and P. Van Dooren, On Kogbetliantz's SVD algorithm in the presence of Clusters, Lin. Alg. and Its Appl. 95, (1987), pp136–160.

[16] C. Davis and W. Kahan, The rotation of eigenvectors by a perturbation. III, SIAM J. Numer. Anal. 7, (1970), pp.1–46.

[17] J. Demmel, On floating point errors in Cholesky, LAPACK Working Note # 14, Univ. of Tenn. CS-89-87, Oct. 1989.

[18] J. Demmel and W. Kahan, Accurate Singular Values of Bidiagonal Matrices, SIAM J. Sci. Stat. Comput. 11, (1990), pp.873-912.

[19] J. Demmel and K. Veselić, Jacobi's method is more accurate than QR, to appear in SIAM J. Mat. Anal. Appl., 1992.

[20] B. L. R. De Moor and G. H. Golub, Generalized Singular Value Decompositions: A proposal for a standardized nomenclature, Manuscript NA-89-05, Numerical Analysis Project, Stanford University, Stanford, Calif., 1989

[21] B. L. R. De Moor and H. Zha, A tree of generalizations of the ordinary singular value decomposition, ESAT-SISTA report 1989-21, Katholieke Universiteit Leuven, Belgium, 1989.

[22] J. J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart, LINPACK User's Guide, SIAM Press, Philadelphia, 1978.

[23] J. J. Du Croz and N. J. Higham, Stability of methods for matrix inversion, IMA J. Numer. Anal. 12, (1992), pp.1–20.

[24] L. M. Ewerbring and F. T. Luk, Canonical correlations and generalized SVD: Applications and New Algorithms, J. Comput. Applied Math. 27 (1989), pp.37-52.

[25] K. V. Fernando and S. J. Hammarling, A product induced singular value decomposition for two matrices and balanced realization, in Linear Algebra in Signals Systems and Control, B. N. Datta et al eds. SIAM Press, Philadelphia, (1988), pp.128–140.

[26] K. V. Fernando, Linear convergence of the row cyclic Jacobi and Kogbetliantz methods, Numer. Math. 56, (1989), pp.73–91.

[27] G. E. 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.

[28] G. H. Golub and C. F. Van Loan, Matrix Computations (2nd ed), The Johns Hopkins Univ. Press, Baltimore, MD, 1989.

[29] P. C. Hansen, Regularization, GSVD and truncated GSVD, BIT, (1989), pp.491–504.

[30] V. Hari and K. Veselić: On Jacobi methods for singular value decomposition, SIAM J. of Sci. Stat. Comp. 8. (1987), pp.741–754.

[31] M. T. Heath, J. A. Laub, C. C. Paige and R. C. Ward, Computing the singular value decomposition of a product of two matrices. SIAM J. Sci. Stat. Comput. 7, (1986), pp.1147–1159.

[32] B. Kågström, The generalized singular value decomposition and $(A - \lambda B)$-problem. BIT 24, (1984), pp.568–583.

[33] I. M. Kaplan and C. Van Loan, On computing the CS-decomposition with systolic arrays, Tech. Report TR84-647, Computer Science Department, Cornell Univ. Ithaca, NY, 1984.

[34] L. Kaufman, Modifying the Shougen-Shuquin algorithm for the banded symmetric generalized matrix eigenvalue problem. Bell Lab. Tech. Rep. 1991.

[35] E. G. Kogbetliantz, Solution of linear equations by diagonalization of coefficients matrix, Quart. Appl. Math. 13, (1955), pp.123–132.

[36] R. Li, Bounds on perturbations of generalized singular values and associated subspaces, to appear at SIAM J. Mat. Anal. Appl.

[37] F. T. Luk, A parallel method for computing the generalized singular value decomposition, J. Para. Dist. Comp. 2, (1985), pp.250–260.

[38] F. T. Luk and S. Qiao, Computing the CS-decomposition on systolic arrays, SIAM J. Sci. Stat. Comput. 7, (1986), pp.1121–1125.

