Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 1998-97

A QR-type Reduction for Computing the SVD of a General Matrix Product/Quotient[1]

Delin Chu, Lieven De Lathauwer and Bart De Moor[2]

July 2003

Numerische Mathematik
Vol. 95, No. 1, July 2003, pp. 101–121

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/SISTA/delathauwer/reports/ldl-98-97.ps.gz*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SCD-SISTA, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium, Tel. 32/16/32 11 43, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.ac.be/sista*. E-mail: *Lieven.DeLathauwer@esat.kuleuven.ac.be*, *Bart.DeMoor@esat.kuleuven.ac.be*.

Abstract

In this paper, a QR-type reduction technique is developed for the computation of the SVD of a general matrix product/quotient $A = A_1^{s_1} A_2^{s_2} \cdots A_m^{s_m}$ with $A_i \in \mathbf{R}^{n\times n}$ and $s_i = 1$ or $s_i = -1$. First the matrix A is reduced by at most m QR-factorizations to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$, where $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}$ and $(Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$. Then the SVD of A is obtained by computing the CSD (Cosine-Sine Decomposition) of $Q_{11}^{(1)}$ and $Q_{21}^{(1)}$ using the Matlab command *gsvd*. The performance of the proposed method is verified by some numerical examples.

Acknowledgements

This work is supported by grants from several funding agencies:

1. Research Council KUL: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering)
2. Flemish Government: FWO (Fund for Scientific Research - Flanders) projects G292.95 and G256.97, FWO Research Communities ICCoS and ANMMM,
3. Belgian Federal Government: DWTC (IUAP IV-02 and IUAP V-22).

Part of this research was carried out while D. Chu was a visiting researcher at the K.U.Leuven. L. De Lathauwer holds a permanent research position with the French CNRS; he also holds a honorary post-doctoral research mandate with the FWO. B. De Moor is a full professor at the K.U.Leuven. The scientific responsibility is assumed by the authors.

A QR-type Reduction for Computing the SVD of a General Matrix Product/Quotient*

Delin Chu* Lieven De Lathauwer† and Bart De Moor‡

March 23, 2002

Abstract

In this paper, a QR-type reduction technique is developed for the computation of the SVD of a general matrix product/quotient $A = A_1^{s1} A_2^{s2} \cdots A_m^{sm}$ with $A_i \in \mathbf{R}^{n\times n}$ and $s_i = 1$ or $s_i = -1$. First the matrix A is reduced by at most m QR-factorizations to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$, where $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}$ and $(Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$. Then the SVD of A is obtained by computing the CSD (Cosine-Sine Decomposition) of $Q_{11}^{(1)}$ and $Q_{21}^{(1)}$ using the Matlab command *gsvd*. The performance of the proposed method is verified by some numerical examples.

Keywords: SVD, CSD, QR-factorization.

AMS subject classification: 65F15, 65H15.

1 Introduction

This paper deals with a new method for the computation of the Singular Value Decomposition (SVD) of a sequence of matrices in product/quotient form. The simplest forms of these Generalized

*This work is supported by grants from several funding agencies:

1. Research Council KUL: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering)
2. Flemish Government: FWO (Fund for Scientific Research - Flanders) projects G292.95 and G256.97, FWO Research Communities ICCoS and ANMMM,
3. Belgian Federal Government: DWTC (IUAP IV-02 and IUAP V-10-29).

Part of this research was carried out while D. Chu was a visiting researcher at the K.U.Leuven. L. De Lathauwer holds a permanent research position with the French CNRS; he also holds a honorary post-doctoral research mandate with the FWO. B. De Moor is a full professor at the K.U.Leuven. The scientific responsibility is assumed by the authors.

*Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email: matchudl@math.nus.edu.sg

†ETIS, UPRES-A 8051, ENSEA / UCP, 6, avenue du Ponceau, BP 44, F 95014 Cergy-Pontoise Cedex, France. Email: Lieven.DeLathauwer@ensea.fr

‡SCD-SISTA, Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium. Email: Bart.DeMoor@esat.kuleuven.ac.be

SVD's (GSVD), for two matrices, are the well-known Quotient SVD (QSVD) (introduced in [17] and refined in [14]) and Product SVD (PSVD) (proposed in [9], refining ideas introduced in [13]). One of the three possible forms involving three matrices, is the so-called Restricted SVD (RSVD) (introduced in its explicit form in [20] and further developed and discussed in [5]). General schemes have been discussed in e.g. [6, 7].

The GSVD is one of the 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, open and closed loop balancing, etc.

Like the QSVD, PSVD and RSVD, the SVD of a general matrix product/quotient has many applications. For example, it is important for the estimation of Lyapunov exponents for dynamic systems [12]. Consider difference equations

$$\Theta_{k+1} = \Phi_k^{s_k}\Theta_k,\ \Theta_0 = I,\ \Phi_k \in \mathbf{R}^{n\times n},\ s_k = 1 \text{ or } s_k = -1. \tag{1}$$

The ith Lyapunov exponent is then defined by

$$\lambda_i = \lim_{k\to\infty}\log(\sigma_i(\Theta_k))/k,$$

where $\sigma_i(\Theta_k)$ is the ith biggest singular value of Θ_k. Discretizations of ordinary differential equations may also lead to sequences of matrix products/quotients [11].

Up to now, the calculation of the QSVD has been extensively studied. The fact that the Cosine-Sine Decomposition (CSD) of a partitioned column-orthogonal matrix $\begin{bmatrix} A \\ C \end{bmatrix}$ corresponds to the QSVD of the couple (A, C), forms the basis for two backward stable algorithms, proposed in [16] and [18]. Recently, the computation of the QSVD via the CSD and the Lanczos bidiagonalization process has been studied in [21]. The Kogbetliantz algorithm has been generalized in [15] for the computation of the QSVD. In [15], the 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 the algorithm in [15], also backward stable, is given in [2]. This technique is implemented as the LAPACK procedure STGSJA() [2].

The Kogbetliantz algorithm has also been generalized for the computation of the RSVD [19]. However, Jacobi-type algorithms typically have a (moderately) higher complexity than QR-type algorithms. Moreover, the Jacobi approach has so far only been followed for problems involving at most three matrices; generalizing the scheme for long matrix sequences proves to be hard. Recently an implicit bidiagonalization QR-type algorithm for the computation of the SVD of a general matrix product/quotient has been proposed in [11]. For quotient factors this scheme requires solving a series of upper triangular systems of linear equations during the bidiagonalization.

In this paper, we generalize the idea in [4] for computing the RSVD to propose a QR-type *reduction* technique for computing the SVD of a general matrix product/quotient

$$A = A_1^{s_1}A_2^{s_2}\cdots A_m^{s_m}$$

with $A_i \in \mathbf{R}^{n\times n}$, $s_i = 1$ or $s_i = -1$. Like in [11], we assume for simplicity that the matrices A_i in (2) are square; as was pointed out in [7], this does not affect the generality of what follows.

We will show that the matrix A can be reduced by at most m QR-factorizations to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ with $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}, (Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$. The main advantage of this QR-type reduction is the way in which quotients are dealt with. Finally the SVD of A can be found by resorting, e.g., to Van Loan's CSD method [18].

The next section describes how the QR-type reduction can be realized. Section 3 contains a number of numerical experiments. Section 4 presents the conclusions.

2 QR-Type Reduction

Consider a matrix A of the following form:

$$\begin{cases} A = A_1^{s_1} A_2^{s_2} \cdots A_m^{s_m}, \quad m \geq 2, \quad A_i \in \mathbf{R}^{n\times n},\ s_i = 1 \text{ or } s_i = -1, \\ A_i \text{ is nonsingular if } s_i = -1. \end{cases} \tag{2}$$

The purpose of this section is to show how the matrix A can be reduced to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ with $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}, (Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$, by performing at most m QR-factorizations.

