MATHEMATICS OF COMPUTATION
https://doi.org/10.1090/mcom/3444
Article electronically published on April 25, 2019

A QD-TYPE METHOD FOR COMPUTING GENERALIZED SINGULAR VALUES OF BF MATRIX PAIRS WITH SIGN REGULARITY TO HIGH RELATIVE ACCURACY

RONG HUANG

Abstract. Structured matrices such as Vandermonde and Cauchy matrices frequently appear in various areas of modern computing, and they tend to be badly ill-conditioned, but a desirable property is that they admit accurate bidiagonal factorizations (BFs). In this paper, we propose a qd-type method to compute the generalized singular values of BF matrix pairs. A mechanism involving sign regularity of BF generators is provided to guarantee that there is no subtraction of like-signed numbers for the qd-type method. Consequently, all the generalized singular values are computed to high relative accuracy, independent of any conventional condition number. Error analysis and numerical experiments are presented to confirm the high relative accuracy.

1. Introduction

The study of high-accuracy computations is an active research topic of great interest in recent years. For the singular value problem of a single matrix, high-accuracy algorithms have been constructed only for a few classes of matrices such as diagonally dominant matrices [14, 15, 45], bidiagonal matrices [12], acyclic matrices [10], certain sign regular matrices [25–27], and matrices for which a rank-revealing decomposition can be accurately computed [8, 9, 11]. In particular, qd-type algorithms have been developed to compute singular values with high relative accuracy [18, 27, 37]. However, there are many situations in which one needs to find the generalized singular values (GSVs) of a matrix pair, which were introduced by Van Loan [43] and further developed by Paige and Saunders [36]: the set of GSVs of a matrix pair $(A, B) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{m\times p}$ is defined as

$$\sigma(A, B) = \{\sigma \geq 0 : \ A^T A - \sigma^2 B^T B \text{ is singular}\}.$$

If B is nonsingular, then the GSVs of (A, B) are the singular values of AB^{-1}. Indeed, the GSV problem arises in many applications such as DNA microarray analysis [1], weighted least squares [5], constraint least squares [21], information retrieval [22], and linear discriminant analysis [38]. Moreover, a variety of standard algorithms such as Jacobi-type algorithms [3, 35], CS algorithms [42, 44], and QR-type algorithms [6, 20] have been developed for the GSV problem, and a common

Received by the editor May 24, 2018, and, in revised form, January 7, 2019, and February 25, 2019.

2010 *Mathematics Subject Classification.* Primary 65F15, 15A18, 15A23.

Key words and phrases. Structured matrices, generalized singular values, qd-type methods, bidiagonal factorizations, high relative accuracy.

The author was supported by the National Natural Science Foundation of China (Grant Nos. 11871020 and 11471279) and the Natural Science Foundation for Distinguished Young Scholars of Hunan Province (Grant No. 2017JJ1025).

feature of these standard algorithms is backward stability. Unfortunately, many structured matrices arising in applications are very ill-conditioned, so standard stable algorithms fail to compute the tiny GSVs with correct digits. Our interest of this paper is to accurately compute all the GSVs of structured matrix pairs.

Structured matrices such as Vandermonde and Cauchy matrices frequently appear in various areas of modern computing [33, 39, 40], and they tend to be badly ill-conditioned [34], but a desirable property is that they admit certain bidiagonal-factorization (BF) forms, which will be fully exploited in our computations later. Before proceeding, we recall Neville elimination, which is a very effective method to derive BF forms. Roughly, Neville elimination is a classical elimination method for the triangularization of a matrix which, different from the standard Gaussian elimination, produces zeros in a column by subtracting from each row an adequate multiple of the previous one [19]. For example, given $A \in \mathbb{R}^{n\times p}$, this elimination procedure consists of at most $t = \min\{n, p\}$ successive steps

$$A = A^{(0)} \to A^{(1)} \to \cdots \to A^{(t)} = U,$$

where $U \in \mathbb{R}^{n\times p}$ is upper triangular, and for all $1 \le k \le t$, $A^{(k)} = (a_{ij}^{(k)}) \in \mathbb{R}^{n\times p}$ is obtained from $A^{(k-1)} = (a_{ij}^{(k-1)}) \in \mathbb{R}^{n\times p}$ as follows:

$$L_k A^{(k-1)} = A^{(k)},$$

where $L_k = E_{k+1}(-\alpha_{k+1,k}) \cdots E_n(-\alpha_{nk})$ with $\alpha_{ik} =: a_{ik}^{(k-1)}/a_{i-1,k}^{(k-1)}$ $(k+1 \le i \le n)$. Here, denote by $E_i(x) \in \mathbb{R}^{n\times n}$ $(2 \le i \le n)$ the matrix obtained from the identity matrix $I_n \in \mathbb{R}^{n\times n}$ by replacing its $(i, i-1)$th entry with x. A complete Neville elimination consists of reducing A into an upper triangular matrix U and then converting U^T into a diagonal form. In the sequel,

$$\begin{cases} X_i = \textbf{bilow}(d_1, \ldots, d_n; x_{i+1}, \ldots, x_n) \in \mathbb{R}^{n\times n}, \\ Y_i = \textbf{biupp}(d_1, \ldots, d_n; y_{i+1}, \ldots, y_n) \in \mathbb{R}^{n\times n}, \end{cases}$$

where X_i (or Y_i) is lower bidiagonal (or upper bidiagonal) with diagonal entries d_k $(1 \le k \le n)$, and $(k, k-1)$th (or $(k-1, k)$th) entries x_k (or y_k) for $i+1 \le k \le n$ and 0 otherwise; and set the convention $X_i = Y_i = \mathrm{diag}(d_k) \in \mathbb{R}^{n\times n}$ if $i \ge n$. Therefore, when the complete Neville elimination is successfully performed to reduce A into a diagonal matrix, then A has a BF form defined as follows.