[39] F. T. Luk and H. T. Park, On parallel Jacobi orderings, SIAM J. Sci. Stat. Comput. Vol.10, (1989), pp.18–26.

[40] C. C. Paige and M. A. Saunders, Towards a generalized singular value decomposition, SIAM J. Numer. Anal. 18, (1981), pp.398–405.

[41] C. C. Paige, A note on a result of Sun Ji-guang: sensitivity of the CS and GSV decomposition, SIAM J. Numer. Anal. 21, (1984), pp.186–191.

[42] C. C. Paige, The general linear model and generalized singular value decomposition, Lin. Alg. Appl. 70, (1985), pp.269–284.

[43] C. C. Paige, Computing the generalized singular value decomposition, SIAM J. Sci. Stat. Comput. 7, (1986), pp.1126–1146.

[44] C. C. Paige and P. Van Dooren, A note on the convergence of Kogbetliantz's iterative algorithm for obtaining the singular value decomposition, Lin. Alg. and Appl. 77, (1986), pp.301–313.

[45] B. N. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[46] W. Shougen and Z. Shuquin, An algorithm for $Ax = \lambda Bx$ with symmetric positive definite A and B, SIAM J. Mat. Anal. Appl. 1991.

[47] J. M. Speiser and C. F. Van Loan, Signal processing computations using the generalized singular value decomposition, Proc. SPIE Vol.495, Real Time Signal Processing VII, (1984), pp.47-55.

[48] G. W. Stewart, On the perturbation of pseudo-inverses, projections and linear least squares problems, SIAM Rev. 19, (1977). pp.634–662.

[49] G. W. Stewart, Computing the CS-decomposition of a partitioned orthonormal matrix, Numer. Math. 40. pp.297-306(1982).

[50] G. W. Stewart, An Updating Algorithm for Subspace Tracking, Department of Computer, Science, University of Maryland, CS-TR 2494, 1990. To appear in IEEE Transactions on Signal Processing.

[51] G. W. Stewart and J. Sun, Matrix Perturbation Theory, Academic Press, NY, 1991.

[52] G. W. Stewart, On the early history of the singular value decomposition, IMA Tech. Rep. series 952, Univ. of Minnesota, 1992.

[53] Sun Ji-guang, Perturbation analysis for the generalized singular value problem, SIAM J. Numer. Anal. 20, (1983). pp.611–625.

[54] S. Van Huffel and J. Vandewalle, Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in A are subject to error, SIAM J. Mat. Anal. Appl. 10, (1989), pp.294–315.

[55] S. Van Huffel and J. Vandewalle, The Total Least Squares Problem, Computational Aspects and Analysis, SIAM Press, Philadelphia, 1991.

[56] C. F. Van Loan, Generalizing the singular value decomposition, SIAM J. Numer. Anal. 13, (1976), pp.76–83.

[57] C. F. Van Loan, Computing the CS and the generalized singular value decomposition, Numer. Math. 46, (1985), pp.479–491.

[58] C. F. Van Loan, On the method of weighting for equality-constrained least-squares problems, SIAM J. Numer. Anal. 22, (1985), pp.851–864.

[59] C. F. Van Loan, A unitary method for the ESPRIT direction-of-arrival estimation algorithm, 87-862, Department of Computer Science, Cornell Univ. Ithaca, 1987

[60] A. van der Sluis, Condition numbers and equilibration of matrices, Numer. Math. 14, (1969), pp.241–254.

[61] K. Veselić, private communication, 1992

[62] J. W. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, England, 1965.

[63] H. Zha, A numerical algorithm for computing restricted singular value decomposition of matrix triplets, to appear in Lin. Alg. & Appl., 168, (1992), pp.1–26.