First, the following three lemmas describe the cases $m = 2$ and $m = 3$:

Lemma 1 *[14, 15] Assume that $A_1, A_2 \in \mathbf{R}^{n\times n}$ and A_2 is nonsingular. Let the QR factorization of $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ be*

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} R^{(1)} \\ 0 \end{bmatrix},\ R^{(1)}, Q_{11}^{(1)}, Q_{12}^{(1)}, Q_{21}^{(1)}, Q_{22}^{(1)} \in \mathbf{R}^{n\times n}.$$

Then $Q_{21}^{(1)}$ is nonsingular and

$$A_1 A_2^{-1} = Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}.$$

Lemma 2 *Given matrices $A_i \in \mathbf{R}^{n\times n}, i = 1, 2, 3$.*

(a) Assume A_3 is nonsingular. Let the QR factorization of $\begin{bmatrix} A_2 \\ A_3 \end{bmatrix}$ be

$$\begin{bmatrix} A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} R^{(2)} \\ 0 \end{bmatrix},\ R^{(2)}, Q_{11}^{(2)}, Q_{12}^{(2)}, Q_{21}^{(2)}, Q_{22}^{(2)} \in \mathbf{R}^{n\times n}. \tag{3}$$

Then $Q_{21}^{(2)}$ is nonsingular. Furthermore, let the QR factorization of $\begin{bmatrix} A_1 Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix}$ be

$$\begin{bmatrix} A_1 Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} R^{(1)} \\ 0 \end{bmatrix},\ R^{(1)}, Q_{11}^{(1)}, Q_{12}^{(1)}, Q_{21}^{(1)}, Q_{22}^{(1)} \in \mathbf{R}^{n\times n}. \tag{4}$$

We have that $Q_{21}^{(1)}$ is nonsingular and

$$A_1 A_2 A_3^{-1} = Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}.$$

(b) Assume A_1 and A_3 nonsingular. Let the QR factorization of $\begin{bmatrix} A_2 \\ A_3 \end{bmatrix}$ *be*

$$\begin{bmatrix} A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(2)} \end{bmatrix}, \; R^{(2)}, Q_{11}^{(2)}, Q_{12}^{(2)}, Q_{21}^{(2)}, Q_{22}^{(2)} \in \mathbf{R}^{n\times n}, \tag{5}$$

and the QR factorization of $\begin{bmatrix} A_1^T Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix}$ *be*

$$\begin{bmatrix} A_1^T Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(1)} \end{bmatrix}, \; R^{(1)}, Q_{11}^{(1)}, Q_{12}^{(1)}, Q_{21}^{(1)}, Q_{22}^{(1)} \in \mathbf{R}^{n\times n}. \tag{6}$$

Then $Q_{11}^{(2)}$ and $Q_{21}^{(1)}$ are nonsingular, and

$$A_1^{-1} A_2 A_3^{-1} = Q_{11}^{(1)} (Q_{21}^{(1)})^{-1}.$$

Proof. (a) This part follows directly from Lemma 1:

$$A_2 A_3^{-1} = Q_{11}^{(2)} (Q_{21}^{(2)})^{-1}, \; (A_1 Q_{11}^{(2)})(Q_{21}^{(2)})^{-1} = Q_{11}^{(1)} (Q_{21}^{(1)})^{-1}.$$

(b) As A_1 and A_3 are nonsingular, $Q_{11}^{(2)}, Q_{22}^{(2)}, R^{(2)}$ and $Q_{12}^{(1)}, Q_{21}^{(1)}, R^{(1)}$ are nonsingular as well. Since

$$\begin{bmatrix} (Q_{11}^{(2)})^T & (Q_{21}^{(2)})^T \end{bmatrix} \begin{bmatrix} A_1 Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = 0,$$

we have that

$$\begin{bmatrix} A_1 Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{R}^{(1)} \end{bmatrix}, \quad \text{for some } \tilde{R}^{(1)} \in \mathbf{R}^{n\times n};$$

$\tilde{R}^{(1)}$ is nonsingular because $Q_{21}^{(1)}$ is nonsingular. Hence,

$$\begin{bmatrix} A_1 Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} (R^{(2)})^{-1} \tilde{R}^{(1)},$$

i.e.,

$$\begin{bmatrix} Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} A_1^{-1} A_2 \\ A_3 \end{bmatrix} (R^{(2)})^{-1} \tilde{R}^{(1)}.$$

Therefore, we have that

$$A_1^{-1} A_2 A_3^{-1} = Q_{11}^{(1)} (Q_{21}^{(1)})^{-1}.$$

□

The next lemma is the version of Lemma 2 for the reduction of $A_1^{s_1} A_2^{s_2} A_3^{s_3}$ from left to right.

Lemma 3 *Given matrices $A_i \in \mathbf{R}^{n\times n}, i = 1, 2, 3$.*

(a) Assume A_2 and A_3 are nonsingular. Let the QR factorization of $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ be given by

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} R^{(2)} \\ 0 \end{bmatrix}, \; R^{(2)}, Q_{11}^{(2)}, Q_{12}^{(2)}, Q_{21}^{(2)}, Q_{22}^{(2)} \in \mathbf{R}^{n\times n} \tag{7}$$

and let the QR factorization of $\begin{bmatrix} Q_{11}^{(2)} \\ A_3 Q_{21}^{(2)} \end{bmatrix}$ be given by

$$\begin{bmatrix} Q_{11}^{(2)} \\ A_3 Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} R^{(1)} \\ 0 \end{bmatrix}, \; R^{(1)}, Q_{11}^{(1)}, Q_{12}^{(1)}, Q_{21}^{(1)}, Q_{22}^{(1)} \in \mathbf{R}^{n\times n}. \tag{8}$$

Then we have that

$$A_1 A_2^{-1} A_3^{-1} = Q_{11}^{(1)} (Q_{21}^{(1)})^{-1},$$

in which $Q_{21}^{(1)}$ is nonsingular.

(b) Assume A_2 is nonsingular. Let the QR factorization of $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ be given by

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(2)} \end{bmatrix}, \; R^{(2)}, Q_{11}^{(2)}, Q_{12}^{(2)}, Q_{21}^{(2)}, Q_{22}^{(2)} \in \mathbf{R}^{n\times n}, \tag{9}$$

and let the QR factorization of $\begin{bmatrix} Q_{11}^{(2)} \\ A_3^T Q_{21}^{(2)} \end{bmatrix}$ be given by

$$\begin{bmatrix} Q_{11}^{(2)} \\ A_3^T Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(1)} \end{bmatrix}, \; R^{(1)}, Q_{11}^{(1)}, Q_{12}^{(1)}, Q_{21}^{(1)}, Q_{22}^{(1)} \in \mathbf{R}^{n\times n}. \tag{10}$$

Then we have that

$$A_1 A_2^{-1} A_3 = Q_{11}^{(1)} (Q_{21}^{(1)})^{-1},$$

in which $Q_{21}^{(1)}$ is nonsingular.

Proof. The proof is similar to that of Lemma 2 and hence is omitted here. □

Assume a matrix $A = A_1^{s_1} A_2^{s_2} A_3^{s_3}$, reduced to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ with $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}$, $(Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$, as shown in Lemma 2. Now we explain how the matrix $\hat{A}^{\hat{s}} A$ or $A\hat{A}^{\hat{s}}$, in which $\hat{A} \in \mathbf{R}^{n\times n}$, $\hat{s} \in \{\pm 1\}$, and $\hat{A}$ is nonsingular if $\hat{s} = -1$, can be reduced to the form $\hat{Q}_{11}^{(1)}(\hat{Q}_{21}^{(1)})^{-1}$ with $\hat{Q}_{11}^{(1)}, \hat{Q}_{21}^{(1)} \in \mathbf{R}^{n\times n}, (\hat{Q}_{11}^{(1)})^T \hat{Q}_{11}^{(1)} + (\hat{Q}_{21}^{(1)})^T \hat{Q}_{21}^{(1)} = I$, by performing at most one extra QR-decomposition.