Definition 1.1. A matrix $A \in \mathbb{R}^{n\times p}$ has a BF form if it can be factorized as

$$A = L_1^{-1} \cdots L_t^{-1} D U_t^{-1} \cdots U_1^{-1}, \quad t = \min\{n, p\}, \tag{1.1}$$

where

$$\begin{cases} D = \mathrm{diag}(\alpha_{kk}) \in \mathbb{R}^{n\times p}, \\ L_i = \textbf{bilow}(1, \ldots, 1; -\alpha_{i+1,i}, \ldots, -\alpha_{ni}) \in \mathbb{R}^{n\times n}, \qquad \forall\ 1 \le i \le t, \\ U_i = \textbf{biupp}(1, \ldots, 1; -\alpha_{i,i+1}, \ldots, -\alpha_{ip}) \in \mathbb{R}^{p\times p}, \end{cases} \tag{1.2}$$

with the convention $L_i = I_n$ if $i \ge n$, and $U_i = I_p$ if $i \ge p$.

Now let us store the nontrivial elements α_{ij} in the factors of (1.2) as the generator matrix $G = (\alpha_{ij}) \in \mathbb{R}^{n\times p}$, and denote by $A =: \mathcal{BF}(\alpha_{ij}) \in \mathbb{R}^{n\times p}$ the BF matrix generated by these generators in the procedure (1.1). It must be pointed out that by virtue of Neville elimination, we have the following generator formula:

$$\alpha_{ij} = \begin{cases} \frac{\det A[1:i]}{\det A[1:i-1]}, \ i=j, \\ \frac{\det A[i-j+1:i|1:j]}{\det A[i-j+1:i-1|1:j-1]} \cdot \frac{\det A[i-j:i-2|1:j-1]}{\det A[i-j:i-1|1:j]}, \ i>j, \\ \frac{\det A[1:i|j-i+1:j]}{\det A[1:i-1|j-i+1:j-1]} \cdot \frac{\det A[1:i-1|j-i:j-2]}{\det A[1:i|j-i:j-1]}, \ j>i, \end{cases}$$

provided that the involved minors above are nonzero. Here, denote by $A[i:j|k:l]$ the submatrix of $A \in \mathbb{R}^{n\times q}$ having row and column indexes in the ranges i through j and k through l, respectively; and $A[i:j|i:j]$ is abbreviated as $A[i:j]$. So far, it has been found out that many well-known structured matrices belong to the class of BF matrices according to the available factorizations in the literature [7, 9, 13, 27–32]. For example, a Vandermonde matrix $A = [x_i^{j-1}]_{i,j=1}^{n,p}$ with distinct nodes x_i $(1 \le i \le n)$ is a BF matrix whose generators are the following:

$$\alpha_{ij} = \begin{cases} \prod_{k=1}^{i-1}(x_i - x_k), \ \text{if } i=j, \\ \prod_{k=i-j+1}^{i-1} \frac{x_i - x_k}{x_{i-1}-x_{k-1}}, \ \text{if } i>j, \\ x_i, \ \text{if } i<j, \end{cases} \tag{1.3}$$

and a Cauchy matrix $A = \left[\frac{1}{x_i+y_j}\right]_{i,j=1}^{n,p}$ with distinct nodes x_i and y_j is a BF matrix whose generators are the following:

$$\alpha_{ij} = \begin{cases} \frac{1}{x_i+y_i} \cdot \prod_{k=1}^{i-1} \frac{(x_i-x_k)(y_i-y_k)}{(x_i+y_k)(y_i+x_k)}, \ \text{if } i=j, \\ \frac{x_{i-j}+y_j}{x_i+y_j} \prod_{k=i-j+1}^{i-1} \frac{x_i-x_k}{x_{i-1}-x_{k-1}} \prod_{k=1}^{j-1} \frac{x_{i-1}+y_k}{x_i+y_k}, \ \text{if } i>j, \\ \frac{y_{j-i}+x_i}{y_j+x_i} \prod_{k=j-i+1}^{j-1} \frac{y_j-y_k}{y_{j-1}-y_{k-1}} \prod_{k=1}^{i-1} \frac{y_{j-1}+x_k}{y_j+x_k}, \ \text{if } i<j. \end{cases} \tag{1.4}$$

In [24], high-accuracy algorithms have been proposed for the eigenvalue problem of consecutive-rank-descending matrices, and they belong to the class of BF matrices. Besides these mentioned matrices, other structured matrices admitting BF forms include Cauchy-Vandermonde matrices [30], Bernstein-Vandermonde matrices [28], Hilbert and Pascal matrices [27], quasi-Cauchy matrices [9], q-Bernstein-Vandermonde matrices [7], generalized Vandermonde matrices [13], Jacobi-Stirling matrices [31], Said-Ball-Vandermonde matrices [29], Lupaş matrices [32], and rank-structured matrices [23]. In a word, this class under consideration is a wide class of matrices containing many well-known structured matrices.

More interestingly, for various types of BF matrices $A \in \mathbb{R}^{n\times p}$, the generator matrices $G = (\alpha_{ij}) \in \mathbb{R}^{n\times p}$ often possess an amazing sign regularity as follows:

$$\begin{cases} \operatorname{sign}(\alpha_{ii}) = s_i, \ \forall\ 1 \le i \le t = \min\{n,p\}, \\ \operatorname{sign}(\alpha_{i1}, \ldots, \alpha_{i,\min\{i-1,p\}}) = r_i, \ \forall\ 2 \le i \le n, \\ \operatorname{sign}(\alpha_{1i}, \ldots, \alpha_{\min\{i-1,n\},i}) = c_i, \ \forall\ 2 \le i \le p, \\ r_i c_i = s_{i-1}s_i \text{ or } 0, \ \forall\ 2 \le i \le t, \end{cases} \tag{1.5}$$

and we say that A has the *sign sequence* $(s_1, \ldots, s_t; r_2, \ldots, r_n; c_2, \ldots, c_p)$ (here, we mean by $\operatorname{sign}(\alpha) = 1$ or -1 that all the nonzero entries of a vector $\alpha \in \mathbb{R}^n$ are positive or negative, respectively; and $\operatorname{sign}(\alpha) = 0$ if the vector $\alpha = 0$). A remarkable example is that the wide class of totally positive matrices, which has been extensively studied and arisen in approximation theory, computer aided geometric design, statistics, and other fields [2, 4, 11, 27, 41], must satisfy the sign regularity