Appendix: Specifications of SGGSVD

```
      SUBROUTINE SGGSVD( COMPU, COMPV, COMPQ, M, P, N, KA, KB, A, LDA,
     $                   B, LDB, ALPHA, BETA, U, LDU, V, LDV, Q, LDQ,
     $                   IWORK, WORK, INFO )
*
* ======================================================================
*
*    Copyright (C) 1992, Zhaojun Bai, James Demmel and Hongyuan Zha
*    All rights reserved.  Use at your own risk.  April 3, 1992
*    Please send any comments by e-mail to
*
*      bai@ms.uky.edu   or
*      demmel@wsparc.berkeley.edu  or
*      zha@cholesky.stanford.edu
*
* ======================================================================
*
*     .. Scalar Arguments ..
      CHARACTER          COMPQ, COMPU, COMPV
      INTEGER            INFO, KA, KB, LDA, LDB, LDQ, LDU, LDV, M, N, P
*     ..
*     .. Array Arguments ..
      INTEGER            IWORK( * )
      REAL               A( LDA, * ), ALPHA( * ), B( LDB, * ),
     $                   BETA( * ), Q( LDQ, * ), U( LDU, * ),
     $                   V( LDV, * ), WORK( * )
*     ..
*
*  Purpose
*  =======
*
*  SGGSVD computes the generalized singular value decomposition of the
*  M by N matrix A and P by N matrix B:
*
*          [ U'   ]*[ A ]*Q = [ D1 ]*[ 0 R ]
*          [    V'] [ B ]     [ D2 ]
*
*  where U, V and Q are orthogonal matrices, R is an upper tridiagonal
*  matrix, D1 and D2 are "diagonal" matrices,  KA and KB are integer
*  outputs defined as follows and Z' means the transpose of Z.
*
*  If M-KA-KB >= 0,
*
*     U'*A*Q = D1*[ 0 R ]
*            = KA      [ I   0 ] * [   0      R11  R12 ] KA
*              KB      [ 0   C ]   [   0       0   R22 ] KB
*              M-KA-KB [ 0   0 ]
*
*     V'*B*Q = D2*[ 0 R ]
*            = KB      [ 0   S ] * [   0      R11  R12 ] KA
*              P-KB    [ 0   0 ]   [   0      0    R22 ] KB
*                        KA  KB      N-KA-KB   KA   KB
*  where
*
*    C = diag( ALPHA(KA+1), ... , ALPHA(KA+KB) ),
*    S = diag( BETA(KA+1),  ... , BETA(KA+KB) ), C**2 + S**2 = I.
*    The nonsingular triangular matrix R = [ R11 R12 ]
*                                          [   0 R22 ]
*    is stored in A(1:KA+KB,N-KA-KB+1:N).
*
*  if M-KA-KB < 0,
*
*  U'*A*Q = D1*[ 0 R ]
*         = KA     [ I   0    0   ] * [ 0     R11  R12  R13   ] KA
*           M-KA   [ 0   C    0   ]   [ 0      0   R22  R23   ] M-KA
*                                     [ 0      0     0  R33   ] KA+KB-M
*
*  V'*B*Q = D2*[ 0 R ]
*         = M-KA   [ 0   S    0   ] * [ 0     R11  R12  R13   ] KA
*           KA+KB-M[ 0   0    I   ]   [ 0     0    R22  R23   ] KB
*           P-KB   [ 0   0    0   ]   [ 0     0     0   R33   ] KA+KB-M
*                    KA M-KA KA+KB-M  N-KA-KB  KA  M-KA KA+KB-M
*
*  where
*
*    C = diag( ALPHA(KA+1), ... , ALPHA(M) ),
*    S = diag( BETA(KA+1),  ... , BETA(M) ), C**2 + S**2 = I.
*    R = [ R11 R12 R13 ] is a nonsingular upper triangular matrix,
*        [  0  R22 R23 ]
*        [  0   0  R33 ]
*    [R11 R12 R13 ] is stored in A(1:M, N-KA-KB+1:N)
*    [ 0  R22 R23 ]
*    and R33 is stored in B(M-KA+1:KB,N+M-KA-KA+1:N)
*
*  The computation of the orthogonal transformation matrices U, V and Q
*  are optional.
*
*  The algorithm is based on the following papers:
*
*   C. Paige, Computing the generalized singular value decomposition,
*      SIAM J. Sci. Stat. Comput. 7(1986), pp.1126-1146.
*   Z. Bai and J. Demmel, Computing the generalized singular value
*      decomposition. No. UCB/CSD 91/645, Computer Science Division,
*      Univ. of Calif.,  Berkeley, August 1991. Also as Research report
*      91-09, Dept. of Math., Univ. of Kentucky, Lexington, August 1991.