We consider the reduction of $\hat{A}^{\hat{s}} A$ first. The case $\hat{s} = 1$ is trivial: compute the QR factorization of $\begin{bmatrix} \hat{A} Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix}$ to get

$$\begin{bmatrix} \hat{A} Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(1)} & \hat{Q}_{12}^{(1)} \\ \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \end{bmatrix} \begin{bmatrix} \hat{R}^{(1)} \\ 0 \end{bmatrix}, \; \hat{R}^{(1)}, \hat{Q}_{11}^{(1)}, \hat{Q}_{12}^{(1)}, \hat{Q}_{21}^{(1)}, \hat{Q}_{22}^{(1)} \in \mathbf{R}^{n\times n}.$$

In the case $\hat{s} = -1$, application of Lemma 2 to the matrix product $\hat{A}^{-1}Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ would lead to the following definition of orthogonal matrix $\begin{bmatrix} \hat{Q}_{11}^{(1)} & \hat{Q}_{12}^{(1)} \\ \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \end{bmatrix}$:

$$\begin{bmatrix} Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(2)} & \hat{Q}_{12}^{(2)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{R}^{(2)} \end{bmatrix}, \ \hat{R}^{(2)}, \hat{Q}_{11}^{(2)}, \hat{Q}_{12}^{(2)}, \hat{Q}_{21}^{(2)}, \hat{Q}_{22}^{(2)} \in \mathbf{R}^{n \times n}, \tag{11}$$

$$\begin{bmatrix} \hat{A}^T\hat{Q}_{11}^{(2)} \\ \hat{Q}_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(1)} & \hat{Q}_{12}^{(1)} \\ \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{R}^{(1)} \end{bmatrix}, \ \hat{R}^{(1)}, \hat{Q}_{11}^{(1)}, \hat{Q}_{12}^{(1)}, \hat{Q}_{21}^{(1)}, \hat{Q}_{22}^{(1)} \in \mathbf{R}^{n \times n}. \tag{12}$$

However, this would impose the calculation of two kernels. On the other hand, one QR-factorization can always be avoided by combining Eq. (11) with the computation in the preceding step. If $s_1 = 1$, then Eqs. (4) and (11) can be combined to

$$\begin{bmatrix} A_1Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(2)} & \hat{Q}_{12}^{(2)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{\mathcal{R}}^{(2)} \end{bmatrix}, \ \hat{\mathcal{R}}^{(2)}, \hat{Q}_{11}^{(2)}, \hat{Q}_{12}^{(2)}, \hat{Q}_{21}^{(2)}, \hat{Q}_{22}^{(2)} \in \mathbf{R}^{n \times n}.$$

If $s_1 = -1$, then Eqs. (6) and (11) can be combined to

$$\begin{bmatrix} A_1^TQ_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(2)} & \hat{Q}_{12}^{(2)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{R}}^{(1)} \\ 0 \end{bmatrix}, \ \hat{\mathcal{R}}^{(1)}, \hat{Q}_{11}^{(2)}, \hat{Q}_{12}^{(2)}, \hat{Q}_{21}^{(2)}, \hat{Q}_{22}^{(2)} \in \mathbf{R}^{n \times n}.$$

Now, we consider the reduction of $A\hat{A}^{\hat{s}}$. The case $\hat{s} = -1$ is trivial: compute the QR factorization of $\begin{bmatrix} Q_{11}^{(1)} \\ \hat{A}Q_{21}^{(1)} \end{bmatrix}$ to get

$$\begin{bmatrix} Q_{11}^{(1)} \\ \hat{A}Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(1)} & \hat{Q}_{12}^{(1)} \\ \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \end{bmatrix} \begin{bmatrix} \hat{R}^{(1)} \\ 0 \end{bmatrix}, \ \hat{R}^{(1)}, \hat{Q}_{11}^{(1)}, \hat{Q}_{12}^{(1)}, \hat{Q}_{21}^{(1)}, \hat{Q}_{22}^{(1)} \in \mathbf{R}^{n \times n}.$$

In the case $\hat{s} = 1$, application of Lemma 3 to the matrix product $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}\hat{A}$ would lead to the following definition of orthogonal matrix $\begin{bmatrix} \hat{Q}_{11}^{(1)} & \hat{Q}_{12}^{(1)} \\ \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \end{bmatrix}$:

$$\begin{bmatrix} Q_{11}^{(1)} \\ Q_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(2)} & \hat{Q}_{12}^{(2)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{R}^{(2)} \end{bmatrix}, \ \hat{R}^{(2)}, \hat{Q}_{11}^{(2)}, \hat{Q}_{12}^{(2)}, \hat{Q}_{21}^{(2)}, \hat{Q}_{22}^{(2)} \in \mathbf{R}^{n \times n}, \tag{13}$$

$$\begin{bmatrix} \hat{Q}_{11}^{(2)} \\ \hat{A}^T\hat{Q}_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(1)} & \hat{Q}_{12}^{(1)} \\ \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{R}^{(1)} \end{bmatrix}, \ \hat{R}^{(1)}, \hat{Q}_{11}^{(1)}, \hat{Q}_{12}^{(1)}, \hat{Q}_{21}^{(1)}, \hat{Q}_{22}^{(1)} \in \mathbf{R}^{n \times n}. \tag{14}$$

Fortunately, one QR-factorization can be avoided by combining Eq. (13) with the computation in the preceding step: If $s_2 = -1$, we compute the QR factorization of $\begin{bmatrix} Q_{11}^{(2)} \\ A_2Q_{21}^{(2)} \end{bmatrix}$ to get

$$\begin{bmatrix} Q_{11}^{(2)} \\ A_2Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(2)} & \hat{Q}_{12}^{(2)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{\mathcal{R}}^{(2)} \end{bmatrix}, \ \hat{\mathcal{R}}^{(2)}, \hat{Q}_{11}^{(2)}, \hat{Q}_{12}^{(2)}, \hat{Q}_{21}^{(2)}, \hat{Q}_{22}^{(2)} \in \mathbf{R}^{n \times n}.$$

If $s_2 = 1$, then we compute the QR factorization of $\begin{bmatrix} Q_{11}^{(2)} \\ A_2^T Q_{21}^{(2)} \end{bmatrix}$ to get

$$\begin{bmatrix} Q_{11}^{(2)} \\ A_2^T Q_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} \hat{Q}_{11}^{(2)} & \hat{Q}_{12}^{(2)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} \hat{\mathcal{R}}^{(1)} \\ 0 \end{bmatrix}, \ \hat{\mathcal{R}}^{(1)}, \hat{Q}_{11}^{(2)}, \hat{Q}_{12}^{(2)}, \hat{Q}_{21}^{(2)}, \hat{Q}_{22}^{(2)} \in \mathbf{R}^{n \times n}.$$

For $m > 3$, the global reduction of A to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ can then be realized by first applying Lemma 2 to $A_{j-2}^{s_{j-2}} A_{j-1}^{s_{j-1}} A_j^{s_j}$, and subsequently taking into account the matrices A_{j-3}, A_{j-4}, ..., A_1 and A_{j+1}, A_{j+2}, ..., A_m. The overall algorithm is presented as Alg. 1. First we give two examples.

