SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 580–601
© 2000 Society for Industrial and Applied Mathematics

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ON THE COMPUTATION OF THE RESTRICTED SINGULAR VALUE DECOMPOSITION VIA THE COSINE-SINE DECOMPOSITION*

DELIN CHU†, LIEVEN DE LATHAUWER‡, AND BART DE MOOR‡

Abstract. In this paper, we show that the restricted singular value decomposition of a matrix triplet $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$ can be computed by means of the cosine-sine decomposition. In the first step, the matrices A, B, C are reduced to a lower-dimensional matrix triplet $\mathcal{A}, \mathcal{B}, \mathcal{C}$, in which $\mathcal{B}$ and $\mathcal{C}$ are nonsingular, using orthogonal transformations such as the QR-factorization with column pivoting and the URV decomposition. In the second step, the components of the restricted singular value decomposition of A, B, C are derived from the singular value decomposition of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. Instead of explicitly forming the latter product, a link with the cosine-sine decomposition, which can be computed by Van Loan's method, is exploited. Some numerical examples are given to show the performance of the presented method.

Key words. singular value decomposition, restricted singular value decomposition, cosine-sine decomposition, QR-factorization, URV decomposition

AMS subject classifications. 65F15, 65H15

PII. S0895479898346983

1. Introduction.

In past years, several generalizations of the singular value decomposition (SVD), related to a sequence of matrices in product/quotient form, have been proposed and their properties analyzed. These generalized SVDs (GSVDs) are essential numerical linear algebraic tools in signal processing and identification. Possible applications include source separation, stochastic realization, generalized Gauss–Markov estimation problems, generalized total linear least squares problems, open and closed loop balancing, etc. [7, 6].

Most well known are the product SVD (PSVD) (discussed in [11, 13]), the quotient SVD (QSVD) (introduced in [17] and refined in [14, 21]), and the restricted SVD (RSVD) (introduced in its explicit form in [19] and further developed and discussed in [4, 5]). The PSVD of a pair of matrices A, B is related to the SVD of A^TB, the QSVD of a pair of matrices A, C corresponds to the SVD of AC^{-1} if C is nonsingular, and the RSVD of a matrix triplet A, B, C shows the SVD of $B^{-1}AC^{-1}$ if B and C are nonsingular.

*Received by the editors November 6, 1998; accepted for publication (in revised form) by L. Eldén May 30, 2000; published electronically September 7, 2000. This research was partially supported by the Flemish Government through (1) the Research Council K.U. Leuven: Concerted Research Actions GOA-MIPS (Model-Based Information Processing Systems) and GOA-MEFISTO-666 (Mathematical Engineering for Information and Communication Systems Technology), (2) the Fund for Scientific Research–Flanders (F.W.O.) project G.0256.97 (Numerical Algorithms for Subspace System Identification, Extension to Special Cases), (3) the F.W.O. Research Communities ICCoS (Identification and Control of Complex Systems) and ANMMM (Advanced Numerical Methods for Mathematical Modelling); and by the Belgian State, Prime Minister's Office—Federal Office for Scientific, Technical and Cultural Affairs through the Interuniversity Poles of Attraction Programmes IUAP P4-02 (Modeling, Identification, Simulation and Control of Complex Systems) and IUAP P4-24 (Intelligent Mechatronic Systems (IMechS)). The scientific responsibility is assumed by the authors.

http://www.siam.org/journals/simax/22-2/34698.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matchudl@math.nus.edu.sg). This author was a visiting researcher with the K.U. Leuven during part of this research.

‡Department of Electrical Engineering, Research Group SISTA, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium (lieven.delathauwer@esat.kuleuven.ac.be, bart.demoor@esat.kuleuven.ac.be)

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

As far as the GSVDs related to a matrix inverse is concerned, the calculation of the QSVD has been extensively studied. A key idea, which is also exploited in this paper, is the link between the cosine-sine decomposition (CSD) [12] of a partitioned column-orthogonal matrix $\left[\begin{smallmatrix} A \\ C \end{smallmatrix}\right]$ and the QSVD of the couple (A, C): if the CSD of a column-orthogonal matrix $\left[\begin{smallmatrix} A \\ C \end{smallmatrix}\right]$, with $A \in \mathbf{R}^{n\times m}$ and $C \in \mathbf{R}^{p\times m}$, where we assume that $n \geq m$, is given by

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I_{n-p} & 0 \\ 0 & C \\ 0 & -S \end{bmatrix} W,$$

in which $U, W \in \mathbf{R}^{n\times n}$ and $V \in \mathbf{R}^{p\times p}$ are orthogonal and

$$\begin{aligned} C &= \operatorname{diag}(c_1, \ldots, c_p) \in \mathbf{R}^{p\times p}, & c_i &= \cos\theta_i, \\ S &= \operatorname{diag}(s_1, \ldots, s_p) \in \mathbf{R}^{p\times p}, & s_i &= \sin\theta_i, \end{aligned}$$

with $0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_j \leq \pi/2$, then we have that the SVD of $AC^\dagger$ is given by

$$AC^\dagger = -UCS^\dagger W^T.$$

(An analogous link can be derived for the case $n \leq m$.) This idea forms the basis for two backward stable algorithms proposed by Stewart [16] and Van Loan [18]. Recently, Zha has studied the computation of the QSVD via the CSD and the Lanczos bidiagonalization process. The Kogbetliantz algorithm has been generalized for the computation of the QSVD by Paige [15]; his elegant implementation initially transforms A and C into a pair of triangular matrices and preserves the triangular form in the iterative phase by using suitable plane rotations. A variation of Paige's algorithm, also backward stable, is given by Bai and Demmel [2]. This technique is implemented as the LAPACK procedure STGSJA() [1]. High relative accuracy Jacobi-type algorithms for the PSVD and the QSVD are developed in Drmač [8, 9, 10].

For the computation of the RSVD, the Kogbetliantz algorithm has been generalized by Zha [20]. However, it is well known that Jacobi-type algorithms, which use Jacobi transformations, typically have a higher complexity than QR-type algorithms, which are based on QR-factorizations [3, 12, 15]. So far, nobody has proposed a QR-type algorithm for the RSVD based on the CSD. In this paper, we show that such an algorithm can in fact be derived in an easy way.

The paper is organized as follows. In section 2 the RSVD theorem is recalled. In section 3 we present our numerical method for computing the RSVD. Some numerical examples are given in section 4. Conclusions are drawn in section 5.

2. The RSVD theorem. The RSVD is a simultaneous decomposition of three matrices with compatible dimensions to quasi-diagonal forms, i.e., to matrices consisting of diagonal and zero blocks with at most one nonvanishing diagonal block per block row and column. In this section we briefly recall its definition and importance.

Given matrices A, B, C with compatible size, we denote

$$\begin{aligned} &r_a = \operatorname{rank}(A), \quad r_b = \operatorname{rank}(B), \quad r_c = \operatorname{rank}(C), \\ &r_{ab} = \operatorname{rank}\begin{bmatrix} A & B \end{bmatrix}, \quad r_{ac} = \operatorname{rank}\begin{bmatrix} A \\ C \end{bmatrix}, \quad r_{abc} = \operatorname{rank}\begin{bmatrix} A & B \\ C & 0 \end{bmatrix}, \\ &k_1 = r_{abc} - r_b - r_c, \quad k_2 = r_{ac} + r_b - r_{abc}, \quad k_3 = r_{ab} + r_c - r_{abc}, \\ &\mu = r_{abc} - r_{ab}, \quad \nu = r_{abc} - r_{ac}, \quad \eta = r_{abc} + r_a - r_{ab} - r_{ac}. \end{aligned}$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The RSVD is now described in the following theorem.

THEOREM 2.1 (RSVD theorem). *Given $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$, there exist nonsingular matrices $X \in \mathbf{R}^{n\times n}, Y \in \mathbf{R}^{m\times m}$ and orthogonal matrices $U \in \mathbf{R}^{l\times l}, V \in \mathbf{R}^{p\times p}$ such that*

$$XAY = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{array} \\ \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & S_A & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

$$XBU = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{array} \begin{array}{c} \begin{array}{ccc} l-r_b & k_2 & \nu \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \tag{2.1}$$

$$VCY = \begin{array}{c} \\ p-r_c \\ k_3 \\ \mu \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{bmatrix} \end{array},$$

in which $S_A = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}$, with $\Sigma \in \mathbf{R}^{\eta\times\eta}$ positive diagonal.

Equation (2.1) can be interpreted as a decomposition of A, relative to the column space of B and the row space of C. With an appropriate partitioning of X^{-1} and Y^{-1} it can be written as

$$A = (X^{-1})_1(Y^{-1})_1 + (X^{-1})_2(Y^{-1})_2 + (X^{-1})_3(Y^{-1})_3 + (X^{-1})_4\Sigma(Y^{-1})_4,$$

in which the four terms can geometrically be classified as follows:

	In column space B	Not in column space B
In row space C	$(X^{-1})_4\Sigma(Y^{-1})_4$	$(X^{-1})_3(Y^{-1})_3$
Not in row space C	$(X^{-1})_2(Y^{-1})_2$	$(X^{-1})_1(Y^{-1})_1$

Obviously, the term $(X^{-1})_4\Sigma(Y^{-1})_4$ represents the *restriction* of the linear operator represented by the matrix A to the column space of B and the row space of C; the term $(X^{-1})_1(Y^{-1})_1$ is the restriction of A to the orthogonal complements of these spaces, and so on.