(1.5) with the sign sequence $(1,\dots,1;1,\dots,1;1,\dots,1)$ [19]. When considering the subclasses of Vandermonde matrices by the positive or negative property of nodes,

$$\begin{cases}\mathcal{V}_{n,p}^{+}=\{A|\ A=[x_i^{j-1}]_{i,j=1}^{n,p},\ \text{where } x_i>0,\ \forall\ i\},\\ \mathcal{V}_{n,p}^{-}=\{A|\ A=[x_i^{j-1}]_{i,j=1}^{n,p},\ \text{where } x_i<0,\ \forall\ i\},\end{cases}$$

for any $A\in\mathcal{V}_{n,p}^{+}$ or $A\in\mathcal{V}_{n,p}^{-}$, there exists a permutation matrix P such that the nodes of $B=PA$ are ascending-ordered or descending-ordered whose generators (1.3) satisfy the sign regularity (1.5) with sign sequence $(1,\dots,1;1,\dots,1;1,\dots,1)$ or $(1,-1,\dots,(-1)^{t-1};1,\dots,1;-1,\dots,-1)$ $(t=\min\{n,p\})$, respectively. When considering the subclasses of Cauchy matrices by the positive or negative property of nodes,

$$\begin{cases}\mathcal{C}_{n,p}^{+}=\{A|\ A=\left[\frac{1}{x_i+y_j}\right]_{i,j=1}^{n,p},\ \text{where } x_i+y_j>0,\ \forall\ i,j\},\\ \mathcal{C}_{n,p}^{-}=\{A|\ A=\left[\frac{1}{x_i+y_j}\right]_{i,j=1}^{n,p},\ \text{where } x_i+y_j<0,\ \forall\ i,j\},\end{cases}$$

for any $A\in\mathcal{C}_{n,p}^{+}$ or $A\in\mathcal{C}_{n,p}^{-}$, there exist permutation matrices P_1 and P_2 such that the nodes of $B=P_1AP_2$ are ascending-ordered whose generators (1.4) satisfy the sign regularity (1.5) with the sign sequence $(1,\dots,1;1,\dots,1;1,\dots,1)$ or $(-1,\dots,-1;1,\dots,1;1,\dots,1)$, respectively. In numerical experiments later, we list more structured matrices having the sign regularity (1.5). Moreover, we will find out that the sign regularity plays a crucial role in high-accuracy computations. Fortunately, the generators for (1.3) and (1.4) are computed to high relative accuracy because products and quotients of the factors $x-y$ and $x+y$ involving initial data x and y are computable to high relative accuracy [11]. At present, many classes of structured matrices whose BF forms can be computed with high relative accuracy have been identified in the literature [7, 9, 13, 27–32]. Therefore, our main aim of this paper is to design a high-accuracy algorithm for the GSV problem of BF matrix pairs.

It is a desirable goal to provide algorithms to accurately compute the GSVs as warranted by the data. In [16], Drmač considered the GSV problem for a matrix pair of full column rank, and provided a tangent algorithm and its generalization [17] to accurately compute the GSVs as predicted by the theoretical perturbations via matrix entries. In this paper, we consider the GSV problem for a BF matrix pair $(A,B)\in\mathbb{R}^{n\times p}\times\mathbb{R}^{m\times p}$ of full rank with one of them being full column rank. Our main contributions are the following:

- We show that all the GSVs $\sigma_i(A,B)$ are determined to high relative accuracy by the generators α_{ij} and β_{ij} of $A=:\mathcal{BF}(\alpha_{ij})$ and $B=:\mathcal{BF}(\beta_{ij})$ satisfying the sign regularity (1.5) with (3.13) later, i.e.,

$$|\sigma_i(\tilde{A},\tilde{B})-\sigma_i(A,B)|\le\frac{8(n+m)p\epsilon}{1-8(n+m)p\epsilon}\sigma_i(A,B),\quad \forall\ 1\le i\le p, \tag{1.6}$$

  where $\tilde{A}=:\mathcal{BF}(\tilde{\alpha}_{ij})\in\mathbb{R}^{n\times p}$ and $\tilde{B}=:\mathcal{BF}(\tilde{\beta}_{ij})\in\mathbb{R}^{m\times p}$ with $\tilde{\alpha}_{ij}=\alpha_{ij}(1+\epsilon_{ij})$ and $\tilde{\beta}_{ij}=\beta_{ij}(1+\epsilon'_{ij})$ for all i and j, $|\epsilon_{ij}|,|\epsilon'_{ij}|\le\epsilon$ and $8(n+m)p\epsilon<1$.
- We provide a qd-type method to compute the GSVs $\sigma_i(A,B)$. The method is to first implicitly compute the factor R of the thin QR factorization of B, and then implicitly reduce AR^{-1} into a bidiagonal matrix by orthogonal transformations, and finally, compute singular values of the bidiagonal matrix as the GSVs. All the GSVs, no matter how large or small they are, are

computed to high relative accuracy as predicted by the strong perturbation result (1.6), i.e.,

$$|\hat{\sigma}_i(A,B)-\sigma_i(A,B)| \le \frac{O(16(n+m+p)p^2)\mu}{1-O(16(n+m+p)p^2)\mu}\sigma_i(A,B), \quad \forall\ 1\le i\le p,$$

independent of any traditional condition number, where μ is the unit roundoff. Throughout the paper, all the computed quantities wear a hat.