*   Z. Bai and H. Zha, A new preprocessing algorithm for the computation
*      of the generalized singular value decomposition, manuscript,1992.
*
*  Arguments
*  =========
*
*  COMPU   (input) CHARACTER*1
*          = 'N': Orthogonal matrix U is not computed;
*          = 'U': U is computed.
*
```

```
*  COMPV   (input) CHARACTER*1
*          = 'N': Orthogonal matrix V is not computed;
*          = 'V': V is computed
*
*  COMPQ   (input) CHARACTER*1
*          = 'N': Orthogonal matrix Q is not computed;
*          = 'Q': Q is computed.
*
*  M       (input) INTEGER
*          The number of rows of the matrix A.  M >= 0.
*
*  P       (input) INTEGER
*          The number of rows of the matrix B.  P >= 0.
*
*  N       (input) INTEGER
*          The number of columns of the matrices A and B.  N >= 0.
*
*  KA      (output) INTEGER
*  KB      (output) INTEGER
*          The index of subblocks in the decomposition, see Purpose
*          for details.
*
*  A       (input/output) REAL array, dimension (LDA,N)
*          On entry, A contains the matrix whose decomposition is to
*          be computed.
*          On exit, A contains the triangular matrix R, or part of R,
*          see Purpose for details.
*
*  LDA     (input) INTEGER
*          The leading dimension of matrix. LDA >= MAX(M,1).
*
*  B       (input/output) REAL array, dimension (LDB,N)
*          On entry, B contains the matrix whose decomposition is to
*          be computed.
*          On exit, B contains the triangular matrix R if appropriate,
*          see Purpose for details.
*
*  LDB     (input) INTEGER
*          The leading dimension of matrix. LDA >= MAX(P,1).
*
*  ALPHA   (output) REAL array, dimension (KA+KB)
*  BETA    (output) REAL array, dimension (KA+KB)
*          On exit, ALPHA and BETA contain the nontrivial generalized
*          singular value pairs of A and B;
*          if M-KA-KB >= 0,
*            ALPHA(1:KA) = ONE,  ALPHA(KA+1:KA+KB) = C,
*            BETA(1:KA)  = ZERO, BETA(KA+1:KA+KB)  = S,
*          if M-KA-KB < 0,
*            ALPHA(1:KA)=ONE,  ALPHA(KA+1:M)=C, ALPHA(M+1:KA+KB)=ZERO
*            BETA(1:KA) =ZERO, BETA(KA+1:M) =S, BETA(M+1:KA+KB) =ONE
*
*  U       (output) REAL array, dimension (LDU,M)
*          On exit, if COMPU = 'U', U contains the orthogonal matrix
*           which premultiplies matrix A.
*           If COMPU = 'N', U is not referenced.
*
*  LDU      (input) INTEGER
*           The leading dimension of matrix U. LDU >= MAX(M,1).
*
*  V        (output) REAL array, dimension (LDV,P)
*           On exit, if COMPU = 'V', V contains the orthogonal matrix
*           which premultiplies matrix B.
*           If COMPV = 'N', U is not referenced.
*
*  LDV      (input) INTEGER
*           The leading dimension of matrix V. LDA >= MAX(P,1).
*
*  Q        (output) REAL array, dimension (LDQ,N)
*           On exit, if COMPU = 'Q', Q contains the orthogonal matrix
*           which post-multiplies matrix B.
*           If COMPQ = 'N', Q is not referenced.
*
*  LDQ      (input) INTEGER
*           The leading dimension of matrix. LDQ >= MAX(N,1).
*
*  IWORK    (workspace) INTEGER array, dimension (N)
*
*  WORK     (workspace) REAL array, dimension (MAX( M, P, 3*N )+N)
*
*  INFO     (output)INTEGER
*           On exit, if INFO
*             = 0; successful return.
*             < 0; if INFO = -i, the i-th argument had an illegal value.
*             > 0; if INFO = 1, the preprocessing failed.
*                  if INFO = 2, the Kogbetliantz procedure failed.
*
* ======================================================================
```