The decomposition can also be seen as an ordinary SVD with different inner products in row and column space. Consider the maximization of the bilinear form $\phi(x, y) = x^T Ay$ over all vectors x, y subject to $x^T BB^T x = 1$ and $y^T C^T Cy = 1$. Assuming $\eta \neq 0$, one can show that the maximum is equal to the largest diagonal element of Σ and the optimizing vectors are $x = x_1$ (first column vector of X) and $y = y_1$ (first column vector of Y). ($\eta = 0$ is a special case, in which either the maximum is 0 or the norm constraints cannot be satisfied.) Other extrema of the objective function, constrained to lie in subspaces that are orthogonal to x_1 with respect to the inner product defined by BB^T and orthogonal to y_1 with respect to the

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

inner product defined by C^TC, can be found in an obvious recursive manner. These extrema correspond to the columns of X and Y.

Considering the fact that knowledge of the column space of B and the row space of C is present in (2.1), it comes as no surprise that the RSVD plays a crucial role in a basic problem such as the analysis of the rank of $A + BDC$ for varying $D \in \mathbf{R}^{l\times p}$; also the influence of changing D on the rank of

$$M = \begin{bmatrix} A & B \\ C & D^T \end{bmatrix}$$

can be examined by means of the RSVD. Another application is the minimization of $\|y\|^2 + \|x\|^2$ over all vectors x, y, z satisfying

$$b = Ax + By, \quad z = Cx,$$

where A, B, C, b are given. In this variant of the conventional linear least squares problem the matrix B allows us to take into account the geometrical distribution of the noise. For example, the error term By is restricted to the column space of B; other components of the observation vector b are considered as error-free. The matrix C represents a weighting of the components of x, e.g., due to a priori information that some components are more likely or less costly than others.

The RSVD was introduced in [19]. A discussion of its properties and applications can be found in [5].

3. Computation of the RSVD via the CSD. [19] contained a constructive proof of Theorem 2.1. However, as pointed out in [19], this procedure is not useful from a numerical point of view, since in this constructive proof nonorthogonal transformations are used to scale nonsingular matrices to identity matrices and to eliminate certain submatrices in some *intermediate steps* during the decomposition. This will cause numerical instability if the underlying matrices or the pivoting matrices are ill-conditioned. In [20], Zha generalized the Kogbetliantz algorithm to compute the RSVD. In this section we show that the RSVD can be computed via the CSD.

Our method consists of three different stages. Given an arbitrary matrix triplet A, B, C, the first stage is a preprocessing QR reduction step which produces a possibly lower-dimensional triplet $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ with $\mathcal{B}$ and $\mathcal{C}$ nonsingular (section 3.1). This preprocessing indicates very clearly how we can get the nonsingular matrices X and Y in the RSVD (2.1). The second stage is the actual calculation of the SVD of matrix $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ via the CSD (section 3.2). The third stage consists of the backtransformation of the results to the original matrix spaces (section 3.3). An outline of the overall algorithm is presented in section 3.4.

3.1. Preprocessing reduction. Before we give our preprocessing reduction for the RSVD, we need to recall the QR-factorization with column pivoting and URV decomposition [12], which will be the building blocks of our constructive proof of Lemmas 3.1 and 3.2 below.

It is well known that any matrix $A \in \mathbf{R}^{m\times n}$ can be factorized as

$$UA = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix} \Pi, \tag{3.1}$$

where U and Π are orthogonal matrix and permutation matrix, respectively, R_1 is nonsingular and upper triangular. The factorization (3.1) is called the QR-factorization

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

of A with column pivoting. Note that we do not need memory to store the matrix Π during the computation of (3.1). Moreover, if we denote

$$R := \begin{bmatrix} R_1 & R_2 \end{bmatrix} \Pi,$$

then R is of full row rank and

$$UA = \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

If we continue to squeeze $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ into upper triangular form by applying a sequence of Householder transformations, then we have the following URV decomposition of A, i.e., we get an orthogonal matrix V such that

$$UAV = \begin{bmatrix} \mathcal{R} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.2}$$

with $\mathcal{R}$ nonsingular and upper triangular.

To compute the RSVD (2.1) of the triplet A, B, and C, first we reduce A, B, and C to lower dimensional submatrices $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ with $\mathcal{B}$ and $\mathcal{C}$ nonsingular, applying orthogonal transformations to compress the rows and/or columns of certain matrices in such a way that the RSVD of A, B, C can easily be derived from the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. The idea is that a compression of the rows and/or columns of the matrices that are involved allows for a more efficient formulation of the actual core computations.

Preprocessing reduction step is necessary for computing QSVD and RSVD; see [2, 3, 15, 20]. In all of the above-cited papers and in our paper, the QR-factorization with column pivoting and the URV decomposition are used to compress the rows and/or columns of a matrix [12]. (Of course, the SVD method can also be used to compress rows and/or columns of a matrix, but it is much more expensive.) The rank determination of a matrix is an ill-conditioned problem, which always arises in the computation of the QSVD and RSVD. The common and effective tools are QR-factorization with column pivoting, URV decomposition, and SVD.

We will now state two lemmas and one theorem on which our preprocessing step is based. These three results describe how the given matrices can be brought in a desired form by means of orthogonal transformations. Lemma 3.1 deals with two matrices; it is a tool to prove Lemma 3.2, which discusses the actual situation of three matrices. Theorem 3.4 is a refinement of Lemma 3.2.

LEMMA 3.1. *Given matrices $A \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{p\times m}$, there exist three orthogonal matrices $U \in \mathbf{R}^{n\times n}, W \in \mathbf{R}^{m\times m}, V \in \mathbf{R}^{p\times p}$ such that*

$$UAW = \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & 0 \end{bmatrix} \end{matrix},$$

$$VCW = \begin{matrix} \\ p-r_c \\ r_c \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_{22} & 0 \end{bmatrix} \end{matrix}, \tag{3.3}$$

where A_{11} and C_{22} are nonsingular.

Proof. We prove Lemma 3.1 constructively by the following algorithm.

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ALGORITHM 1.

Input: $A \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{p\times m}$.

Output: *Orthogonal matrices* $U \in \mathbf{R}^{n\times n}, V \in \mathbf{R}^{p\times p}, W \in \mathbf{R}^{m\times m}$ *and the form* (3.3).

Step 1. *Perform a URV decomposition of* C *to get orthogonal matrices* V *and* W *such that*

$$VCW =: \begin{matrix} \\ p-r_c \\ r_c \end{matrix} \begin{matrix} \begin{matrix} m-r_c & r_c \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & C_{22} \end{bmatrix} \end{matrix}$$

with C_{22} *nonsingular. Set*

$$AW =: \begin{matrix} \begin{matrix} m-r_c & r_c \end{matrix} \\ \begin{bmatrix} A_1 & A_2 \end{bmatrix} \end{matrix}.$$

Step 2. *Perform a URV decomposition of* A_1 *(note that* $\operatorname{rank}(A_1) = r_{ac} - r_c$*) to get orthogonal matrices* U *and* W_2 *such that*

$$UA_1W_2 =: \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & 0 \\ 0 & 0 \end{bmatrix} \end{matrix}$$

with A_{11} *nonsingular. Set*

$$UA_2 =: \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix}, \; W := W \begin{bmatrix} W_2 & & \\ & I_{r_c} & \\ & & \end{bmatrix} \begin{bmatrix} I & & \\ & 0 & I_{m-r_{ac}} \\ & I_{r_c} & \end{bmatrix}.$$

Step 3. *Output* U, V, W *and* UAW, VCW.

It is easy to see that (U^TAW, V^TCW) are in the form (3.3). ∎

Based on Lemma 3.1, we have the following lemma.

LEMMA 3.2. *Given* $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$, *there exist orthogonal matrices* $P \in \mathbf{R}^{n\times n}, Q \in \mathbf{R}^{m\times m}, U_b \in \mathbf{R}^{l\times l}, V_c \in \mathbf{R}^{p\times p}$ *such that*

$$PAQ = \begin{matrix} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & 0 \\ 0 & A_{22} & A_{23} & A_{24} & 0 \\ 0 & 0 & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix},$$

$$PBU_b = \begin{matrix} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} l-r_b & k_2 & \nu \end{matrix} \\ \begin{bmatrix} 0 & 0 & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & 0 \\ 0 & 0 & B_{43} \\ 0 & 0 & 0 \end{bmatrix} \end{matrix}, \tag{3.4}$$

$$V_cCQ = \begin{matrix} \\ p-r_c \\ k_3 \\ \mu \end{matrix} \begin{matrix} \begin{matrix} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{23} & 0 & 0 \\ 0 & 0 & C_{33} & C_{34} & 0 \end{bmatrix} \end{matrix},$$

where A_{11}, A_{22}, A_{33}, B_{22}, B_{43}, C_{23}, *and* C_{34} *are nonsingular.*

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. Similarly to the proof of Lemma 3.1, we prove Lemma 3.2 constructively by the following algorithm.

ALGORITHM 2.

Input: $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$.

Output: *Orthogonal matrices* $P \in \mathbf{R}^{n\times n}, Q \in \mathbf{R}^{m\times m}, U_b \in \mathbf{R}^{l\times l}, V_c \in \mathbf{R}^{p\times p}$ *and the form* (3.4).