Example 1 *Let $A = A_1^{-1} A_2^{-1} A_3^{-1} A_4 A_5^{-1}$ in which $A_i \in \mathbf{R}^{n \times n}$, $i = 1, \cdots, 5$ and A_1, A_2, A_3 and A_5 are nonsingular. For $i = 1, \cdots, 4$, let orthogonal matrices $\begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix}$ with $Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n \times n}$ satisfy*

$$\begin{aligned}
\begin{bmatrix} A_4 \\ A_5 \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(4)} & Q_{12}^{(4)} \\ Q_{21}^{(4)} & Q_{22}^{(4)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(4)} \end{bmatrix}, \ R^{(4)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_3^T Q_{11}^{(4)} \\ Q_{21}^{(4)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(3)} & Q_{12}^{(3)} \\ Q_{21}^{(3)} & Q_{22}^{(3)} \end{bmatrix} \begin{bmatrix} R^{(3)} \\ 0 \end{bmatrix}, \ R^{(3)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_2^T Q_{11}^{(3)} \\ Q_{21}^{(3)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} R^{(2)} \\ 0 \end{bmatrix}, \ R^{(2)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_1^T Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(1)} \end{bmatrix}, \ R^{(1)} \in \mathbf{R}^{n \times n}.
\end{aligned}$$

Then we have that

$$A = Q_{11}^{(1)} (Q_{21}^{(1)})^{-1}.$$

Example 2 *Let $A = A_1^{-1} A_2 A_3^{-1} A_4^{-1} A_5 A_6^{-1}$ in which $A_i \in \mathbf{R}^{n \times n}$, $i = 1, \cdots, 6$ and A_1, A_3, A_4 and A_6 are nonsingular. For $i = 1, \cdots, 5$, let orthogonal matrices $\begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix}$ with $Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n \times n}$*

$$\begin{aligned}
\begin{bmatrix} A_5 \\ A_6 \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(5)} & Q_{12}^{(5)} \\ Q_{21}^{(5)} & \tilde{Q}_{22}^{(5)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(5)} \end{bmatrix}, \ R^{(5)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_4^T Q_{11}^{(5)} \\ Q_{21}^{(5)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(4)} & Q_{12}^{(4)} \\ Q_{21}^{(4)} & Q_{22}^{(4)} \end{bmatrix} \begin{bmatrix} R^{(4)} \\ 0 \end{bmatrix}, \ R^{(4)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_3^T Q_{11}^{(4)} \\ Q_{21}^{(4)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(3)} & Q_{12}^{(3)} \\ Q_{21}^{(3)} & Q_{22}^{(3)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(3)} \end{bmatrix}, \ R^{(3)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_2 Q_{11}^{(3)} \\ Q_{21}^{(3)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(2)} & Q_{12}^{(2)} \\ Q_{21}^{(2)} & Q_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(2)} \end{bmatrix}, \ R^{(2)} \in \mathbf{R}^{n \times n}, \\
\begin{bmatrix} A_1^T Q_{11}^{(2)} \\ Q_{21}^{(2)} \end{bmatrix} &= \begin{bmatrix} Q_{11}^{(1)} & Q_{12}^{(1)} \\ Q_{21}^{(1)} & Q_{22}^{(1)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(1)} \end{bmatrix}, \ R^{(1)} \in \mathbf{R}^{n \times n}.
\end{aligned}$$

Then we have that

$$A = Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}.$$

Algorithm 1
Input: *Matrix A of the form (2).*
Output: *Matrices $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}$ such that $(Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$ and $A = Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$.*
Init:

$$\begin{cases} \text{If all } s_i = 1, \text{ set } A_{m+1} := I, s_{m+1} := -1; m := m+1, \\ \text{If all } s_i = -1, \text{ set } A_{m+1} := A_m, A_m := I, s_m = 1, s_{m+1} := -1; m := m+1, \\ \text{If } -s_1 = \ldots = -s_j = s_{j+1} = \ldots = s_m = 1, \text{ apply procedure to } A^T. \end{cases}$$

Determine maximal j such that $s_j = 1$ and $s_{j+1} = -1$. Set $s_{m+1} := -1$, $s_0 = -s_{j+2}$, $Q_{11}^{(j+1)} = I$, $Q_{21}^{(j+1)} = A_{j+1}$.
Loop: for $i = j, j-1, \ldots, 1$, do:

- *Case $s_i = 1$ and $s_{i-1} = 1$. Compute the QR factorization of* $\begin{bmatrix} A_i Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix}$:

$$\begin{bmatrix} A_i Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} R^{(i)} \\ 0 \end{bmatrix}, \; R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n\times n}.$$

- *Case $s_i = 1$ and $s_{i-1} = -1$. Compute the QR factorization of* $\begin{bmatrix} A_i Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix}$:

$$\begin{bmatrix} A_i Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(i)} \end{bmatrix}, \; R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n\times n}.$$

- *Case $s_i = -1$ and $s_{i-1} = 1$. Compute the QR factorization of* $\begin{bmatrix} A_i^T Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix}$:

$$\begin{bmatrix} A_i^T Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(i)} \end{bmatrix}, \; R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n\times n}.$$

- *Case $s_i = -1$ and $s_{i-1} = -1$. Compute the QR factorization of* $\begin{bmatrix} A_i^T Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix}$:

$$\begin{bmatrix} A_i^T Q_{11}^{(i+1)} \\ Q_{21}^{(i+1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} R^{(i)} \\ 0 \end{bmatrix}, \; R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n\times n}.$$

End loop.
Set $Q_{11}^{(j+1)} := Q_{11}^{(1)}$ $Q_{21}^{(j+1)} := Q_{21}^{(1)}$. Loop: for $i = j+2, j+3, \ldots, m$ do:

- *Case $s_i = 1$ and $s_{i+1} = 1$. Compute the QR factorization of* $\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i^T Q_{21}^{(i-1)} \end{bmatrix}$:

$$\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i^T Q_{21}^{(i-1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} R^{(i)} \\ 0 \end{bmatrix}, \; R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n\times n}.$$

- *Case $s_i = 1$ and $s_{i+1} = -1$. Compute the QR factorization of* $\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i^T Q_{21}^{(i-1)} \end{bmatrix}$:

$$\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i^T Q_{21}^{(i-1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(i)} \end{bmatrix}, \ R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n \times n}.$$

- *Case $s_i = -1$ and $s_{i+1} = 1$. Compute the QR factorization of* $\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i Q_{21}^{(i-1)} \end{bmatrix}$:

$$\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i Q_{21}^{(i-1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} 0 \\ R^{(i)} \end{bmatrix}, \ R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n \times n}.$$

- *Case $s_i = -1$ and $s_{i+1} = -1$. Compute the QR factorization of* $\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i Q_{21}^{(i-1)} \end{bmatrix}$:

$$\begin{bmatrix} Q_{11}^{(i-1)} \\ A_i Q_{21}^{(i-1)} \end{bmatrix} = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \begin{bmatrix} R^{(i)} \\ 0 \end{bmatrix}, \ R^{(i)}, Q_{11}^{(i)}, Q_{12}^{(i)}, Q_{21}^{(i)}, Q_{22}^{(i)} \in \mathbf{R}^{n \times n}.$$

End loop.

Obviously, Algorithm 1 requires only $m-1$ QR factorizations if $s_{j-1} = 1$ and $s_j = -1$ for some $1 \le j \le m$, and requires m QR factorizations if all s_i, $i = 1, \cdots, m$, are the same.

In the following we explain that the computations involved in Algorithm 1 can be posed as left and right orthogonal transformations of a large matrix whose sub-blocks are the A_i or their transposes, several unit matrices, and the rest being zero matrices.

To have a good understanding of this reformulation, we first reconsider Examples 1 and 2.

- In Example 1, define

$$\mathcal{Q}_i := \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix}, \ i = 1, \cdots, 4,$$

$$\mathcal{M}_4 := \begin{bmatrix} A_4 \\ A_5 \end{bmatrix}^T, \ \mathcal{M}_3 := \begin{bmatrix} A_3 & 0 \\ 0 & I_n \end{bmatrix}, \ \mathcal{M}_2 := \begin{bmatrix} A_2 & 0 \\ 0 & I_n \end{bmatrix}, \ \mathcal{M}_1 := \begin{bmatrix} A_1 & 0 \\ 0 & I_n \end{bmatrix},$$

$$U := \begin{bmatrix} \mathcal{Q}_4^T & 0 \\ 0 & \mathcal{Q}_2^T \end{bmatrix}, \ V := \begin{bmatrix} I_n & 0 & 0 \\ 0 & \mathcal{Q}_3 & 0 \\ 0 & 0 & \mathcal{Q}_1 \end{bmatrix}, \ M := \begin{bmatrix} \mathcal{M}_4^T & \mathcal{M}_3 & 0 \\ 0 & \mathcal{M}_2^T & \mathcal{M}_1 \end{bmatrix}.$$