The rest of the paper is organized as follows. In Section 2, some core transformations are provided in order to design our qd-type method. In Section 3, the qd-type method is proposed to compute the GSVs of a BF matrix pair of full rank in a preferable complexity. A mechanism involving sign regularity of BF generators is derived to guarantee that there is no subtraction of like-signed numbers for the qd-type method. In Section 4, we first develop a strong perturbation analysis to show that all the GSVs are determined to high relative accuracy by the BF generators, and we then present an error analysis to illustrate that the qd-type method computes all the GSVs to high relative accuracy as warranted by the BF generators. In Section 5, numerical experiments are performed to confirm the claimed high relative accuracy.

2. Core transformations

In this section, we provide core transformations for designing our qd-type method.

Lemma 2.1. *Let*

$$L_r=\mathbf{bilow}(1,\dots,1;l_{r+1},\dots,l_n)\in\mathbb{R}^{n\times n},\ \forall\ 1\le r\le n-1.$$

Then L_r^{-1} has the QR factorization as follows:

$$L_r^{-1}=Q_r^TX_r^{-1}, \tag{2.1}$$

where $Q_r=G_r\cdots G_{n-1}$ consists of the Givens rotations $G_i\in\mathbb{R}^{n\times n}$ with

$$G_i[i,i+1]=\begin{bmatrix} c_i & -s_i \\ s_i & c_i\end{bmatrix}, \quad r\le i\le n-1,$$

and

$$X_r=\mathbf{biupp}(1,\dots,1,d_r,\dots,d_n;x_{r+1},\dots,x_n)\in\mathbb{R}^{n\times n};$$

here,

$$\begin{cases} c_n=1, \\ d_{i+1}=\sqrt{c_{i+1}^2+l_{i+1}^2}, \\ c_i=\frac{c_{i+1}}{d_{i+1}},\ s_i=x_{i+1}=\frac{l_{i+1}}{d_{i+1}}, \\ d_r=c_r, \end{cases} \qquad i=n-1,\dots,r.$$

In particular,

$$d_i>0,\ \operatorname{sign}(x_i)=\operatorname{sign}(l_i)\ (i\ne r), \quad \forall\ r\le i\le n.$$

Proof. By a straight calculation, we have $L_rQ_r^T=X_r$, thus, the result is true. □

Lemma 2.2. *Let*

$$\begin{cases} L_r=\mathbf{bilow}(1,\dots,1;l_{r+1},\dots,l_n)\in\mathbb{R}^{n\times n},\ \forall\ 1\le r\le n-1, \\ X=\mathbf{biupp}(d_1,\dots,d_n;x_2,\dots,x_n)\in\mathbb{R}^{n\times n},\ \textit{where } d_i>0,\ \forall\ 1\le i\le n. \end{cases}$$

Then in the absence of breakdown,

$$X^{-1}L_r^{-1} = \bar{L}_r^{-1}\bar{X}^{-1}, \tag{2.2}$$

where

$$\begin{cases} \bar{L}_r = \mathbf{bilow}(1,\ldots,1;\bar{l}_{r+1},\ldots,\bar{l}_n) \in \mathbb{R}^{n\times n}, \\ \bar{X} = \mathbf{biupp}(d_1,\ldots,d_{r-1},\bar{d}_r,\ldots,\bar{d}_n;x_2,\ldots,x_n) \in \mathbb{R}^{n\times n}; \end{cases}$$

here,

$$\begin{cases} z_n = d_n, \\ \bar{d}_i = z_i + l_i x_i, \\ \begin{cases} \bar{l}_i = 0,\ z_{i-1} = d_{i-1}, \ \text{if } l_i = 0, \\ \bar{l}_i = \frac{d_{i-1}l_i}{\bar{d}_i},\ z_{i-1} = \frac{d_{i-1}z_i}{\bar{d}_i},\ \text{otherwise}, \end{cases} & i = n,\ldots,r+1. \\ \bar{d}_r = z_r, \end{cases}$$

Moreover, the computation of (2.2) *is subtraction-free if and only if* $\operatorname{sign}(l_i x_i) = 1$ *or* 0 *for all* $r+1 \le i \le n$. *In this case, no breakdown occurs for* (2.2), *and*

$$\bar{d}_i > 0,\ \operatorname{sign}(\bar{l}_i) = \operatorname{sign}(l_i)\ (i \ne r), \quad \forall\ r \le i \le n.$$

Proof. By comparing the entries in both sides of $L_r X = \bar{X}\bar{L}_r$, we have

$$\begin{cases} \bar{d}_n = z_n + l_n x_n \text{ with } z_n = d_n, \\ l_i d_{i-1} = \bar{l}_i \bar{d}_i, & i = n,\ldots,r+1, \\ d_{i-1} + l_{i-1}x_{i-1} = \bar{d}_{i-1} + \bar{l}_i x_i, \end{cases}$$

with the convention $l_r = 0$, from which it follows that

- if $l_i = 0$, then $\bar{l}_i = 0$, thus, $\bar{d}_{i-1} = z_{i-1} + l_{i-1}x_{i-1}$ with $z_{i-1} = d_{i-1}$;
- if $l_i \ne 0$, then $l_i d_{i-1} \ne 0$, and so, in the absence of breakdown, $\bar{l}_i = \frac{l_i d_{i-1}}{\bar{d}_i}$, thus, $\bar{d}_{i-1} = z_{i-1} + l_{i-1}x_{i-1}$ with $z_{i-1} = d_{i-1} - \bar{l}_i x_i = \frac{d_{i-1}z_i}{\bar{d}_i}$.