Step 1. *Apply Algorithm* 1 *to* (A, C) *to get orthogonal matrices* P, V_c, *and* Q *such that*

$$PAQ =: \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & A_{13} & 0 \\ 0 & A_{33} & 0 \end{bmatrix} \end{matrix},$$

$$V_cCQ =: \begin{matrix} \\ p-r_c \\ r_c \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_{23} & \end{bmatrix} \end{matrix},$$

where A_{11} *and* C_{23} *are nonsingular. Denote*

$$PB =: \begin{matrix} r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{bmatrix} B_1 \\ B_3 \end{bmatrix}.$$

Then we have that $\operatorname{rank}\begin{bmatrix} A_{33} & B_3 \end{bmatrix} = r_{ab}+r_c-r_{ac} =: t$.

Step 2. *Apply Algorithm* 1 *to* (A_{33}^T, B_3^T) *to get orthogonal matrices* U_b, P_2, *and* Q_2 *such that*

$$P_2A_{33}Q_2 =: \begin{matrix} \\ t-x \\ x \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} t-x & r_c+x-t \end{matrix} \\ \begin{bmatrix} A_{33} & 0 \\ A_{43} & A_{44} \\ 0 & 0 \end{bmatrix} \end{matrix}, \quad P_2B_3U_b =: \begin{matrix} \\ t-x \\ x \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} l-x & x \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & B_{43} \\ 0 & 0 \end{bmatrix} \end{matrix}$$

with A_{33} *and* B_{43} *nonsingular. Set*

$$B_1U_b =: \begin{matrix} \begin{matrix} l-x & x \end{matrix} \\ \begin{bmatrix} B_{11} & B_{13} \end{bmatrix} \end{matrix}, \quad A_{13}Q_2 =: \begin{matrix} \begin{matrix} t-x & r_c+x-t \end{matrix} \\ \begin{bmatrix} A_{13} & A_{14} \end{bmatrix} \end{matrix}, \quad C_{23}Q_2 =: \begin{matrix} \begin{matrix} t-x & x \end{matrix} \\ \begin{bmatrix} C_{23} & C_{24} \end{bmatrix} \end{matrix},$$

$$P := \begin{bmatrix} I_{r_{ac}-r_c} & \\ & P_2 \end{bmatrix} P, \quad Q := Q \begin{bmatrix} I_{r_{ac}-r_c} & & \\ & Q_2 & \\ & & I_{m-r_{ac}} \end{bmatrix}.$$

We have that $\begin{bmatrix} C_{23} & C_{24} \end{bmatrix}$ *is nonsingular.*

Step 3. *Perform a QR-factorization of* C_{24} *with column pivoting to get orthogonal matrix* V_3 *such that*

$$V_3C_{24} =: \begin{matrix} t-x \\ r_c+x-t \end{matrix} \begin{bmatrix} 0 \\ C_{34} \end{bmatrix}$$

with C_{34} *nonsingular. Set*

$$V_3C_{23} =: \begin{matrix} t-x \\ r_c+x-t \end{matrix} \begin{bmatrix} C_{23} \\ C_{33} \end{bmatrix}, \quad V_c := \begin{bmatrix} I_{p-r_c} & \\ & V_3 \end{bmatrix} V_c.$$

Here, C_{23} *is nonsingular.*

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Step 4. *Perform a URV decomposition of* B_{11} *(note that* $\operatorname{rank}(B_{11}) = r_b - x$*) to get orthogonal matrices* P_4 *and* U_4 *such that*

$$P_4 B_{11} U_4 =: \begin{matrix} \\ r_{ac}+x-r_b-r_c \\ r_b - x \end{matrix} \begin{matrix} \begin{matrix} l-r_b & r_b-x \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & B_{22} \end{bmatrix} \end{matrix}$$

with B_{22} *nonsingular. Set*

$$P_4 B_{13} =: \begin{matrix} r_{ac}+x-r_b-r_c \\ r_b - x \end{matrix} \begin{bmatrix} B_{13} \\ B_{23} \end{bmatrix},$$

$$P_4 \begin{bmatrix} A_{11} & A_{13} & A_{14} \end{bmatrix} =: \begin{matrix} \\ r_{ac}+x-r_b-r_c \\ r_b - x \end{matrix} \begin{matrix} \begin{matrix} r_{ac}+x-r_b-r_c & r_b-x & t-x & r_c+x-t \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \end{bmatrix} \end{matrix},$$

$$P := \begin{bmatrix} P_4 & \\ & I_{n+r_c-r_{ac}} \end{bmatrix}, \quad U_b := U_b \begin{bmatrix} U_4 & \\ & I_x \end{bmatrix}.$$

We know that $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ *is nonsingular.*

Step 5. *Perform a QR-factorization of* $\begin{bmatrix} A_{21} & A_{22} \end{bmatrix}^T$ *with column pivoting to get orthogonal matrix* Q_5 *such that*

$$\begin{bmatrix} A_{21} & A_{22} \end{bmatrix} Q_5 =: \begin{matrix} \begin{matrix} r_{ac}+x-r_b-r_c & r_b-x \end{matrix} \\ \begin{bmatrix} 0 & A_{22} \end{bmatrix} \end{matrix}$$

with A_{22} *nonsingular. Set*

$$\begin{bmatrix} A_{11} & A_{12} \end{bmatrix} Q_5 =: \begin{matrix} \begin{matrix} r_{ac}+x-r_b-r_c & r_b-x \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \end{matrix}, \quad Q := Q \begin{bmatrix} Q_5 & \\ & I_{m+r_c-r_{ac}} \end{bmatrix}.$$

It is easy to see that A_{11} *is nonsingular.*

Step 6. *Output* $P, Q, U_b, V_c, PAQ, PBU_b$*, and* $V_c CQ$*.*

A simple calculation yields that $r_{abc} = r_{ac} + x$, i.e., $x = r_{abc} - r_{ac}$. Hence, matrices PAQ, PBU_b, and $V_c CQ$ are in the form (3.4). ∎

Algorithm 2 is implemented using only orthogonal transformations; hence, it is numerically stable.

The reduction procedure above is similar to the one proposed in Zha [20], but it is more convenient for the computation of the RSVD, because the condensed form (3.4) gives an explicit way to get the nonsingular matrix X and Y in the RSVD (2.1). Note that in Algorithm 2

$$P_2 \in \mathbf{R}^{(n+r_c-r_{ac})\times(n+r_c-r_{ac})}, \quad Q_2, V_3 \in \mathbf{R}^{r_c \times r_c},$$
$$U_4 \in \mathbf{R}^{(l+r_{ac}-r_{abc})\times(l+r_{ac}-r_{abc})}, \quad P_4, Q_5 \in \mathbf{R}^{(r_{ac}-r_c)\times(r_{ac}-r_c)},$$
$$n + r_c - r_{ac} \le n, \quad r_{ac} - r_c \le r_a \le n, \quad r_c \le m,\ l + r_{ac} - r_{abc} \le l,$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

so, besides the necessary arrays for matrices A, B, C, P, Q, U_b, and V_c, two extra arrays of at most $n \times n$ and $\max(l,m) \times \max(l,m)$ orders are enough for the temporal storage of the "temporal" matrices Q_2, P_2, V_3, P_4, U_4, and Q_5. So, the memory requirement of Algorithm 2 is the same as that in the reduction procedure of [20]; hence, it is acceptable.

In order to get the nonsingular matrices X and Y in the RSVD (2.1) we have to eliminate submatrices A_{12}, A_{13}, A_{14}, A_{23}, A_{24}, A_{43}, B_{13}, B_{23}, and C_{33} in (3.4). Similar to Zha [20], these submatrices can be eliminated using nonsingular matrices A_{11}, A_{22}, A_{33}, B_{43}, and C_{34} by solving some coupled linear matrix equations as follows:

$$X_1 B_{43} = \begin{bmatrix} B_{13} \\ B_{23} \end{bmatrix}, \tag{3.5}$$

$$A_{11} Y_1 = A_{12}, \ \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} Y_2 = \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix} - X_1 \begin{bmatrix} A_{43} & A_{44} \end{bmatrix}, \tag{3.6}$$

$$C_{34} Y_3 = C_{33}, \ X_2 A_{33} = A_{43} - A_{44} Y_3. \tag{3.7}$$

However, if some of matrices A_{11}, A_{22}, A_{33}, B_{43}, and C_{34} are highly ill-conditioned, the numerical computations of linear matrix equations (3.5), (3.6), and (3.7) will encounter numerical difficulties and lead to large computational errors. Fortunately, we will show that we can refine (3.4) and eliminate submatrices A_{12}, A_{13}, A_{14}, A_{23}, A_{24}, B_{13}, B_{23}, and C_{33} using only orthogonal transformations such as QR-factorization with column pivoting.