Then U, V are orthogonal and

$$UMV = R =: \begin{bmatrix} \mathcal{R}_4 & \mathcal{R}_3 & 0 \\ 0 & \mathcal{R}_2 & \mathcal{R}_1 \end{bmatrix},$$

where

$$\mathcal{R}_4 = \begin{bmatrix} R^{(4)} \\ 0 \end{bmatrix}, \ \mathcal{R}_3 = \begin{bmatrix} (R^{(3)})^T & 0 \\ \star & \star \end{bmatrix}, \ \mathcal{R}_2 = \begin{bmatrix} R^{(2)} & \star \\ 0 & \star \end{bmatrix}, \ \mathcal{R}_1 = \begin{bmatrix} 0 & (R^{(1)})^T \\ \star & \star \end{bmatrix}.$$

- In Example 2, define

$$\mathcal{Q}_i := \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix},\ i = 1, \cdots, 5,$$

$$\mathcal{M}_5 := \begin{bmatrix} A_5 \\ A_6 \end{bmatrix}^T,\ \mathcal{M}_4 := \begin{bmatrix} A_4 & 0 \\ 0 & I_n \end{bmatrix},$$

$$\mathcal{M}_3 := \begin{bmatrix} A_3 & 0 \\ 0 & I_n \end{bmatrix},\ \mathcal{M}_2 := \begin{bmatrix} A_2^T & 0 \\ 0 & I_n \end{bmatrix},\ \mathcal{M}_1 := \begin{bmatrix} A_1 & 0 \\ 0 & I_n \end{bmatrix},$$

$$U := \begin{bmatrix} \mathcal{Q}_5^T & 0 & 0 \\ 0 & \mathcal{Q}_3^T & 0 \\ 0 & 0 & \mathcal{Q}_1^T \end{bmatrix},\ V = \begin{bmatrix} I_n & 0 & 0 \\ 0 & \mathcal{Q}_4 & 0 \\ 0 & 0 & \mathcal{Q}_2 \end{bmatrix},\ M := \begin{bmatrix} \mathcal{M}_5^T & \mathcal{M}_4 & 0 \\ 0 & \mathcal{M}_3^T & \mathcal{M}_2 \\ 0 & 0 & \mathcal{M}_1^T \end{bmatrix}.$$

Then U, V are orthogonal and

$$UMV = R =: \begin{bmatrix} \mathcal{R}_5 & \mathcal{R}_4 & 0 \\ 0 & \mathcal{R}_3 & \mathcal{R}_2 \\ 0 & 0 & \mathcal{R}_1 \end{bmatrix},$$

where

$$\mathcal{R}_5 = \begin{bmatrix} 0 \\ R^{(5)} \end{bmatrix},\ \mathcal{R}_4 = \begin{bmatrix} (R^{(4)})^T & 0 \\ \star & \star \end{bmatrix},\ \mathcal{R}_3 = \begin{bmatrix} 0 & \star \\ R^{(3)} & \star \end{bmatrix},$$

$$\mathcal{R}_2 = \begin{bmatrix} 0 & (R^{(2)})^T \\ \star & \star \end{bmatrix},\ \mathcal{R}_1 = \begin{bmatrix} 0 & \star \\ R^{(1)} & \star \end{bmatrix}.$$

Now, we go back to Algorithm 1. For simplicity, we assume without loss of generality that $j = m - 1$ in Algorithm 1, i.e., $s_m = -1$ and $s_{m-1} = 1$.

Define

$$\mathcal{M}_{m-1} := \begin{bmatrix} A_{m-1} \\ A_m \end{bmatrix}^T,\quad \mathcal{Q}_i := \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix},\ i = 1, \cdots, m-1.$$

For $i = 1, \cdots, m-2$,

$$\mathcal{M}_i := \begin{bmatrix} A_i^T & 0 \\ 0 & I_n \end{bmatrix},\ \text{if } s_i = 1 \text{ and } s_{i-1} = 1,\ \text{or } s_i = 1 \text{ and } s_{i-1} = -1,$$

$$\mathcal{M}_i := \begin{bmatrix} A_i & 0 \\ 0 & I_n \end{bmatrix},\ \text{if } s_i = -1 \text{ and } s_{i-1} = 1,\ \text{or } s_i = -1 \text{ and } s_{i-1} = -1.$$

Set

$$U := \begin{bmatrix} \mathcal{Q}_{m-1}^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathcal{Q}_2^T \end{bmatrix} \text{ if } m \text{ is odd},\quad U := \begin{bmatrix} \mathcal{Q}_{m-1}^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathcal{Q}_1^T \end{bmatrix} \text{ if } m \text{ is even},$$

$$V := \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & \mathcal{Q}_{m-2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathcal{Q}_1 \end{bmatrix} \text{ if } m \text{ is odd},\quad V := \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & \mathcal{Q}_{m-2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathcal{Q}_2 \end{bmatrix} \text{ if } m \text{ is even},$$

$$M = \begin{bmatrix} \mathcal{M}_{m-1}^T & \mathcal{M}_{m-2} & 0 & 0 & 0 \\ 0 & \mathcal{M}_{m-3}^T & \mathcal{M}_{m-4} & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \mathcal{M}_2^T & \mathcal{M}_1 \end{bmatrix} \text{ if } m \text{ is odd,}$$

$$M = \begin{bmatrix} \mathcal{M}_{m-1}^T & \mathcal{M}_{m-2} & 0 & 0 & 0 \\ 0 & \mathcal{M}_{m-3}^T & \mathcal{M}_{m-4} & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \mathcal{M}_3^T & \mathcal{M}_2^T \\ 0 & 0 & 0 & 0 & \mathcal{M}_1^T \end{bmatrix} \text{ if } m \text{ is even.}$$

Then U and V are orthogonal matrices, and

$$UMV = R, \tag{15}$$

where

$$R = \begin{bmatrix} \mathcal{R}_{m-1} & \mathcal{R}_{m-2} & 0 & 0 & 0 \\ 0 & \mathcal{R}_{m-3} & \mathcal{R}_{m-4} & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \mathcal{R}_2 & \mathcal{R}_1 \end{bmatrix} \text{ if } m \text{ is odd,}$$

$$R = \begin{bmatrix} \mathcal{R}_{m-1} & \mathcal{R}_{m-2} & 0 & 0 & 0 \\ 0 & \mathcal{R}_{m-3} & \mathcal{R}_{m-4} & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \mathcal{R}_3 & \mathcal{R}_2 \\ 0 & 0 & 0 & 0 & \mathcal{R}_1 \end{bmatrix} \text{ if } m \text{ is even,}$$

$$\mathcal{R}_{m-1} = \begin{bmatrix} R^{(m-1)} \\ 0 \end{bmatrix} \text{ or } \mathcal{R}_{m-1} = \begin{bmatrix} 0 \\ R^{(m-1)} \end{bmatrix},$$

$\mathcal{R}_i \in \mathbf{R}^{n\times n}$, $i = 1, \cdots, m-2$, are of one of the following forms

$$\begin{bmatrix} R^{(i)} & \star \\ 0 & \star \end{bmatrix}, \begin{bmatrix} R^{(i)} & 0 \\ \star & \star \end{bmatrix}, \begin{bmatrix} \star & \star \\ 0 & R^{(i)} \end{bmatrix}, \begin{bmatrix} \star & 0 \\ \star & R^{(i)} \end{bmatrix},$$

$R^{(i)} \in \mathbf{R}^{n\times n}$ $(i = 1, \cdots, m-1)$ are nonsingular.

Let's use $\bar{X}$ denote the computed X using finite precision arithmetic, as opposed to exact arithmetic, and ϵ denote the machine precision. From (15), we have [10]

$$\|\bar{U}^T\bar{U} - I_{2n}\| \approx \epsilon, \quad \|\bar{V}^T\bar{V} - I_{2n}\| \approx \epsilon, \quad \|\bar{U}M\bar{V} - \bar{R}\| \approx \epsilon\|M\|. \tag{16}$$

Hence, algorithm 1 is backward stable in the sense that (16) holds.