Recent IMA Preprints

#	Author/s	Title
866	**Gui–Qiang Chen and Tai–Ping Liu**,	Zero relaxation and dissipation limits for hyperbolic conservation laws
867	**Gui–Qiang Chen and Jian–Guo Liu**,	Convergence of second-order schemes for isentropic gas dynamics
868	**Aleksander M. Simon and Zbigniew J. Grzywna**,	On the Larché–Cahn theory for stress-induced diffusion
869	**Jerzy Łuczka, Adam Gadomski and Zbigniew J. Grzywna**,	Growth driven by diffusion
870	**Mitchell Luskin and Tsorng-Whay Pan**,	Nonplanar shear flows for nonaligning nematic liquid crystals
871	**Mahmoud Affouf**,	Unique global solutions of initial-boundary value problems for thermodynamic phase transitions
872	**Richard A. Brualdi and Keith L. Chavey**,	Rectangular L-matrices
873	**Xinfu Chen, Avner Friedman and Bei Hu**,	The thermistor problem with zero–one conductivity II
874	**Raoul LePage**,	Controlling a diffusion toward a large goal and the Kelly principle
875	**Raoul LePage**,	Controlling for optimum growth with time dependent returns
876	**Marc Hallin and Madan L. Puri**,	Rank tests for time series analysis a survey
877	**V.A. Solonnikov**,	Solvability of an evolution problem of thermocapillary convection in an infinite time interval
878	**Horia I. Ene and Bogdan Vernescu**,	Viscosity dependent behaviour of viscoelastic porous media
879	**Kaushik Bhattacharya**,	Self-accommodation in martensite
880	**D. Lewis, T. Ratiu, J.C. Simo and J.E. Marsden**,	The heavy top: a geometric treatment
881	**Leonid V. Kalachev**,	Some applications of asymptotic methods in semiconductor device modeling
882	**David C. Dobson**,	Phase reconstruction via nonlinear least-squares
883	**Patricio Aviles and Yoshikazu Giga**,	Minimal currents, geodesics and relaxation of variational integrals on mappings of bounded variation
884	**Patricio Aviles and Yoshikazu Giga**,	Partial regularity of least gradient mappings
885	**Charles R. Johnson and Michael Lundquist**,	Operator matrices with chordal inverse patterns
886	**B.J. Bayly**,	Infinitely conducting dynamos and other horrible eigenproblems
887	**Charles M. Elliott and Stefan Luckhaus**,	'A generalised diffusion equation for phase separation of a multi-component mixture with interfacial free energy'
888	**Christian Schmeiser and Andreas Unterreiter**,	The derivation of analytic device models by asymptotic methods
889	**LeRoy B. Beasley and Norman J. Pullman**,	Linear operators that strongly preserve the index of imprimitivity
890	**Jerry Donato**,	The Boltzmann equation with lie and cartan
891	**Thomas R. Hoffend Jr., Peter Smereka and Roger J. Anderson**,	A method for resolving the laser induced local heating of moving magneto-optical recording media
892	**E.G. Kalnins, Willard Miller, Jr. and Sanchita Mukherjee**,	Models of q-algebra representations: the group of plane motions
893	**T.R. Hoffend Jr. and R.K. Kaul**,	Relativistic theory of superpotentials for a nonhomogeneous, spatially isotropic medium
894	**Reinhold von Schwerin**,	Two metal deposition on a microdisk electrode
895	**Vladimir I. Oliker and Nina N. Uraltseva**,	Evolution of nonparametric surfaces with speed depending on curvature, III. Some remarks on mean curvature and anisotropic flows
896	**Wayne Barrett, Charles R. Johnson, Raphael Loewy and Tamir Shalom**,	Rank incrementation via diagonal perturbations
898	**Mingxiang Chen, Xu-Yan Chen and Jack K. Hale**,	Structural stability for time-periodic one-dimensional parabolic equations
899	**Hong-Ming Yin**,	Global solutions of Maxwell's equations in an electromagnetic field with the temperature-dependent electrical conductivity
900	**Robert Grone, Russell Merris and William Watkins**,	Laplacian unimodular equivalence of graphs
901	**Miroslav Fiedler**,	Structure-ranks of matrices
902	**Miroslav Fiedler**,	An estimate for the nonstochastic eigenvalues of doubly stochastic matrices
903	**Miroslav Fiedler**,	Remarks on eigenvalues of Hankel matrices
904	**Charles R. Johnson, D.D. Olesky, Michael Tsatsomeros and P. van den Driessche**,	Spectra with positive elementary symmetric functions
905	**Pierre-Alain Gremaud**,	Thermal contraction as a free boundary problem
906	**K.L. Cooke, Janos Turi and Gregg Turner**,	Stabilization of hybrid systems in the presence of feedback delays
907	**Robert P. Gilbert and Yongzhi Xu**,	A numerical transmutation approach for underwater sound propagation
908	**LeRoy B. Beasley, Richard A. Brualdi and Bryan L. Shader**,	Combinatorial orthogonality
909	**Richard A. Brualdi and Bryan L. Shader**,	Strong hall matrices
910	**Håkan Wennerström and David M. Anderson**,	Difference versus Gaussian curvature energies; monolayer versus bilayer curvature energies applications to vesicle stability
911	**Shmuel Friedland**,	Eigenvalues of almost skew symmetric matrices and tournament matrices
912	**Avner Friedman, Bei Hu and J.L. Velazquez**,	A Free Boundary Problem Modeling Loop Dislocations in Crystals
913	**Ezio Venturino**,	The Influence of Diseases on Lotka-Volterra Systems
914	**Steve Kirkland and Bryan L. Shader**,	On Multipartite Tournament Matrices with Constant Team Size