LEMMA 3.3. *Given matrices*

$$\mathcal{A}_{11} \in \mathbf{R}^{n_1 \times n_1}, \quad \mathcal{A}_{12} \in \mathbf{R}^{n_1 \times n_2}, \quad \mathcal{A}_{21} \in \mathbf{R}^{\tilde{n}_2 \times n_1}, \quad \mathcal{A}_{22} \in \mathbf{R}^{\tilde{n}_2 \times n_2}$$

with $\mathcal{A}_{11}$ nonsingular, let $\mathcal{P}_{21} \in \mathbf{R}^{\tilde{n}_2 \times n_1}$ and $\mathcal{P}_{22} \in \mathbf{R}^{\tilde{n}_2 \times \tilde{n}_2}$ satisfy

$$\begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}^T = I_{\tilde{n}_2}.$$

Denote

$$\begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \tilde{\mathcal{A}}_{22}.$$

Then $\mathcal{P}_{22}$ is nonsingular,

$$\begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{12} \\ \tilde{\mathcal{A}}_{22} \end{bmatrix}, \tag{3.8}$$

and, furthermore,

$$\mathcal{A}_{22} - \mathcal{A}_{21} \mathcal{A}_{11}^{-1} \mathcal{A}_{12} = (\mathcal{P}_{22}^T - \mathcal{P}_{12}^T \mathcal{P}_{11}^{-T} \mathcal{P}_{21}^T) \tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}^{-1} \tilde{\mathcal{A}}_{22}. \tag{3.9}$$

Consequently, if $\mathcal{A}_{12} = 0$, $\tilde{n}_2 = n_2$, and $\mathcal{A}_{22}$ is nonsingular, then $\tilde{\mathcal{A}}_{22}$ is also nonsingular.

Proof. Equation (3.8) is obvious. Let $\mathcal{P} = \begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}$ be orthogonal and denote

$$\begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \tilde{\mathcal{A}}_{12}.$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Then we have

$$\mathcal{A}_{11} = P_{11}^T \tilde{\mathcal{A}}_{11}.$$

Hence, $\mathcal{P}_{11}$ and $\tilde{\mathcal{A}}_{11}$ are nonsingular. Note that $\mathcal{P}$ is orthogonal, so $\mathcal{P}_{22}$ is also nonsingular. Moreover,

$$\mathcal{P}_{21}\mathcal{P}_{11}^T + \mathcal{P}_{22}\mathcal{P}_{12}^T = 0, \quad \mathcal{P}_{21}\mathcal{P}_{21}^T + \mathcal{P}_{22}\mathcal{P}_{22}^T = I,$$

which, with the nonsingularity of $\mathcal{P}_{11}$ and $\mathcal{P}_{22}$, gives

$$\begin{aligned} \mathcal{P}_{22}^T - \mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T &= \mathcal{P}_{22}^{-1}(\mathcal{P}_{22}\mathcal{P}_{22}^T - \mathcal{P}_{22}\mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T) \\ &= \mathcal{P}_{22}^{-1}(\mathcal{P}_{22}\mathcal{P}_{22}^T + \mathcal{P}_{21}\mathcal{P}_{21}^T) \\ &= \mathcal{P}_{22}^{-1}. \end{aligned} \tag{3.10}$$

On other hand, we also have

$$\mathcal{A}_{21} = \mathcal{P}_{12}^T\tilde{\mathcal{A}}_{11}, \quad \mathcal{A}_{22} = \mathcal{P}_{12}^T\tilde{\mathcal{A}}_{12} + \mathcal{P}_{22}^T\tilde{\mathcal{A}}_{22}, \quad \mathcal{A}_{12} = \mathcal{P}_{11}^T\tilde{\mathcal{A}}_{12} + \mathcal{P}_{21}^T\tilde{\mathcal{A}}_{22}.$$

Hence, we have

$$\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12} = (\mathcal{P}_{22}^T - \mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T)\tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}^{-1}\tilde{\mathcal{A}}_{22},$$

i.e., (3.9) is true. Moreover, if $\mathcal{A}_{12} = 0, \tilde{n}_2 = n_2$ and $\mathcal{A}_{22}$ is nonsingular, then, $\tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}\mathcal{A}_{22}$ is nonsingular. □

In general, for matrix $\mathcal{A}_{ij}(i,j = 1,2)$ with suitable sizes, if $\mathcal{A}_{11}$ is nonsingular but very ill-conditioned, the computation of $\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12}$ will not be numerically stable. Fortunately, (3.9) in the proof of Lemma 3.3 gives that we can cancel "the instability factor" of $\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12}$ by multiplying $\mathcal{P}_{22}$. Now, although $\begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}$ is not orthogonal, it and the form (3.8) are computed by only orthogonal transformations, for example, the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix}$ with column pivoting.

If we apply Lemma 3.3 to the form (3.4), we have the following result.

THEOREM 3.4. *Given $A \in \mathbf{R}^{n\times m}$, $B \in \mathbf{R}^{n\times l}$, and $C \in \mathbf{R}^{p\times m}$, there exist nonsingular matrices $\mathcal{X} \in \mathbf{R}^{n\times n}$, $\mathcal{Y} \in \mathbf{R}^{m\times m}$ and orthogonal matrices $U_b \in \mathbf{R}^{l\times l}$ and $V_c \in \mathbf{R}^{p\times p}$ such that*

$$\mathcal{X}A\mathcal{Y} = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n - r_{ab} \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m - r_{ac} \end{array} \\ \begin{bmatrix} \mathcal{A}_{11} & 0 & 0 & 0 & 0 \\ 0 & \mathcal{A}_{22} & 0 & 0 & 0 \\ 0 & 0 & \mathcal{A}_{33} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{A} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

$$\mathcal{X}BU_b = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n - r_{ab} \end{array} \begin{array}{c} \begin{array}{ccc} l - r_b & k_2 & \nu \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathcal{B}_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathcal{B} \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \tag{3.11}$$

$$V_cC\mathcal{Y} = \begin{array}{c} \\ p - r_c \\ k_3 \\ \mu \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m - r_{ac} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{C}_{23} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{C} & 0 \end{bmatrix} \end{array},$$

where $\mathcal{A}_{11}, \mathcal{A}_{22}, \mathcal{A}_{33}$, $\mathcal{B}_{22}$, $\mathcal{C}_{23}$, $\mathcal{B}$, *and* $\mathcal{C}$ *are all nonsingular. Moreover,* $\mathcal{X}$, $\mathcal{Y}$, *and the condensed form* (3.11) *are computed using only orthogonal transformations which are numerically stable.*

Proof. We prove Theorem 3.4 constructively by the following algorithm.

ALGORITHM 3.

Input: $A \in \mathbf{R}^{n\times m}$, $B \in \mathbf{R}^{n\times l}$, *and* $C \in \mathbf{R}^{p\times m}$.

Output: *Nonsingular matrices* $\mathcal{X} \in \mathbf{R}^{n\times n}$ *and* $\mathcal{Y} \in \mathbf{R}^{m\times m}$, *orthogonal matrices* $U_b \in \mathbf{R}^{l\times l}$ *and* $V_c \in \mathbf{R}^{p\times p}$, *and the form* (3.11).

Step 1. *Perform Algorithm* 2 *to get orthogonal matrices* P, Q, U_b, *and* V_c *and the form* (3.4). *Set*

$$\mathcal{X} := P, \quad \mathcal{Y} =: Q.$$

Step 2. *Perform the QR-factorization of* $\left[\begin{smallmatrix} B_{13} \\ B_{43} \end{smallmatrix}\right]$ *with column pivoting to get* $\mathcal{U}_{11}^{(2)} \in \mathbf{R}^{k_1\times k_1}$ *and* $\mathcal{U}_{12}^{(2)} \in \mathbf{R}^{k_1\times\nu}$ *such that*

$$\begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \begin{bmatrix} B_{13} \\ B_{43} \end{bmatrix} =: 0, \quad \begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix}^T = I_{k_1}.$$

Set

$$\mathcal{A}_{11} := \mathcal{U}_{11}^{(2)} A_{11}, \quad A_{12} := \mathcal{U}_{11}^{(2)} A_{12}, \qquad \begin{matrix} k_3 & \mu \end{matrix}$$
$$\begin{bmatrix} A_{13} & A_{14} \end{bmatrix} := \begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \begin{bmatrix} A_{13} & A_{14} \\ A_{43} & A_{44} \end{bmatrix},$$

and

$$\mathcal{X} := \begin{bmatrix} \mathcal{U}_{11}^{(2)} & 0 & \mathcal{U}_{12}^{(2)} & 0 \\ 0 & I_{k_2+k_3} & 0 & 0 \\ 0 & 0 & I_\nu & 0 \\ 0 & 0 & 0 & I_{n-r_{ab}} \end{bmatrix} \mathcal{X}.$$

Since A_{11} *is nonsingular, Lemma* 3.3 *gives that* $\mathcal{A}_{11}$ *is also nonsingular.*

Step 3. *Perform the QR-factorization of* $\left[\begin{smallmatrix} B_{23} \\ B_{43} \end{smallmatrix}\right]$ *with column pivoting to get* $\mathcal{U}_{11}^{(3)} \in \mathbf{R}^{k_2\times k_2}$ *and* $\mathcal{U}_{12}^{(3)} \in \mathbf{R}^{k_2\times\nu}$ *such that*

$$\begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \begin{bmatrix} B_{23} \\ B_{43} \end{bmatrix} =: 0, \quad \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix}^T = I_{k_2}.$$

Set

$$\begin{matrix} k_3 & \mu \end{matrix}$$
$$\begin{bmatrix} A_{23} & A_{24} \end{bmatrix} := \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \begin{bmatrix} A_{23} & A_{24} \\ A_{43} & A_{44} \end{bmatrix}, \quad \tilde{\mathcal{A}}_{22} := \mathcal{U}_{11}^{(3)} A_{22}, \quad \mathcal{B}_{22} := \mathcal{U}_{11}^{(3)} B_{22},$$

and

$$\mathcal{X} := \begin{bmatrix} I_{k_1} & 0 & 0 & 0 & 0 \\ 0 & \mathcal{U}_{11}^{(3)} & 0 & \mathcal{U}_{12}^{(3)} & 0 \\ 0 & 0 & I_{k_3} & 0 & 0 \\ 0 & 0 & 0 & I_\nu & 0 \\ 0 & 0 & 0 & 0 & I_{n-r_{ab}} \end{bmatrix} \mathcal{X}.$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Since A_{22} and B_{22} are nonsingular, Lemma 3.3 gives that $\tilde{\mathcal{A}}_{22}$ and $\mathcal{B}_{22}$ are also nonsingular.