Notice that $L_r X$ is nonsingular. So, we get the desired result $X^{-1}L_r^{-1} = \bar{L}_r^{-1}\bar{X}^{-1}$. Moreover, if $\operatorname{sign}(l_i x_i) = 1$ or 0 for all $r+1 \le i \le n$, since $d_i > 0$ for all i, we inductively get by the formulas of (2.2) that for $i = n,\ldots,r$, $\operatorname{sign}(z_i) = 1$ such that the computation of $\bar{d}_i$ is subtraction-free with $\operatorname{sign}(\bar{d}_i) = 1$. Conversely, if $\bar{d}_i = z_i + l_i x_i$ is subtraction-free for all $r+1 \le i \le n$, since $d_i > 0$ for all i, we inductively have by the formulas of (2.2) that for $i = n,\ldots,r+1$, $\operatorname{sign}(z_i) = 1$ such that $\operatorname{sign}(l_i x_i) = 1$ or 0 with $\operatorname{sign}(\bar{d}_i) = 1$ and $\operatorname{sign}(\bar{l}_i) = \operatorname{sign}(l_i)$. In this case, no breakdown occurs because $\bar{d}_i > 0$ for all $r \le i \le n$. The result is proved. □

Lemma 2.3. *Let*

$$\begin{cases} X_s = \mathbf{biupp}(1,\ldots,1;x_{s+1},\ldots,x_n) \in \mathbb{R}^{n\times n}, \\ U_r = \mathbf{biupp}(1,\ldots,1;u_{r+1},\ldots,u_n) \in \mathbb{R}^{n\times n}, \end{cases} \quad 1 \le s \le r \le n-1.$$

Then in the absence of breakdown,

$$X_s^{-1}U_r^{-1} = \bar{U}_{r+1}^{-1}\bar{X}_s^{-1}, \tag{2.3}$$

where

$$\begin{cases} \bar{X}_s = \mathbf{biupp}(1,\ldots,1;x_{s+1},\ldots,x_r,\bar{x}_{r+1},\ldots,\bar{x}_n) \in \mathbb{R}^{n\times n}, \\ \bar{U}_{r+1} = \mathbf{biupp}(1,\ldots,1;\bar{u}_{r+2},\ldots,\bar{u}_n) \in \mathbb{R}^{n\times n}, \end{cases}$$

with the convention $\bar{U}_n = I_n$, *and*

$$\begin{cases} z_{r+1} = x_{r+1}, \\ \bar{x}_i = z_i + u_i, \\ \begin{cases} \bar{u}_{i+1} = 0, \ z_{i+1} = x_{i+1}, \ \textit{if } x_{i+1}u_i = 0, \\ \bar{u}_{i+1} = \frac{x_{i+1}u_i}{\bar{x}_i}, \ z_{i+1} = \frac{x_{i+1}z_i}{\bar{x}_i}, \ \textit{otherwise}, \end{cases} \quad (i \neq n); \end{cases} \qquad i = r+1, \ldots, n.$$

Moreover, the computation of (2.3) *is subtraction-free if and only if* $\operatorname{sign}(x_i u_i) = 1$ *or* 0 *for all* $r+1 \le i \le n$. *In this case, no breakdown occurs for* (2.3), *and*

$$\begin{cases} \operatorname{sign}(\bar{x}_i) = \operatorname{sign}(x_i) \ \textit{or} \ \operatorname{sign}(u_i), & \textit{whenever } \bar{x}_i \neq 0, \\ \operatorname{sign}(\bar{u}_{i+1}) = \operatorname{sign}(x_{i+1}), & \textit{whenever } \bar{u}_{i+1} \neq 0 \ (i \neq n), \end{cases} \qquad \forall \ r+1 \le i \le n. \tag{2.4}$$

Proof. By comparing the entries in both sides of $U_r X_s = \bar{X}_s \bar{U}_{r+1}$ with the convention $\bar{U}_n = I_n$, we have that

$$\begin{cases} \bar{x}_{r+1} = z_{r+1} + u_{r+1} \text{ with } z_{r+1} = x_{r+1}, \\ u_i x_{i+1} = \bar{x}_i \bar{u}_{i+1} \ (i \neq n), \\ x_{i+1} + u_{i+1} = \bar{x}_{i+1} + \bar{u}_{i+1} \ (i \neq n), \end{cases} \qquad i = r+1, \ldots, n,$$

from which it follows that

- if $u_i x_{i+1} = 0$, then $\bar{u}_{i+1} = 0$, thus, $\bar{x}_{i+1} = z_{i+1} + u_{i+1}$ with $z_{i+1} = x_{i+1}$;
- if $u_i x_{i+1} \neq 0$, then in the absence of breakdown, $\bar{u}_{i+1} = \frac{x_{i+1}u_i}{\bar{x}_i}$, thus, $\bar{x}_{i+1} = z_{i+1} + u_{i+1}$ with $z_{i+1} = x_{i+1} - \bar{u}_{i+1} = \frac{x_{i+1}z_i}{\bar{x}_i}$.

Notice that $U_r X_s$ is nonsingular. So, we get the desired result $\bar{U}_{r+1}^{-1}\bar{X}_s^{-1} = X_s^{-1}U_r^{-1}$. Moreover, if $\operatorname{sign}(x_i u_i) = 1$ or 0 for all $r+1 \le i \le n$, then we inductively get by the formulas of (2.3) that for $i = r+1, \ldots, n$, $\operatorname{sign}(z_i x_i) = 1$ or 0 such that $\bar{x}_i = z_i + u_i$ is subtraction-free with $\operatorname{sign}(\bar{x}_i z_i) = 1$ or 0. Conversely, if $\bar{x}_i = z_i + u_i$ is subtraction-free for all $r+1 \le i \le n$, then we inductively have by the formulas of (2.3) that for $i = r+1, \ldots, n$, $\operatorname{sign}(z_i x_i) = 1$ or 0 such that $\operatorname{sign}(x_i u_i) = 1$ or 0 with $\operatorname{sign}(\bar{x}_i z_i) = 1$ or 0. In this case, if $\bar{x}_i = 0$ for some $r+1 \le i \le n$, since $\bar{x}_i = z_i + u_i$ is subtraction-free, we have that $u_i = 0$ and so $x_{i+1}u_i = 0$, thus, no breakdown occurs for the formula of (2.3). In particular, if $\bar{x}_i \neq 0$, we have $u_i \neq 0$ or $z_i \neq 0$ (and so $x_i \neq 0$), thus $\operatorname{sign}(\bar{x}_i) = \operatorname{sign}(u_i)$ or $\operatorname{sign}(x_i)$; and if $\bar{u}_{i+1} \neq 0$, we have $x_{i+1} \neq 0$, thus $\operatorname{sign}(\bar{u}_{i+1}) = \operatorname{sign}(x_{i+1})$. So, the fact (2.4) is true. □

Remark 2.4. The technical results of Lemmas 2.1, 2.2, and 2.3 are aimed at performing qd-type transformations on BF matrices by working with BF forms later.