915 **Richard A. Brualdi and Jennifer J.Q. Massey**, More on Structure-Ranks of Matrices

916 **Douglas B. Meade**, Qualitative Analysis of an Epidemic Model with Directed Dispersion

917 **Kazuo Murota**, Mixed Matrices Irreducibility and Decomposition

918 **Richard A. Brualdi and Jennifer J.Q. Massey**, Some Applications of Elementary Linear Algebra in Combinations

919 **Carl D. Meyer**, Sensitivity of Markov Chains

920 **Hong-Ming Yin**, Weak and Classical Solutions of Some Nonlinear Volterra Integrodifferential Equations

921 **B. Leinkuhler and A. Ruehli**, Exploiting Symmetry and Regularity in Waveform Relaxation Convergence Estimation

922 **Xinfu Chen and Charles M. Elliott**, Asymptotics for a Parabolic Double Obstacle Problem

923 **Yongzhi Xu and Yi Yan**, An Approximate Boundary Integral Method for Acoustic Scattering in Shallow Oceans

924 **Yongzhi Xu and Yi Yan**, Source Localization Processing in Perturbed Waveguides

925 **Kenneth L. Cooke and Janos Turi**, Stability, Instability in Delay Equations Modeling Human Respiration

926 **F. Bethuel, H. Brezis, B.D. Coleman and F. Hélein**, Bifurcation Analysis of Minimizing Harmonic Maps Describing the Equilibrium of Nematic Phases Between Cylinders

927 **Frank W. Elliott, Jr.**, Signed Random Measures: Stochastic Order and Kolmogorov Consistency Conditions

928 **D.A. Gregory, S.J. Kirkland and B.L. Shader**, Pick's Inequality and Tournaments