Step 4. Perform the QR-factorization of $\begin{bmatrix} C_{33} & C_{34} \end{bmatrix}^T$ *with column pivoting to get* $\mathcal{V}_{11}^{(4)} \in \mathbf{R}^{k_3 \times k_3}$ *and* $\mathcal{V}_{21}^{(4)} \in \mathbf{R}^{\mu \times k_3}$ *such that*

$$\begin{bmatrix} C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix}^T \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix} = I_{k_3}.$$

Denote

$$\begin{array}{c} \\ k_1 \\ k_2 \\ \nu \end{array} \begin{array}{c} \begin{array}{cc} k_3 & \mu \end{array} \\ \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \\ A_{43} & A_{44} \end{bmatrix} \end{array} := \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix}, \quad \tilde{\mathcal{A}}_{33} := A_{33}\mathcal{V}_{11}^{(4)}, \quad \tilde{\mathcal{C}}_{23} := C_{23}\mathcal{V}_{11}^{(4)},$$

and

$$\mathcal{Y} := \mathcal{Y} \begin{bmatrix} I_{k_1+k_2} & 0 & 0 & 0 \\ 0 & \mathcal{V}_{11}^{(4)} & 0 & 0 \\ 0 & \mathcal{V}_{21}^{(4)} & I_\mu & 0 \\ 0 & 0 & 0 & I_{m-r_{ac}} \end{bmatrix}.$$

A_{33} and C_{23} are nonsingular, so we have from Lemma 3.3 that $\tilde{\mathcal{A}}_{33}$ and $\tilde{\mathcal{C}}_{23}$ are nonsingular.

Step 5. Perform the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{14} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{24} \end{bmatrix}^T$ *with column pivoting to get* $\mathcal{V}_{12}^{(5)} \in \mathbf{R}^{(k_1+k_2)\times\mu}$ *and* $\mathcal{V}_{22}^{(5)} \in \mathbf{R}^{\mu\times\mu}$ *such that*

$$\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{14} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{24} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{12}^{(5)} \\ \mathcal{V}_{22}^{(5)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{12}^{(5)} \\ \mathcal{V}_{22}^{(5)} \end{bmatrix}^T \begin{bmatrix} \mathcal{V}_{12}^{(5)} \\ \mathcal{V}_{22}^{(5)} \end{bmatrix} = I_\mu.$$

Denote

$$\tilde{\mathcal{A}}_{44} := A_{44}\mathcal{V}_{22}^{(5)}, \quad \mathcal{C} := C_{34}\mathcal{V}_{22}^{(5)},$$

and

$$\mathcal{Y} := \mathcal{Y} \begin{bmatrix} I_{k_1+k_2} & 0 & \mathcal{V}_{12}^{(5)} & 0 \\ 0 & I_{k_3} & 0 & 0 \\ 0 & 0 & \mathcal{V}_{22}^{(5)} & 0 \\ 0 & 0 & 0 & I_{m-r_{ac}} \end{bmatrix}.$$

Since C_{34} is nonsingular, by Lemma 3.3, $\mathcal{C}$ is nonsingular.

Step 6. Perform the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{13} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{23} \end{bmatrix}^T$ *with column pivoting to get* $\mathcal{V}_{12}^{(6)} \in \mathbf{R}^{(k_1+k_2)\times k_3}$ *and* $\mathcal{V}_{22}^{(6)} \in \mathbf{R}^{k_3\times k_3}$ *such that*

$$\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{13} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{23} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix}^T \begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix} = I_{k_3}.$$

Denote

$$\mathcal{A}_{33} := \tilde{\mathcal{A}}_{33}\mathcal{V}_{22}^{(6)}, \quad A_{43} := A_{43}\mathcal{V}_{22}^{(6)}, \quad \mathcal{C}_{23} := \tilde{\mathcal{C}}_{23}\mathcal{V}_{22}^{(6)},$$

and

$$\mathcal{Y} := \mathcal{Y}\begin{bmatrix} I_{k_1+k_2} & \mathcal{V}_{12}^{(6)} & 0 \\ 0 & \mathcal{V}_{22}^{(6)} & 0 \\ 0 & 0 & I_{\mu+m-r_{ac}} \end{bmatrix}.$$

Since $\tilde{\mathcal{A}}_{33}$ and $\tilde{\mathcal{C}}_{23}$ are nonsingular, by Lemma 3.3, *$\mathcal{A}_{33}$ and $\mathcal{C}_{23}$ are nonsingular.*

Step 7. *Perform the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} & A_{12} \end{bmatrix}^T$ with column pivoting to get $\mathcal{V}_{12}^{(7)} \in \mathbf{R}^{k_1\times k_2}$ and $\mathcal{V}_{22}^{(7)}\mathbf{R}^{k_2\times k_2}$ such that*

$$\begin{bmatrix} \mathcal{A}_{11} & A_{12} \end{bmatrix}\begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix}^T\begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix} = I_{k_2}.$$

Denote

$$\mathcal{A}_{22} := \tilde{\mathcal{A}}_{22}\mathcal{V}_{22}^{(7)}, \quad \mathcal{Y} := \mathcal{Y}\begin{bmatrix} I_{k_1} & \mathcal{V}_{12}^{(7)} & 0 \\ 0 & \mathcal{V}_{22}^{(7)} & 0 \\ 0 & 0 & I_{k_3+\mu+m-r_{ac}} \end{bmatrix}.$$

Note that $\tilde{\mathcal{A}}_{22}$ is nonsingular, so Lemma 3.3 *implies that $\mathcal{A}_{22}$ is nonsingular.*

Step 8. *Perform the QR factorization of $\begin{bmatrix} \mathcal{A}_{33} \\ A_{43} \end{bmatrix}$ with column pivoting to get $\mathcal{U}_{21}^{(8)} \in \mathbf{R}^{\nu\times k_3}$ and $\mathcal{U}_{22}^{(8)} \in \mathbf{R}^{\nu\times\nu}$ such that*

$$\begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix}\begin{bmatrix} \mathcal{A}_{33} \\ A_{43} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix}\begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix}^T = I_\nu.$$

Denote

$$\mathcal{A} := \mathcal{U}_{22}^{(8)}\tilde{\mathcal{A}}_{44}, \quad \mathcal{B} := \mathcal{U}_{22}^{(8)}B_{43}, \quad \mathcal{X} := \begin{bmatrix} I_{k_1+k_2} & 0 & 0 & 0 \\ 0 & I_{k_3} & 0 & 0 \\ 0 & \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} & 0 \\ 0 & 0 & 0 & I_{n-r_{ab}} \end{bmatrix}.$$

Since B_{43} is nonsingular, so by Lemma 3.3, *$\mathcal{B}$ is nonsingular.*

Step 9. *Output $\mathcal{X}, \mathcal{Y}$, U_b, V_c, $\mathcal{X}A\mathcal{Y}$, $\mathcal{X}BU_b$, and $V_cC\mathcal{Y}$.*

A simple calculation using Lemma 3.3 yields that $(\mathcal{X}A\mathcal{Y}, \mathcal{X}BU_b, V_cC\mathcal{Y})$ are in the form (3.11). □

It should be pointed out that Algorithm 3 is also implemented using only orthogonal transformations, so it is numerically stable.

It is easy to see that

$$\begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \in \mathbf{R}^{k_1\times(k_1+\nu)}, \ \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \in \mathbf{R}^{k_2\times(k_2+\nu)},$$

$$\begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix} \in \mathbf{R}^{\nu\times(k_3+\nu)}, \tag{3.12}$$

$$\begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix} \in \mathbf{R}^{(k_3+\mu)\times k_3}, \ \begin{bmatrix} \mathcal{V}_{11}^{(5)} \\ \mathcal{V}_{21}^{(5)} \end{bmatrix} \in \mathbf{R}^{(k_1+k_2+\mu)\times\mu}, \tag{3.13}$$

$$\begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix} \in \mathbf{R}^{(k_1+k_2+k_3)\times k_3}, \ \begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix} \in \mathbf{R}^{(k_1+k_2)\times k_2}, \tag{3.14}$$

$$\begin{aligned} &\max(k_1, k_2, k_3, \nu, k_1+\nu, k_2+\nu, k_3+\nu, k_1+k_2, k_1+k_2+k_3) \le n, \\ &\max(k_3, \mu, k_3+\mu, k_1+k_2+\mu) \le m. \end{aligned} \tag{3.15}$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Thus, the arrays for P and Q in Algorithm 2 can be used for $\mathcal{X}$ and $\mathcal{Y}$ in Algorithm 3, and the two arrays for the "temporal" matrices Q_2, P_2, V_3, P_4, U_4, and Q_5 in Algorithm 2 can be used for the temporal storage of the "temporal" matrices in (3.12), (3.13), and (3.14) in Algorithm 3. Hence, the memory requirements for Algorithm 2 and Algorithm 3 are the same, i.e., when we refine the form (3.4) to get the refined form (3.11), we do not need extra memory.