Now, let

$$X_s^{(1)} = \textbf{biupp}(d_1^{(1)}, \ldots, d_n^{(1)}; -x_{s+1}, \ldots, -x_n) \in \mathbb{R}^{n \times n}, \ \forall \ 1 \le s \le n-1 \tag{2.5}$$

be nonsingular, and let $A =: \mathcal{BF}(\alpha_{ij}) \in \mathbb{R}^{n \times p}$ be as in (1.1). Set $\bar{A} = (X_s^{(1)})^{-1}A$. Then in the absence of breakdown, the BF form of $\bar{A} =: \mathcal{BF}(\bar{\alpha}_{ij}) \in \mathbb{R}^{n \times p}$ is computed by the qd-type transformation as follows:

$$\begin{aligned} \bar{A} &= (X_s^{(1)})^{-1} L_1^{-1} \cdots L_t^{-1} D U_t^{-1} \cdots U_1^{-1} \\ &= \bar{L}_1^{-1} \cdots \bar{L}_t^{-1} (X_s^{(t+1)})^{-1} D U_t^{-1} \cdots U_1^{-1} \\ &= \bar{L}_1^{-1} \cdots \bar{L}_t^{-1} \bar{D} (Y_s^{(t+1)})^{-1} U_t^{-1} \cdots U_1^{-1} \\ &= \bar{L}_1^{-1} \cdots \bar{L}_t^{-1} \bar{D} \bar{U}_t^{-1} \cdots \bar{U}_s^{-1} U_{s-1}^{-1} \cdots U_1^{-1}, \end{aligned} \tag{2.6}$$

where

- first, by Lemma 2.2 recursively,

$$(X_s^{(r)})^{-1}L_r^{-1} = \bar{L}_r^{-1}(X_s^{(r+1)})^{-1}, \quad r = 1, \dots, t, \tag{2.7}$$

  where

$$\begin{cases} \bar{L}_r = \mathbf{bilow}(1, \dots, 1; -\bar{\alpha}_{r+1,r}, \dots, -\bar{\alpha}_{nr}) \in \mathbb{R}^{n\times n}, \\ X_s^{(r+1)} = \mathbf{biupp}(d_1^{(r+1)}, \dots, d_n^{(r+1)}; -x_{s+1}, \dots, -x_n) \in \mathbb{R}^{n\times n}; \end{cases}$$

- further, by a straight calculation,

$$(X_s^{(t+1)})^{-1}D = \bar{D}(Y_s^{(t+1)})^{-1}, \tag{2.8}$$

  where

$$\begin{cases} \bar{D} = \mathbf{diag}(\bar{\alpha}_{ii}) \in \mathbb{R}^{n\times p}, \\ Y_s^{(t+1)} = \mathbf{biupp}(1, \dots, 1; -y_{s+1}^{(t+1)}, \dots, -y_p^{(t+1)}) \in \mathbb{R}^{p\times p}, \end{cases}$$

  with $Y_s^{(t+1)} = I_p$ if $s \geq p$, and

$$\begin{cases} \bar{\alpha}_{ii} = \alpha_{ii}/d_i^{(t+1)}, \ \forall \ 1 \leq i \leq t, \\ \begin{cases} y_i^{(t+1)} = \frac{x_i \cdot \bar{\alpha}_{ii}}{\alpha_{i-1,i-1}}, \ \text{if } s+1 \leq i \leq t, \\ y_i^{(t+1)} = 0, \ \text{otherwise}; \end{cases} \end{cases} \tag{2.9}$$

- finally, if $s < p$ (and so $s < t$), then by Lemma 2.3 recursively,

$$(Y_s^{(r+1)})^{-1}U_r^{-1} = \bar{U}_{r+1}^{-1}(Y_s^{(r)})^{-1}, \ r = t, t-1, \dots, s, \tag{2.10}$$

  where

$$\begin{cases} \bar{U}_{r+1} = \mathbf{biupp}(1, \dots, 1; -\bar{\alpha}_{r+1,r+2}, \dots, -\bar{\alpha}_{r+1,p}) \in \mathbb{R}^{p\times p}, \\ Y_s^{(r)} = \mathbf{biupp}(1, \dots, 1; -y_{s+1}^{(r)}, \dots, -y_p^{(r)}) \in \mathbb{R}^{p\times p}, \end{cases}$$

  with the convention $\bar{U}_{t+1} = I_p$ and $\bar{U}_s = Y_s^{(s)}$.

Observe that each of (2.7), (2.8), and (2.10) costs at most $5(n-r)$, $2 \cdot \max\{t - s, 0\} + t$, and $4(p-r)$ arithmetic operations, respectively. So, the cost of the qd-type transformation (2.6) is at most

$$\sum_{r=1}^{t} 5(n-r) + 2 \cdot \max\{t-s, 0\} + t + \sum_{r=s}^{t} 4(p-r) = O(np),$$

with the convention $\sum_{r=s}^{t} \cdot = 0$ if $s > t$.

Interestingly, the qd-type transformation (2.6) is performed without any subtraction of like-signed numbers for BF matrices satisfying the sign regularity (1.5).