One advantage of Algorithm 1 is that the explicit computation of A_i^{-1} and explicit solution of the corresponding triangular linear system are avoided if $s_i = -1$, an example supporting this issue is given in Appendix.

After reducing A to $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ by Algorithm 1, we can compute the SVD of A by computing the CSD of $Q_{11}^{(1)}$ and $Q_{21}^{(1)}$ by Matlab command *gsvd*.

3 Numerical Experiments

In this section we illustrate the performance of our method by means of some numerical examples. After reducing the matrix $A \in \mathbf{R}^{n\times n}$ in Eq.(2) to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$ by Algorithm 1, the SVD of A is obtained by computing the CSD of $Q_{11}^{(1)}$ and $Q_{21}^{(1)}$ using Van Loan's CSD algorithm [18] (i.e. Matlab command *gsvd*); here we set the parameter $\tau = \frac{1}{\sqrt{2}}$, which minimizes a backward error bound [3]. All our calculations were carried out in MATLAB 5 on a Sun Ultra 5 workstation with IEEE standard (machine accuracy $\epsilon \cong 10^{-16}$).

To quantify the accuracy of the results, we define the residual

$$resSVD = \frac{\|\hat{\Sigma} - \Sigma\|_2}{n\|\hat{\Sigma}\|_2}, \tag{17}$$

in which

$$\begin{aligned} \hat{\Sigma} &= \mathrm{diag}\{\hat{\sigma}_1, \hat{\sigma}_2, \cdots, \hat{\sigma}_n\}, \quad \hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_n > 0, \\ \Sigma &= \mathrm{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_n\}, \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0, \end{aligned}$$

where $\hat{\sigma}_i$ and σ_i, $i = 1, 2, \cdots, n$, are the computed and the exact singular values of A respectively. We use the symbol η to denote a value drawn from a uniform distribution over $[0, 1)$.

Example 3 *In this example, we test our method using some common-used matrix families including Lauchli matrices, Frank matrices, Cauchy matrices, Vandermonde matrices, Wilkinson matrices, Pascal matrices, Toeplitz matrices and Hankel matrices. Take*

$$m = 6, \ s_1 = s_3 = s_4 = s_6 = -1, \ s_2 = s_5 = 1.$$

1) Matrices A_i $(i = 1, \cdots, 6)$ are generalized using the following Matlab commands:

$$\begin{aligned} &X = \mathrm{gallery}('\mathrm{lauchli}', n, 0.1); \ A_1 = X(1:n, 1:n); \\ &A_3 = \mathrm{gallery}('\mathrm{frank}', n, 1); \ A_6 = \mathrm{vander}(\mathrm{randn}(n, 1)); \\ &A_4 = \mathrm{gallery}('\mathrm{cauchy}', \mathrm{randn}(n, 1), \mathrm{randn}(n, 1)); \\ &A_2 = A_1 * A_3; \ A_5 = A_4. \end{aligned}$$

It is easy to see that $A = A_6^{-1}$ and hence we can get the singular values of A by Matlab command svd(inv(A_6)). The computed results are listed in Table 1, in which and in the rest of this section, $\kappa(A_i)$ denotes the 2-norm condition number of the A_i.

n	$resSVD$	$\kappa(A_1)$	$\kappa(A_2)$	$\kappa(A_3)$	$\kappa(A_4)$	$\kappa(A_6)$
6	5.5×10^{-15}	60.1	9.7×10^4	4.2×10^3	1.7×10^3	1.4×10^3
7	8.2×10^{-14}	70.1	7.9×10^5	3.2×10^4	1.9×10^5	2.4×10^5
8	4.2×10^{-13}	80.1	7.3×10^6	2.8×10^5	3.2×10^5	1.9×10^3
9	8.6×10^{-11}	90.1	7.5×10^7	2.7×10^6	2.8×10^7	3.8×10^4
10	7.4×10^{-11}	1.0×10^2	8.4×10^8	2.9×10^7	1.3×10^4	3.6×10^5

Table1

2) Matrices A_i $(i = 1, \cdots, 6)$ are generalized using the following Matlab commands:

$$A_1 = \text{wilkinson}(n);\ A_4 = \text{pascal}(n);$$
$$c = \text{randn}(n, 1);\ r = \text{randn}(1, n);\ r(1, 1) = c(1, 1);\ A_3 = \text{toeplitz}(c, r);$$
$$d = \text{rand}(n, 1);\ A_6 = \text{hankel}(d);$$
$$A_2 = A_3;\ A_5 = A_4 * A_6.$$

Obviously, $A = A_1^{-1}$ and we can get the singular values of A by Matlab command svd(inv(A_1)). *The computed results are listed in the Table 2.*

n	$resSVD$	$\kappa(A_1)$	$\kappa(A_3)$	$\kappa(A_4)$	$\kappa(A_5)$	$\kappa(A_6)$
6	1.7×10^{-14}	13.1	29.7	1.1×10^{5}	1.3×10^{5}	7.2
7	3.1×10^{-13}	14.1	29.3	1.5×10^{6}	2.3×10^{6}	1.8×10^{3}
8	9.4×10^{-11}	19.7	7.8	2.1×10^{7}	1.8×10^{11}	1.5×10^{6}
9	7.7×10^{-12}	18.6	81.8	2.9×10^{8}	1.4×10^{10}	5.3×10^{2}
10	6.3×10^{-11}	24.7	34.4	4.2×10^{9}	5.2×10^{11}	1.0×10^{5}

Table2

Example 4 *We compare our Algorithm 1 with the method in [11] in this example. Let*

$$A_1 = \text{pascal}(n),\ A_2 = A_1,\ s_1 = -1,\ s_2 = 1.$$

Obviously, $A = I_n$. The computed $resSVD$ are listed in Table 3.

n	13	14	15	16	17	18
$resSVD_1$	2.6×10^{-7}	2.9×10^{-6}	3.1×10^{-5}	3.4×10^{-4}	0.002	0.02
$resSVD_2$	1.7×10^{-6}	4.5×10^{-5}	3.5×10^{-4}	8.4×10^{-4}	0.02	1.7
$resSVD_3$	2.0×10^{-6}	4.2×10^{-5}	2.6×10^{-4}	8.7×10^{-3}	0.07	1.14
$\kappa(A_1)$	1.3×10^{13}	1.9×10^{14}	2.8×10^{15}	4.2×10^{16}	6.4×10^{17}	9.4×10^{18}

Table3

In Table 3, $resSVD_1$, $resSVD_2$ and $resSVD_3$ denote the $resSVD$ computed by our Algorithm 1, the method in [11], and the command svd(inv(A_1) $*$ A_2), *respectively.*

In the next 6 examples, $m = 6$, $s_1 = s_3 = s_4 = s_6 = -1$ and $s_2 = s_5 = 1$. we have

$$\begin{aligned} A_1 &= U_1\Sigma_1V_1, \quad A_2 = U_1\Sigma_2V_2, \quad A_3 = \Sigma_3V_2, \\ A_4 &= U_4\Sigma_4, \quad A_5 = U_4\Sigma_5V_5, \quad A_6 = U_6\Sigma_6V_5, \end{aligned}$$

in which U_1, U_4, U_6, V_1, V_2 and V_5 are randomly chosen orthogonal matrices; the entries of $\Sigma_i = \text{diag}\{\sigma_{i1}, \cdots, \sigma_{in}\}$, $i = 1, \cdots, 6$, are specified below. For values of $n = 35, 36, \cdots, 80$, the obtained residuals $resSVD$ are plotted in Figure 2.