Obviously, we can scale $\mathcal{A}_{11}, \mathcal{A}_{22}, \mathcal{A}_{33}$, $\mathcal{B}_{22}$, $\mathcal{B}$, $\mathcal{C}_{23}$, and $\mathcal{C}$ in (3.11) to identity matrices by SVD or QR-factorization. As with Zha [20], this step cannot be avoided since the complete RSVD is computed, just like the computation of the complete QSVD [15]. Therefore, now the main problem for the RSVD (3.3) is how to compute the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, which is the main work of the next subsection.

3.2. CSD stage. An obvious way to compute the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is first forming $\mathcal{Z} = \mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ explicitly and then computing the SVD of $\mathcal{Z}$. As is well known, this approach is not numerically stable and will lead to large numerical errors when $\mathcal{B}$ or $\mathcal{C}$ is ill-conditioned, so generally it is not reliable, and hence not recommended.

In the following we show how the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ can be computed without forming $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ explicitly by solving a CSD problem.

LEMMA 3.5. *Let $\mathcal{A} \in \mathbf{R}^{\nu\times\mu}, \mathcal{B} \in \mathbf{R}^{\nu\times\nu}, \mathcal{C} \in \mathbf{R}^{\mu\times\mu}$ with $\mathcal{B}$ and $\mathcal{C}$ nonsingular. Let $\mathcal{P}_1 \in \mathbf{R}^{\nu\times\nu}$, $\mathcal{P}_2 \in \mathbf{R}^{\mu\times\nu}$, $\tilde{\mathcal{Q}}_1 \in \mathbf{R}^{\nu\times\mu}$, $\tilde{\mathcal{Q}}_2 \in \mathbf{R}^{\mu\times\mu}$ satisfy*

$$\mathcal{P}_1^T\mathcal{P}_1 + \mathcal{P}_2^T\mathcal{P}_2 = I_\nu, \qquad \begin{bmatrix} \mathcal{P}_1^T & \mathcal{P}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix} = 0,$$

$$\tilde{\mathcal{Q}}_1^T\tilde{\mathcal{Q}}_1 + \tilde{\mathcal{Q}}_2^T\tilde{\mathcal{Q}}_2 = I_\mu, \qquad \begin{bmatrix} \tilde{\mathcal{Q}}_1^T & \tilde{\mathcal{Q}}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix} = 0;$$

then we have that

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = \tilde{\mathcal{Q}}_1\tilde{\mathcal{Q}}_2^{-1}.$$

Proof. Let $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2 \in \mathbf{R}^{\mu\times\mu}$ be such that $\begin{bmatrix} \tilde{\mathcal{P}}_1 & \mathcal{P}_1 \\ \tilde{\mathcal{P}}_2 & \mathcal{P}_2 \end{bmatrix}$ is an orthogonal matrix; then

$$\begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{P}}_1 & \mathcal{P}_1 \\ \tilde{\mathcal{P}}_2 & \mathcal{P}_2 \end{bmatrix} \begin{bmatrix} \mathcal{R}_1 \\ 0 \end{bmatrix}$$

for some matrix $\mathcal{R}_1 \in \mathbf{R}^{\mu\times\mu}$. As $\mathcal{C}$ is nonsingular, $\tilde{\mathcal{P}}_2$ and $\mathcal{R}_1$ are nonsingular as well. Since

$$\begin{bmatrix} \mathcal{P}_1^T & \mathcal{P}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = 0,$$

we have that

$$\begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{P}}_1 & \mathcal{P}_1 \\ \tilde{\mathcal{P}}_2 & \mathcal{P}_2 \end{bmatrix} \begin{bmatrix} \mathcal{R}_2 \\ 0 \end{bmatrix} \quad \text{for some matrix } \mathcal{R}_2 \in \mathbf{R}^{\mu\times\mu}.$$

Because $\mathcal{B}$ is nonsingular, thus

$$\operatorname{rank} \begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \mu.$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Consequently, $\mathcal{R}_2$ is nonsingular, which, along with the nonsingularity of $\tilde{\mathcal{P}}_2$, also yields that $\tilde{\mathcal{Q}}_2$ is nonsingular. Hence,

$$\begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix} \mathcal{R}_1^{-1}\mathcal{R}_2,$$

that is,

$$\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} \mathcal{R}_2^{-1}\mathcal{R}_1 = \begin{bmatrix} \mathcal{B}^{-1}\mathcal{A} \\ \mathcal{C} \end{bmatrix}.$$

Therefore, we have that

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = \tilde{\mathcal{Q}}_1\tilde{\mathcal{Q}}_2^{-1}. \qquad \square$$

According to Lemma 3.5, the CSD of $\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix}$ reveals the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. On the other hand, it is also possible to obtain the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ from the orthogonal complement of $\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix}$. To explain this, let us consider $\mathcal{Q}_1 \in \mathbf{R}^{\nu\times\nu}$, $\mathcal{Q}_2 \in \mathbf{R}^{\mu\times\nu}$ such that

$$\mathcal{Q}_1^T\mathcal{Q}_1 + \mathcal{Q}_2^T\mathcal{Q}_2 = I_\nu, \quad \begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} \mathcal{R},$$

in which $\mathcal{R}$ is nonsingular. Since $\tilde{\mathcal{Q}}_2$ is nonsingular, $\mathcal{Q}_1$ is nonsingular as well. If the CSD of $\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix}$ is given by

$$\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{U} & \\ & \mathcal{V} \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \mathcal{W},$$

where $\mathcal{U}, \mathcal{W} \in \mathbf{R}^{\nu\times\nu}$, $\mathcal{V} \in \mathbf{R}^{\mu\times\mu}$ are orthogonal, $\mathcal{U}\Sigma_1\mathcal{W}$ and $\mathcal{V}\Sigma_2\mathcal{W}$ are the SVDs of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively, then we have that

$$\tilde{\mathcal{Q}}_1\tilde{\mathcal{Q}}_2^{-1} = (-\mathcal{U})(\Sigma_2\Sigma_1^{-1})\mathcal{V}^T,$$

that is,

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = (-\mathcal{U})(\Sigma_2\Sigma_1^{-1})\mathcal{V}^T. \tag{3.16}$$

Equivalently, the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is given by $(-\mathcal{U})(\Sigma_2\Sigma_1^{-1})\mathcal{V}^T$. $\square$

Depending on the dimensions μ and ν, we choose to base the RSVD algorithm on either $\mathcal{Q}_1$ and $\mathcal{Q}_2$, or $\tilde{\mathcal{Q}}_1$ and $\tilde{\mathcal{Q}}_2$. If $\mu < \nu$, we use $\tilde{\mathcal{Q}}_1$ and $\tilde{\mathcal{Q}}_2$. If $\mu > \nu$, we use $\mathcal{Q}_1$ and $\mathcal{Q}_2$. Since the computation of $\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix}$ itself is cheaper than that of $\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix}$, we also resort to $\mathcal{Q}_1$ and $\mathcal{Q}_2$ in the case $\mu = \nu$.

3.3. The RSVD components. After the computation of the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, the only thing that remains to be done is the backtransformation of the results to the original matrix spaces of A, B, and C.

Assume that orthogonal matrices U_b, V_c and nonsingular matrices $\mathcal{X}, \mathcal{Y}$ satisfy (3.11), the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is given by (3.16), and set

$$\begin{aligned} X &= \operatorname{diag}\{\mathcal{A}_{11}^{-1}, \mathcal{B}_{22}^{-1}, \mathcal{C}_{23}\mathcal{A}_{33}^{-1}, -\mathcal{U}^T\mathcal{B}^{-1}, I_{n-r_{ab}}\}\mathcal{X}, \\ Y &= \mathcal{Y}\operatorname{diag}\{I_{k_1}, \mathcal{A}_{22}^{-1}\mathcal{B}_{22}, \mathcal{C}_{23}^{-1}, \mathcal{C}^{-1}\mathcal{V}^T, I_{m-r_{ac}}\}, \\ U &= U_b\operatorname{diag}\{I_{l-r_b}, I_{k_2}, -\mathcal{U}\}, \quad V = V_c\operatorname{diag}\{I_{p-r_c}, I_{k_3}, \mathcal{V}\}; \end{aligned} \tag{3.17}$$

then it is easy to verify that the RSVD (2.1) of (A, B, C) is given by (XAY, XBU, VCY).

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.4. A numerical algorithm for the RSVD. We have now the following overall numerical method[1] for the computation of the RSVD of a matrix triplet A, B, C.

ALGORITHM 4.

Input: *Matrices* $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$.

Output: *RSVD* (2.1) *of the matrix triplet* (A, B, C).

Step 1: *Compute the condensed form* (3.11).