Theorem 2.5. *Let* $A =: \mathcal{BF}(\alpha_{ij}) \in \mathbb{R}^{n\times p}$ *be of full rank satisfying the sign regularity* (1.5) *with the sign sequence* $(s_1, \dots, s_t; r_2, \dots, r_n; c_2, \dots, c_p)$ $(t = \min\{n, p\})$, *and let* $X_s^{(1)} \in \mathbb{R}^{n\times n}$ $(1 \leq s \leq n-1)$ *be as in* (2.5) *with*

$$\begin{cases} d_i^{(1)} > 0, \ \forall \ 1 \leq i \leq n, \\ \text{sign}(x_i) = r_i \ \textit{or} \ c_i s_{i-1} s_i \ \textit{if} \ r_i x_i \neq 0 \ \textit{or} \ c_i x_i \neq 0, \forall \ s+1 \leq i \leq n. \end{cases} \tag{2.11}$$

Set $\bar{A} = (X_s^{(1)})^{-1}A$. Then the BF form of $\bar{A} =: \mathcal{BF}(\bar{\alpha}_{ij}) \in \mathbb{R}^{n\times p}$ is computed by the qd-type transformation (2.6) *in a subtraction-free way with*

(2.12)

$$\begin{cases} \operatorname{sign}(\bar{\alpha}_{ii}) = s_i,\ \forall\ 1 \le i \le t, \\ \operatorname{sign}(\bar{\alpha}_{i1}, \dots, \bar{\alpha}_{i,\min\{i-1,p\}}) = r_i \ \textit{if}\ (\bar{\alpha}_{i1}, \dots, \bar{\alpha}_{i,\min\{i-1,p\}}) \ne 0,\ \forall\ 2 \le i \le n, \\ \operatorname{sign}(\bar{\alpha}_{1i}, \dots, \bar{\alpha}_{\min\{i-1,n\},i}) = c_i \ \textit{or}\ \operatorname{sign}(x_i)\cdot s_{i-1}s_i \\ \qquad\qquad \textit{if}\ (\bar{\alpha}_{1i}, \dots, \bar{\alpha}_{\min\{i-1,n\},i}) \ne 0,\ \forall\ 2 \le i \le p. \end{cases}$$

Proof. Let A be as in (1.1). Consider $\bar{A} = (X_s^{(1)})^{-1}A$. By the fact (2.11), we have

$$\operatorname{sign}(\alpha_{i1}x_i) = 1 \text{ or } 0, \quad \forall\ s+1 \le i \le n,$$

thus, by Lemma 2.2, for the case $r = 1$ of (2.7),

$$(X_s^{(1)})^{-1}L_1^{-1} = \bar{L}_1^{-1}(X_s^{(2)})^{-1}$$

is subtraction-free with

$$\begin{cases} \operatorname{sign}(\bar{\alpha}_{i1}) = \operatorname{sign}(\alpha_{i1}),\ \forall\ 2 \le i \le n, \\ d_i^{(2)} > 0,\ \forall\ 1 \le i \le n. \end{cases}$$

Hence, taking into account (2.11), we inductively get by Lemma 2.2 that the computation of (2.7) is subtraction-free for all $r = 1, \dots, t = \min\{n, p\}$ with

$$\begin{cases} \operatorname{sign}(\bar{\alpha}_{ir}) = \operatorname{sign}(\alpha_{ir}), \quad \forall\ r+1 \le i \le n, \\ d_i^{(r+1)} > 0,\ \forall\ 1 \le i \le n. \end{cases} \tag{2.13}$$

Further, for the transformation (2.8), since A is of full rank, we have the fact (2.9) in a subtraction-free way with no breakdown, which implies with (2.13) that

$$\begin{cases} \operatorname{sign}(\bar{\alpha}_{ii}) = s_i,\ \forall\ 1 \le i \le t, \\ \operatorname{sign}(y_i^{(t+1)}) = \operatorname{sign}(x_i)\cdot s_{i-1}s_i,\ \forall\ s+1 \le i \le t. \end{cases} \tag{2.14}$$

By (2.11), $\operatorname{sign}(x_i)\cdot s_{i-1}s_i = c_i$ if $c_i x_i \ne 0$. Thus, we inductively get by Lemma 2.3 that the computation of (2.10) is subtraction-free for all $r = t, t-1, \dots, s$ with

$$\begin{cases} y_i^{(r)} = y_i^{(r+1)},\ \forall\ s+1 \le i \le r, \\ \operatorname{sign}(y_i^{(r)}) = \operatorname{sign}(y_i^{(r+1)}) \text{ or } \operatorname{sign}(\alpha_{ri}), \quad \text{if } y_i^{(r)} \ne 0,\ \forall\ r+1 \le i \le p, \\ \operatorname{sign}(\bar{\alpha}_{r+1,i}) = \operatorname{sign}(y_i^{(r+1)}), \quad \text{if } \bar{\alpha}_{r+1,i} \ne 0,\ \forall\ r+2 \le i \le p, \end{cases}$$

which means with (2.14) that if $(\bar{\alpha}_{1i}, \dots, \bar{\alpha}_{\min\{i-1,n\},i}) \ne 0$, then

$$\operatorname{sign}(\bar{\alpha}_{1i}, \dots, \bar{\alpha}_{\min\{i-1,n\},i}) = c_i \text{ or } \operatorname{sign}(x_i)\cdot s_{i-1}s_i, \quad \forall\ 2 \le i \le p.$$