Example 5 *For* $j = 1, \cdots, n$,

$$\sigma_{ij} = (j + \eta_{ij}) * i, \quad i = 1, \cdots, 4, 6,$$

and

$$\sigma_{5j} = (j + \eta_{5j}) * 5 * 10^{-\frac{7}{n} * j}.$$

Example 6 *For* $j = 1, \cdots, n$,

$$\sigma_{ij} = (j + \eta_{ij}) * i, \quad i = 1, \cdots, 4,$$

and

$$\sigma_{5j} = (j + \eta_{5j}) * 5 * 10^{-\frac{7}{n} * j}, \quad \sigma_{6j} = (j + \eta_{6j}) * 6 * 10^{-\frac{6}{n} * j}.$$

Example 7 *For* $j = 1, \cdots, n$,

$$\sigma_{ij} = (j + \eta_{ij}) * i, \quad i = 1, \cdots, 3,$$

and

$$\begin{aligned} \sigma_{ij} &= (j + \eta_{ij}) * i, \quad i = 1, \cdots, 3, \quad \sigma_{4j} = (j + \eta_{4j}) * 4 * 10^{-\frac{3}{n} * j}, \\ \sigma_{5j} &= (j + \eta_{5j}) * 5 * 10^{-\frac{8}{n} * j}, \quad \sigma_{6j} = (j + \eta_{6j}) * 6 * 10^{-\frac{5}{n} * j}. \end{aligned}$$

Example 8 *For* $j = 1, \cdots, n$,

$$\sigma_{ij} = (j + \eta_{ij}) * i, \quad i = 1, 2,$$

and

$$\begin{aligned} \sigma_{3j} &= (j + \eta_{3j}) * 3 * 10^{\frac{3}{n} * j}, \quad \sigma_{4j} = (j + \eta_{4j}) * 4 * 10^{-\frac{3}{n} * j}, \\ \sigma_{5j} &= (j + \eta_{5j}) * 5 * 10^{-\frac{8}{n} * j}, \quad \sigma_{6j} = (j + \eta_{6j}) * 6 * 10^{-\frac{5}{n} * j}. \end{aligned}$$

Example 9 *For* $j = 1, \cdots, n$,

$$\begin{aligned} \sigma_{1j} &= (j + \eta_{1j}), \quad \sigma_{2j} = (j + \eta_{2j}) * 2 * 10^{-\frac{4}{n} * j}, \\ \sigma_{3j} &= (j + \eta_{3j}) * 3 * 10^{\frac{3}{n} * j}, \quad \sigma_{4j} = (j + \eta_{4j}) * 4 * 10^{-\frac{3}{n} * j}, \\ \sigma_{5j} &= (j + \eta_{5j}) * 5 * 10^{-\frac{7}{n} * j}, \quad \sigma_{6j} = (j + \eta_{6j}) * 6 * 10^{-\frac{4}{n} * j}. \end{aligned}$$

Example 10

$$\begin{aligned} \sigma_{1j} &= (j + \eta_{1j}) * 10^{-\frac{3}{n} * j}, \quad \sigma_{2j} = (j + \eta_{2j}) * 2 * 10^{-\frac{4}{n} * j}, \\ \sigma_{3j} &= (j + \eta_{3j}) * 3 * 10^{\frac{3}{n} * j}, \quad \sigma_{4j} = (j + \eta_{4j}) * 4 * 10^{-\frac{3}{n} * j}, \\ \sigma_{5j} &= (j + \eta_{5j}) * 5 * 10^{-\frac{7}{n} * j}, \quad \sigma_{6j} = (j + \eta_{6j}) * 6 * 10^{-\frac{4}{n} * j}. \end{aligned}$$

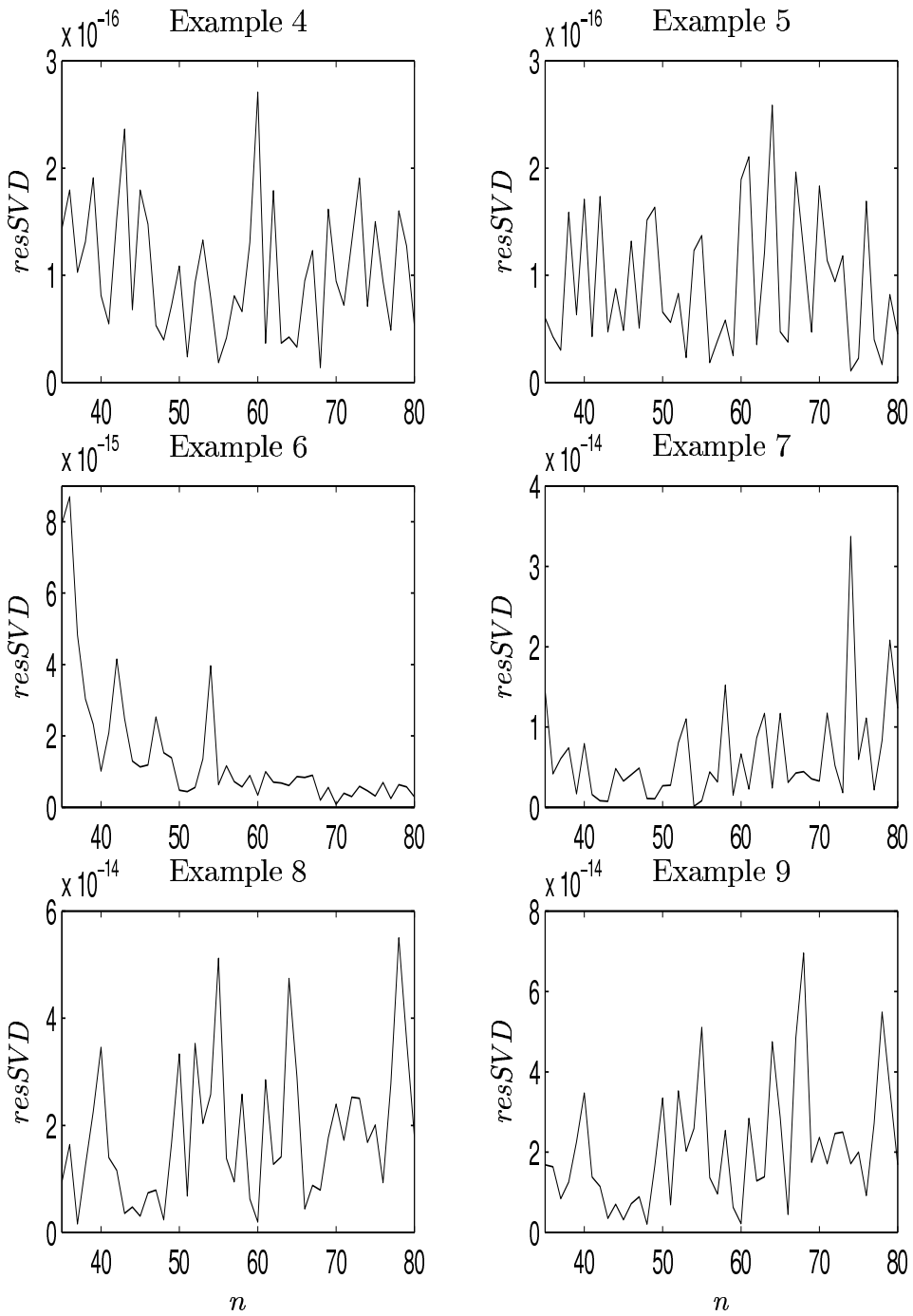

Figure 1: Residuals $resSVD$ (see Eq. (17)) as a function of matrix size n, computed in Examples 7, ..., 12.