Step 2: *Perform the QR-factorization of* $\begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix}$ *with column pivoting to get an orthonormal basis of the following null space:*

$$I = \mathcal{P}_1^T\mathcal{P}_1 + \mathcal{P}_2^T\mathcal{P}_2, \qquad \mathcal{P}_1 \in \mathbf{R}^{\nu\times\nu}, \mathcal{P}_2 \in \mathbf{R}^{\mu\times\nu},$$
$$0 = \begin{bmatrix} \mathcal{P}_1^T & \mathcal{P}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix}.$$

Step 3: *Perform the QR-factorization of* $\begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix}$ *to get an orthonormal basis of the following range space:*

$$I = \mathcal{Q}_1^T\mathcal{Q}_1 + \mathcal{Q}_2^T\mathcal{Q}_2, \qquad \mathcal{Q}_1 \in \mathbf{R}^{\nu\times\nu}, \mathcal{Q}_2 \in \mathbf{R}^{\mu\times\nu},$$
$$\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} \mathcal{R} = \begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix},$$

where $\mathcal{R} \in \mathbf{R}^{\mu\times\mu}$ *is a nonsingular triangular matrix.*

Step 4: *Compute the CSD of* $\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix}$*:*

$$\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{U} & \\ & \mathcal{V} \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \mathcal{W}.$$

Step 5: *Compute matrices* X, Y, U, *and* V *using* (3.17), *in which* $\mathcal{A}_{ii}^{-1}$, $i = 1, 2, 3$, $\mathcal{B}_{22}^{-1}$, $\mathcal{B}^{-1}$, $\mathcal{C}_{23}^{-1}$, *and* $\mathcal{C}^{-1}$ *are computed using their SVDs or QR-factorizations, such that* (XAY, XBU, VCY) *are in the form* (2.1).

Output XAY, XBU, VCY, X, Y, U, *and* V.

In Algorithm 4, Steps 1, 2, and 3 amount to the QR-factorizations with column pivoting and URV decomposition, which can be implemented in a stable and efficient way (see [1]); Step 5 can be carried out by SVD or QR-factorization; Step 4 is the critical stage in Algorithm 4; one can make use of Van Loan's CSD algorithm here [18].

4. Numerical examples. We will verify only the numerical performance of the core part of Algorithm 4 formed by Steps 2, 3, and 4, which computes the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. Steps 2 and 3 are implemented using QR-factorization with column pivoting and URV decomposition. Step 4 is implemented as Van Loan's CSD algorithm [18], in which we set the parameter $\tau = \frac{1}{\sqrt{2}}$, which minimizes a backward error bound [3]. All calculations were carried out in MATLAB 5.0 on an HP 712/80 workstation with IEEE standard (machine accuracy $\epsilon \cong 10^{-16}$).

We assume that $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbf{R}^{\mu\times\mu}$ are square. To quantify the accuracy of the results, we define the residuals

$$resSVD = \frac{\|\hat{\Sigma} - \Sigma\|_2}{\mu\|\hat{\Sigma}\|_2},$$

[1] MATLAB code is available upon request.

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

in which

$$\hat{\Sigma} = \mathrm{diag}\{\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_\mu\}, \quad \Sigma = \mathrm{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_\mu\},$$

where $\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_\mu$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_\mu$ are the exact and the computed singular values of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, respectively.

In a first setup, consisting of eight examples, the triplet $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is of the form

$$\mathcal{A} = R_1\Sigma_1 R_2, \quad \mathcal{B} = R_1\Sigma_2 W_1, \quad \mathcal{C} = W_2\Sigma_3 R_2,$$

in which R_1, R_2, W_1, and W_2 are randomly chosen orthogonal matrices (obtained from a QR-factorization of a matrix of which the entries are drawn from a uniform distribution over $[0, 1)$); the way in which the entries of

$$\Sigma_i = \mathrm{diag}\{\sigma_{i1}, \ldots, \sigma_{i\mu}\}, \quad i = 1, 2, 3,$$

are chosen is specific for each example. Obviously, the exact SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is given by

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = W_1^T \Sigma W_2^T \text{ with } \Sigma = \Sigma_2^{-1}\Sigma_1\Sigma_3^{-1}.$$

We will choose the diagonal elements of Σ_1, Σ_2, and Σ_3 in such a way that different situations in terms of the condition numbers of $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are to be dealt with. In the description below, we use the symbol η to denote a value drawn from a uniform distribution over $[0, 1)$; $\kappa(M)$ is the 2-norm condition number of matrix M.

EXAMPLE 1.

$$\sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 1, 2, 3, \quad j = 1, \ldots, \mu.$$

This is a situation in which $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are well-conditioned.

EXAMPLE 2.

$$\sigma_{1j} = (1 + \eta_{1j}) * 10^{-\frac{9}{\mu}j}, \quad \sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 2, 3, \quad j = 1, \ldots, \mu.$$

This means that $\mathcal{B}$ and $\mathcal{C}$ are well-conditioned, but, on the other hand, $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^9).$$

EXAMPLE 3.

$$\sigma_{2j} = (2 + \eta_{2j}) * 10^{\frac{3}{\mu}j}, \quad \sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 1, 3, \quad j = 1, \ldots, \mu.$$

$\mathcal{A}$ and $\mathcal{C}$ are well-conditioned; $\mathcal{B}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ have a moderate condition number.

EXAMPLE 4.

$$\sigma_{3j} = (3 + \eta_{3j}) * 10^{\frac{4}{\mu}j}, \quad \sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 1, 2, \quad j = 1, \ldots, \mu.$$

$\mathcal{A}$ and $\mathcal{B}$ are well-conditioned; $\mathcal{C}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ have a moderate condition number.

EXAMPLE 5.

$$\sigma_{3j} = (j+3)+\eta_{3j}, \quad \sigma_{1j} = (1+\eta_{1j})*10^{-\frac{9}{\mu}j}, \quad \sigma_{2j} = (2+\eta_{2j})*10^{-\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$\mathcal{C}$ is well-conditioned; $\mathcal{B}$ has a moderate condition number; $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^6).$$

EXAMPLE 6.

$$\sigma_{2j} = (j+2)+\eta_{2j}, \quad \sigma_{1j} = (1+\eta_{1j})*10^{-\frac{9}{\mu}j}, \quad \sigma_{3j} = (3+\eta_{3j})*10^{-\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{B}$ is well-conditioned; $\mathcal{C}$ has a moderate condition number; $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^6).$$

EXAMPLE 7.

$$\sigma_{1j} = (j+1)+\eta_{1j}, \quad \sigma_{2j} = (2+\eta_{2j})*10^{\frac{3}{\mu}j}, \quad \sigma_{3j} = (3+\eta_{3j})*10^{\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{A}$ is well-conditioned; $\mathcal{B}$ and $\mathcal{C}$ have a moderate condition number; $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is ill-conditioned, with $\kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^6)$.

EXAMPLE 8.

$$\sigma_{1j} = (1+\eta_{1j})*10^{-\frac{9}{\mu}j}, \quad \sigma_{2j} = (2+\eta_{2j})*10^{\frac{4}{\mu}j}, \quad \sigma_{3j} = (3+\eta_{3j})*10^{-\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{B}$ and $\mathcal{C}$ have a moderate condition number; $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^{10}).$$

In each of these examples above, the dimension μ was varied between 51 and 100. The values of $resSVD$ that were obtained in Examples 1–4 are plotted in Figure 4.1 and Examples 5–8 are plotted in Figure 4.2. From these figures, we see that our results are satisfactory.

In a second setup, we choose

$$\mathcal{B} = (\text{wilkinson}(\mu) - \mu * I_\mu) * V(\mu), \quad \mathcal{A} = (\text{wilkinson}(\mu) - \mu * I_\mu) * \text{hilb}(\mu),$$
$$\mathcal{C} = U(\mu) * \text{hilb}(\mu),$$

in which $U(\mu), V(\mu) \in \mathbf{R}^{\mu\times\mu}$ are randomly chosen orthogonal matrices and wilkinson(μ) and hilb(μ) are the μth-order Wilkinson matrix and Hilbert matrix, respectively (which can be generated by the "wilkinson" and "hilb" command in MATLAB). The exact singular values of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are obviously equal to 1. They are estimated in 4 different ways:

- By using the algorithm developed in this paper.
- By means of the MATLAB command

  $$\text{svd}(\text{inv}(\mathcal{B}) * \mathcal{A} * \text{inv}(\mathcal{C})),$$

  in which $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is formed explicitly. This technique is referred to as "direct method 1."
- By means of the MATLAB commands

  $$[U1, S1, V1] = \text{svd}(\mathcal{B}), \quad \tilde{\mathcal{B}} = V1 * \text{inv}(S1) * U1',$$