929 **J.W. Demmel, N.J. Higham and R.S. Schreiber**, Block LU Factorization

930 **Victor A. Galaktionov and Juan L. Vazquez**, Regional Blow-Up in a Semilinear Heat Equation with Convergence to a Hamilton-Jacobi Equation

931 **Bryan L. Shader**, Convertible, Nearly Decomposable and Nearly Reducible Matrices

932 **Dianne P. O'Leary**, Iterative Methods for Finding the Stationary Vector for Markov Chains

933 **Nicholas J. Higham**, Perturbation theory and backward error for $AX - XB = C$

934 **Z. Strakos and A. Greenbaum**, Open questions in the convergence analysis of the lanczos process for the real symmetric eigenvalue problem

935 **Zhaojun Bai**, Error analysis of the lanczos algorithm for the nonsymmetric eigenvalue problem

936 **Pierre-Alain Gremaud**, On an elliptic-parabolic problem related to phase transitions in shape memory alloys

937 **Bojan Mohar and Neil Robertson**, Disjoint essential circuits in toroidal maps

938 **Bojan Mohar**, Convex representations of maps on the torous and other flat surfaces

939 **Bojan Mohar and Svatopluk Poljak** Eigenvalues in combinatorial optimization

940 **Richard A. Brualdi, Keith L. Chavey and Bryan L. Shader**, Conditional sign-solvability

941 **Roger Fosdick and Ying Zhang**, The torsion problem for a nonconvex stored energy function

942 **René Ferland and Gaston Giroux**, An unbounded mean-field intensity model:
Propagation of the convergence of the empirical laws and compactness of the fluctuations

943 **Wei-Ming Ni and Izumi Takagi**, Spike-layers in semilinear elliptic singular Perturbation Problems

944 **Henk A. Van der Vorst and Gerard G.L. Sleijpen**, The effect of incomplete decomposition preconditioning on the convergence of conjugate gradients

945 **S.P. Hastings and L.A. Peletier**, On the decay of turbulent bursts

946 **Apostolos Hadjidimos and Robert J. Plemmons**, Analysis of p-cyclic iterations for Markov chains

947 **ÅBjörck, H. Park and L. Eldén**, Accurate downdating of least squares solutions

948 **E.G. Kalnins, William Miller, Jr. and G.C. Williams**, Recent advances in the use of separation of variables methods in general relativity

949 **G.W. Stewart**, On the perturbation of LU, Cholesky and QR factorizations

950 **G.W. Stewart**, Gaussian elimination, perturbation theory and Markov chains

951 **G.W. Stewart**, On a new way of solving the linear equations that arise in the method of least squares

952 **G.W. Stewart**, On the early history of the singular value decomposition

953 **G.W. STewart**, On the perturbation of Markov chains with nearly transient states

954 **Umberto Mosco**, Composite media and asymptotic dirichlet forms

955 **Walter F. Mascarenhas**, The structure of the eigenvectors of sparse matrices

956 **Walter F. Mascarenhas**, A note on Jacobi being more accurate than QR

957 **Raymond H. Chan, James G. Nagy and Robert J. Plemmons**, FFT-based preconditioners for Toeplitz-Block least squares problems

958 **Zhaojun Bai**, The CSD, GSVD, their applications and computations

959 **D.A. Gregory, S.J. Kirkland and N.J. Pullman**, A bound on the exponent of a primitive matrix using Boolean rank

960 **Richard A. Brualdi, Shmuel Friedland and Alex Pothen**, Sparse bases, elementary vectors and nonzero minors of compound matrices

961 **J.W. Demmel**, Open problems in numerical linear algebra

962 **James W. Demmel and William Gragg**, On computing accurate singular values and eigenvalues of acyclic matrices

963 **James W. Demmel**, The inherent inaccuracy of implicit tridiagonal QR

964 **J.J.L. Velázquez**, Estimates on the $(N - 1)$-dimensional Hausdorff measure of the blow-up set for a semilinear heat equation