Obviously, the exact of SVDs of A in Examples 5–10 are

$$A = V_1 \Sigma U_6^T \text{ with } \Sigma = \Sigma_1^{-1}\Sigma_2\Sigma_3^{-1}\Sigma_4^{-1}\Sigma_5\Sigma_6^{-1}.$$

In examples 5–10, we choose the diagonal elements of Σ_i in such a way that different situations in terms of the condition number of A_i and A are to be dealt with. In detail,

- In Example 5, A_1, A_2, A_3, A_4 and A_6 are well-conditioned, A_5 and A are ill-conditioned with $\kappa(A_5) = (10^7)$ and $\kappa(A) = O(10^7)$.
- In Example 6, A_1, A_2, A_3, A_4 and A are well-conditioned, A_5 and A_6 are ill-conditioned with $\kappa(A_5) = O(10^7)$ and $\kappa(A_6) = O(10^6)$.
- In Example 7, A_1, A_2, A_3 and A are well-conditioned, A_4 has a moderate condition number, A_5 and A_6 are ill-conditioned with $\kappa(A_5) = O(10^8)$ and $\kappa(A_6) = (10^5)$.
- In Example 8, A_1 and A_2 are well-conditioned, A_3, A_4 and A have a moderate condition number, A_5 and A_6 are ill-conditioned with $\kappa(A_5) = O(10^8)$ and $\kappa(A_6) = O(10^5)$.
- In Example 9, A_1 is well-conditioned, A_2, A_3, A_4 and A_6 have a moderate condition number, A_5 and A are ill-conditioned with $\kappa(A_5) = O(10^7)$ and $\kappa(A) = O(10^7)$.
- In Example 10, A_1, A_2, A_3, A_4, A_6 and A have a moderate condition number, A_5 is ill-conditioned.

From Tables 1, 2 and 3, and Figure 1, we see that our results are satisfactory.

4 Conclusions

In this paper, we have studied the computation of the SVD of a general matrix product/quotient sequence. First we reduced the sequence by at most m QR-factorizations to the form $Q_{11}^{(1)}(Q_{21}^{(1)})^{-1}$, with $Q_{11}^{(1)}, Q_{21}^{(1)} \in \mathbf{R}^{n\times n}$ and $(Q_{11}^{(1)})^T Q_{11}^{(1)} + (Q_{21}^{(1)})^T Q_{21}^{(1)} = I$. Then we obtain the SVD of A by computing the CSD of $Q_{11}^{(1)}$ and $Q_{21}^{(1)}$ using Matlab command *gsvd*. An advantage of our QR-type reduction is that we avoid to explicitly compute A_i^{-1} when $s_i = -1$ in Eq. (2). Some numerical examples were given to show the performance of the presented method.

Note that in Algorithm 1 s_0 allows to take into account the type of operation required for $i = j + 2$. It is now indeed natural to ask which j we have to choose to star, if there is more than one possibility of $s_j = 1$ and $s_{j+1} = -1$. Also, in which order should individual moves to the left and moves to the right be combined? It seems clear anyway that there is no easy way to fix the order of the operations in an optimal way in advance.

For the theory we developed in this paper, we only need assume that A_j is nonsingular if $s_j = -1$. But, Algorithm 1 can work and (15) is still true in all cases. Hence, we may ask the question: What is the minimum requirement on all A_j $(j = 1, \cdots, m)$ so that our theory still valid and our results for nonsquare matrices meaningful if they have conformable dimensions?

The above two questions are still open and definitely worthy of further investigation!

Acknowledgement

We are very grateful to referees for their kind and invaluable comments and suggestions which have improved the present paper significantly.

Appendix

Take

$$A_1 = \begin{bmatrix} 0.99999999987468 & 0.50000000236877 & 0.33333332305860 & 0.25000001557339 & 0.19999999236251 \\ 0.50000000236877 & 0.33333328856096 & 0.25000019420358 & 0.19999970564605 & 0.16666681102359 \\ 0.33333332305860 & 0.25000019420358 & 0.19999915762723 & 0.16666794344931 & 0.14285651669800 \\ 0.25000001557339 & 0.19999970564605 & 0.16666794344931 & 0.14285520764038 & 0.12500094906810 \\ 0.19999999236251 & 0.16666681102359 & 0.14285651669800 & 0.12500094906810 & 0.11111064566958 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} \sqrt{5} & \sqrt{5} & \sqrt{5} & \sqrt{5} & \sqrt{5} \end{bmatrix} A_1.$$

We compute the SVD of $A_2A_1^{-1}$ by the method in [11] and our Algorithm 1.

First, we compute QR factorization of A_1: $A_1 = QR$. Then compute the solution x of $xR = A_2$. So, the SVD of the $A_2A_1^{-1}$ given by the method in [11] is 5.00120891819209.

Now we use our Algorithm 1 to get that the SVD of $A_1^{-1}b$ is 5.00000781367755.

The exact value of the SVD of $A_2A_1^{-1}$ is 5. Hence, the computed SVD of our Algorithm 1 is much better! This comparison is easy to understand. The exact value of x should satisfy

$$xQ^T = \begin{bmatrix} \sqrt{5} & \sqrt{5} & \sqrt{5} & \sqrt{5} & \sqrt{5} \end{bmatrix}^T.$$

But, the computed x only gives

$$xQ^T = \begin{bmatrix} 2.23668506012794 & 2.22440444340721 & 2.28665944307996 & 2.15938659945574 & 2.27367402298989 \end{bmatrix}.$$

Therefore, the computed x is not acceptable. Since

$$\|Q^TQ - I_5\| = 6.106109938885852 \times 10^{-16},\ \|A_1 - QR\| = 1.426242624291993 \times 10^{-16}.$$

So, the computed Q and R are very accurate. Note that A_1 is ill-conditioned but well-balanced, however,

$$R = \begin{bmatrix} 1.20979796293064 & -0.68882024839475 & -0.49201446313907 & -0.38541132945901 & -0.31775934077732 \\ 0 & -0.13005981043404 & -0.14019190378167 & -0.13270023604854 & -0.12232288892133 \\ 0 & 0 & -0.00806537832143 & -0.01263254593415 & -0.01490828704240 \\ 0 & 0 & 0 & -0.00033807875859 & -0.00068936642389 \\ 0 & 0 & 0 & 0 & 0.00000000000001 \end{bmatrix},$$

thus, the computed R is not only ill-conditioned but also highly *non-balanced*, which leads to catastrophic cancelation in the solution of triangular linear system $xR = A_2$. Such catastrophic

cancelation affects seriously the computational accuracy. The situation is the same even the QR factorization of A_1 with *pivoting* is used. This example indicates that the explicit computation of the inverses of triangular matrices from the QR factorizations and the explicit solutions of corresponding triangular linear systems should be avoided in the computation of the SVD of matrix product/quotient, if this is possible.

References

[1] E. Anderson, Z. Bai, C. Bischof, J.W. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1992.

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

[3] Z. Bai, The CSD, GSVD, their applications and computations, *IMA Preprint 958, University of Minnesota,* April 1992, Submitted to *SIAM Rev.*

[4] D. Chu, L. De Lathauwer and B. De Moor. On the computation of Restricted Singular Value Decomposition via Cosine-Sine Decomposition. *SIAM J. Matrix Anal. Appl.*, 22:580–601, 2001.

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

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

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

[8] B. De Moor and G. H. Golub, Generalized singular value decompositions: a proposal for a standardized nomenclature, *Internal Report 89-10, ESAT-SISTA, K.U.Leuven*, Leuven, Belgium, 1989.

[9] K. Fernando and S. Hammarling, A product induced singular value decomposition for two matrices and balanced realisation, *NAG Technical Report TR8/87.*

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

[11] G. Golub, K. Solna and P. Van Dooren, Computing the SVD of a general matrix product/quotient, *SIAM J. Matrix Anal. Appl.* , to appear.

[12] L. Arnold, H. Crauel and J. Eckmann. *Lyapunov Exponents*, Spring-Verlag, Berlin, New York, 1991.

[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.*, 7:1147–1159, 1986.

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

[15] C.C. Paige, Computing the generalized singular value decomposition, *SIAM J. Sci. Stat. Comput.*, 7:1126–1146, 1986.

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

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

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

[19] H. Zha, A numerical algorithm for computing the restricted singular value decomposition of matrix triplets, *Linear Algebra Appl.*, 168:1–25, 1992.

[20] H. Zha, The restricted singular value decomposition of matrix triplets. *SIAM J. Matrix Anal. Appl.*, 12:172–194, 1991.

[21] H. Zha, Computing the generalized singular values/vectors of large sparse or structured matrix pairs, *Numerische Mathematik*, 72:391–417, 1996.