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

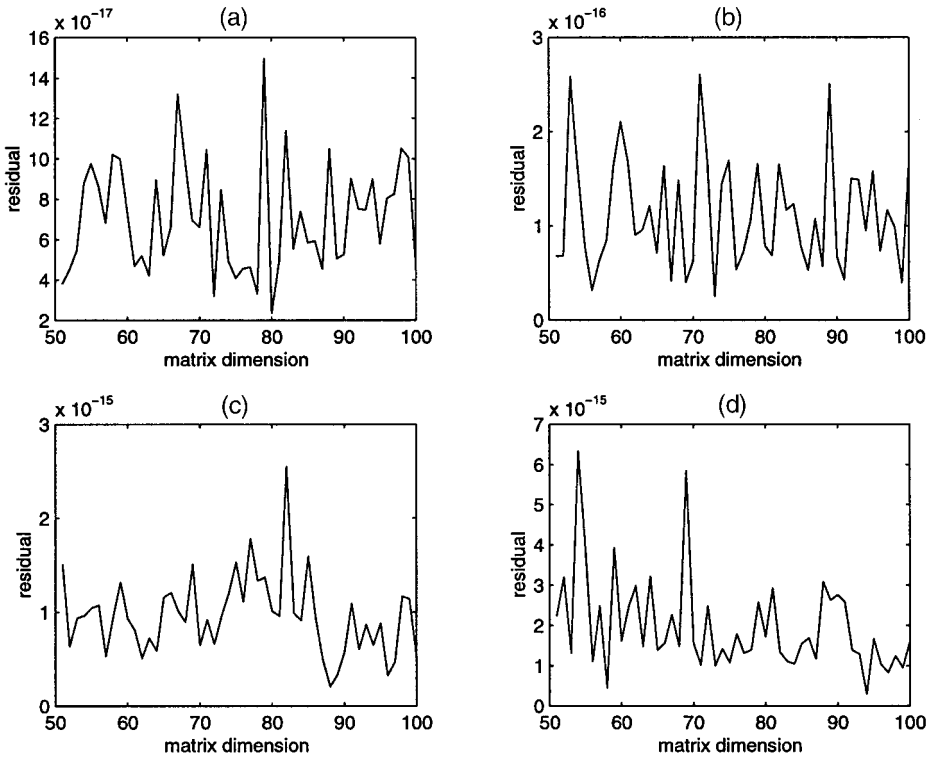


FIG. 4.1. (a)–(d) *depict Examples* 1–4, *respectively.*

$$[U2, S2, V2] = \text{svd}(\mathcal{C}), \quad \tilde{\mathcal{C}} = V2 * \text{inv}(S2) * U2',$$

and

$$\text{svd}(\tilde{\mathcal{B}} * \mathcal{A} * \tilde{\mathcal{C}}).$$

This technique is referred to as "direct method 2."

- By means of the MATLAB command

$$\text{gsvd}(\tilde{\mathcal{A}}, \mathcal{C}),$$

in which

$$\tilde{\mathcal{A}} = \text{hilb}(\mu).$$

It is easy to see that with this particular definition of $\tilde{\mathcal{A}}$, the singular values of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ and $\tilde{\mathcal{A}}\mathcal{C}^{-1}$ are the same. The "gsvd" routine in MATLAB is an implementation of the QSVD algorithm by Van Loan [18]. This technique is referred to as the "GSVD method."

In the following table, we list the values of the residue $resSVD$ obtained by these 4 methods, for $\mu = 8, 9, \ldots, 12$.

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

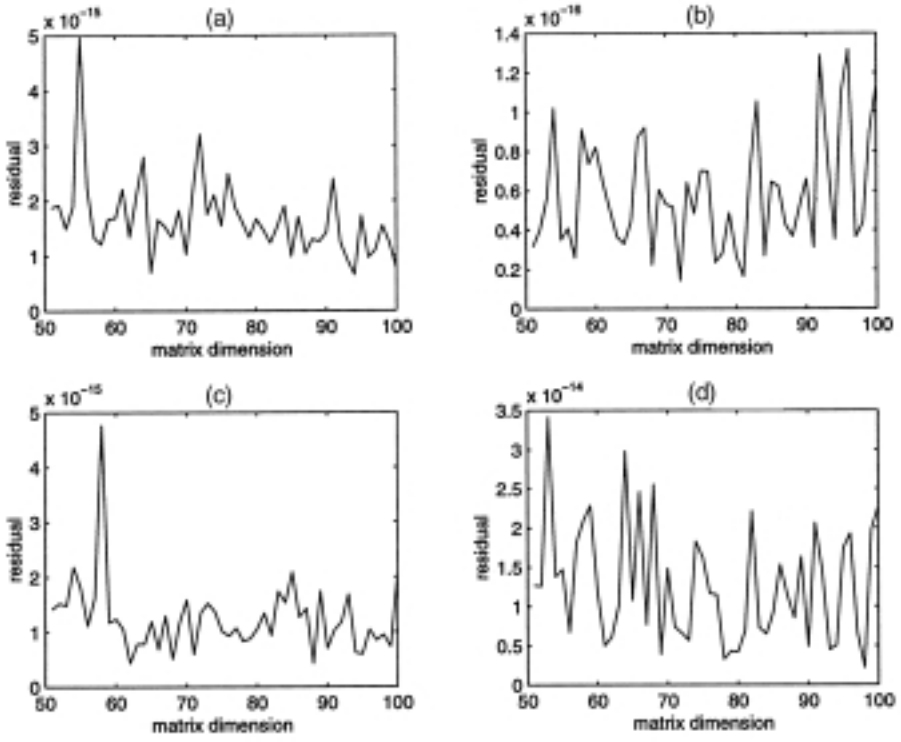


FIG. 4.2. (a)–(d) *depict Examples* 5–8, *respectively.*

	Our method	GSVD method	Direct method 1	Direct method 2
$\mu = 8$	$2.446643637e-08$	$1.106089772e-08$	$1.591006943e-07$	$2.241213379e-07$
$\mu = 9$	$4.635656435e-07$	$3.593259342e-07$	$1.009544058e-05$	$4.035071152e-06$
$\mu = 10$	$1.328935076e-05$	$2.534763366e-06$	$2.875451264e-04$	$1.094955938e-04$
$\mu = 11$	$2.659354824e-04$	$3.018865472e-04$	$2.008595385e-02$	$2.639692842e-03$
$\mu = 12$	$1.686164914e-02$	$1.683184121e-02$	$2.138542385e+00$ Matrix is close to singular! when $\mu = 12$	$1.189748962e-01$

These results clearly show that our method is comparable with the GSVD method and performs better than the direct method 1 and the direct method 2. One of the main reasons for the differences in performance is that Hilbert matrix is very ill-conditioned. For example, for a Hilbert matrix of order μ we have

μ	8	9	10	11	12
Condition number	$1.5258e+10$	$4.9315e+11$	$1.6025e+13$	$5.2196e+14$	$1.712120390592042e+16$

5. Conclusions. We explained how the RSVD of an arbitrary matrix triplet $A \in \mathbf{R}^{n\times m}$, $B \in \mathbf{R}^{n\times l}$, $C \in \mathbf{R}^{p\times m}$ can be computed using a CSD-based QR-type method.

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

First, the matrices A, B, C are reduced to a lower-dimensional triplet $\mathcal{A}, \mathcal{B}, \mathcal{C}$, with $\mathcal{B}$ and $\mathcal{C}$ nonsingular, using orthogonal transformations such as the QR-factorization with column pivoting and the URV decomposition. Next, the restricted singular values and vectors, defined by the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, are computed using Van Loan's CSD algorithm, without having to form $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ explicitly. Some numerical examples were given to illustrate the performance of the presented method.

Our algorithm is of the QR-type. It is well known that, in general, QR-type algorithms do not have a high relative accuracy. On the other hand, the Jacobi-type QSVD algorithm proposed by Drmač [8, 9, 10] does have a high relative accuracy. How this procedure can be generalized for the RSVD is an important topic of further research.

Acknowledgments. We would like to thank Prof. Lars Eldén and the anonymous referees for their kind and detailed comments on an early version of the text. This paper benefited a lot from their valuable suggestions.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bishof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide*, SIAM, Philadelphia, 1992.

[2] Z.J. Bai and J.W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.

[3] Z. Bai, *The CSD, GSVD, their Applications and Computations*, IMA Preprint 958, University of Minnesota, Minneapolis, 1992.

[4] B. De Moor and G.H. Golub, *Generalized Singular Value Decompositions: A Proposal for a Standardized Nomenclature*, Tech. Report 89-10, SISTA, E.E. Dept. (ESAT), K.U. Leuven, Leuven, Belgium, 1989.

[5] B.L.R. De Moor and G.H. Golub, *The restricted singular value decomposition: Properties and applications*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 401–425.

[6] B. De Moor and P. Van Dooren, *Generalizations of the singular value and QR decomposition*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 993–1014.

[7] B. De Moor, *On the structure of generalized singular value and QR decompositions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 347–358.

[8] Z. Drmač, *Fast and accurate algorithms for PSVD and GSVD*, in SIAM Annual Meeting, Stanford University, Stanford, CA, 1997.

[9] Z. Drmač, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.

[10] Z. Drmač, *A tangent algorithm for computing the generalized singular value decomposition*, SIAM J. Numer. Anal., 35 (1998), pp. 1804–1832.

[11] K.V. Fernando and S. 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, Philadelphia, 1988, pp. 128–140.

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

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

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

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

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

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

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

[19] H. Zha, *The restricted singular value decomposition of matrix triplets*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 172–194.
[20] H. Zha, *A numerical algorithm for computing the restricted singular value decomposition of matrix triplets*, Linear Algebra Appl., 168 (1992), pp. 1–25.
[21] H. Zha, *Computing the generalized singular values/vectors of large sparse or structured matrix pairs*, Numer. Math., 72 (1996), pp. 391–417.

Downloaded 11/15/16 to 134.58.253.57. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php