The result is proved. □

3. A QD-TYPE METHOD

In this section, we provide a qd-type method to compute the GSVs of a matrix pair $(A, B) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{m\times p}$ of full rank with one of them being full column rank, whose BF forms are as in (1.1) as follows:

$$\begin{cases} A = F_1^{-1}\cdots F_t^{-1}HC_t^{-1}\cdots C_1^{-1} =: \mathcal{BF}(\alpha_{ij}) \in \mathbb{R}^{n\times p},\ t = \min\{n, p\}, \\ B = L_1^{-1}\cdots L_{t'}^{-1}DU_{t'}^{-1}\cdots U_1^{-1} =: \mathcal{BF}(\beta_{ij}) \in \mathbb{R}^{m\times p},\ t' = \min\{m, p\}, \end{cases} \tag{3.1}$$

where $H \in \mathbb{R}^{n\times p}$ and $D \in \mathbb{R}^{m\times p}$ are diagonal, all $F_i \in \mathbb{R}^{n\times n}$, $L_i \in \mathbb{R}^{m\times m}$, and $C_i, U_i \in \mathbb{R}^{p\times p}$ are lower and upper bidiagonal as in (1.2), respectively. Without loss of generality, assume that A is of full rank and B is of full column rank. Otherwise, we interchange the role of A and B to compute the GSVs of (B, A) by considering that $\sigma_i(A, B) = 1/\sigma_i(B, A)$ for all $1 \le i \le p$.

3.1. **Computing the thin QR factorization.** First, we compute the R factor of the thin QR factorization of the matrix B under the assumption that no breakdown occurs. Set $B_0 = B$. Taking into account the form (3.1) of B_0, by Lemma 2.1, we have an orthogonal matrix $Q_1 \in \mathbb{R}^{m\times m}$ as in (2.1) such that

$$L_1^{-1} = Q_1^T X_1^{-1}, \tag{3.2}$$

with $X_1 = \mathbf{biupp}(d_1^{(1)}, \dots, d_m^{(1)}; -x_2, \dots, -x_m) \in \mathbb{R}^{m\times m}$, thus,

$$\begin{aligned} B_1 = Q_1 B_0 &= X_1^{-1} L_2^{-1} \cdots L_p^{-1} D U_p^{-1} \cdots U_1^{-1} \\ &= \bar{L}_2^{-1} \cdots \bar{L}_p^{-1} \bar{D} \bar{U}_p^{-1} \cdots \bar{U}_1^{-1} \end{aligned} \tag{3.3}$$

is computed by the qd-type transformation (2.6) in $O(mp)$ arithmetic operations. Consequently, the off-diagonal entries in the first column of B_0 are eliminated to be zero with the BF form being preserved. After proceeding with the analogous operations on the columns 2 to p in the procedure

$$B = B_0 \to B_1 \to \cdots \to B_p, \tag{3.4}$$

we have orthogonal transformations $Q_i \in \mathbb{R}^{m\times m}$ $(1 \le i \le p)$ as in (2.1) such that B is reduced into an upper triangular matrix B_p, whose BF form is the following:

$$B_p = Q_p \cdots Q_1 B = \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{\bar{D}} \bar{\bar{U}}_p^{-1} \cdots \bar{\bar{U}}_1^{-1} \\ 0 \end{bmatrix} \in \mathbb{R}^{m\times p}, \tag{3.5}$$

where $\bar{\bar{D}} = \operatorname{diag}(u_{ii}) \in \mathbb{R}^{p\times p}$ and $\bar{\bar{U}}_i = \mathbf{biupp}(1, \dots, 1; -u_{i,i+1}, \dots, -u_{ip}) \in \mathbb{R}^{p\times p}$ $(1 \le i \le p)$. Notice that R of (3.5) is the R factor of the thin QR factorization of B. Therefore, we conclude Algorithm 1 to implicitly compute the factor R in $O(mp^2)$ arithmetic operations.

Algorithm 1 The algorithm implicitly computes the factor R of (3.5) for the thin QR factorization of a BF matrix $B =: \mathcal{BF}(\beta_{ij}) \in \mathbb{R}^{m\times p}$ of full column rank.

1: Set $B_0 = B$ whose BF form is as in (3.1).
2: **for** $i = 1 : p$ **do**
3: Compute the QR factorization $L_i^{-1} = Q_i^T X_i^{-1}$ by (2.1).
4: Compute the BF form of $B_i = Q_i B_{i-1}$ by (2.6), which is denoted as $B_i =: L_{i+1}^{-1} \cdots L_p^{-1} D U_p^{-1} \cdots U_1^{-1}$ for notational simplicity.
5: **end for**

Interestingly, the following result shows that the factor R of (3.5) can be computed without any subtraction of like-signed numbers.

Theorem 3.1. *Let $B \in \mathbb{R}^{m\times p}$ be as in (3.1) of full column rank satisfying the sign regularity (1.5) with the sign sequence $(s_1, \dots, s_p; r_2, \dots, r_m; c_2, \dots, c_p)$, and let $R \in \mathbb{R}^{p\times p}$ be as in (3.5) for the thin QR factorization of B. Then $R =: \mathcal{BF}(u_{ij})$ is computed by Algorithm 1 in a subtraction-free way satisfying that for all $1 \le i \le p$,*

$$\operatorname{sign}(u_{ii}) = s_i; \quad \operatorname{sign}(u_{1i}, \dots, u_{i-1,i}) = c_i \text{ or } r_i s_i s_{i-1} \text{ if } (u_{1i}, \dots, u_{i-1,i}) \ne 0. \tag{3.6}$$

Proof. First, by Lemma 2.1, X_1 of (3.2) is computed in a subtraction-free way satisfying (2.11), and thus, by Theorem 2.5, B_1 of (3.3) is computed by (2.6) in a subtraction-free way whose generators satisfy (2.12). Therefore, when proceeding with the procedure (3.4), we inductively get that the intermediate matrices B_i $(2 \le i \le p)$ are computed by (2.1) and (2.6) in a subtraction-free way whose generators satisfy (2.12). So, the result is true. $\square$

3.2. **The bidiagonal reduction.** Now, using the BF form of the factor R of (3.5), for the notational simplicity, the BF form of $W = R^{-1}$ is denoted as

$$W = DU_p^{-1} \cdots U_1^{-1} \in \mathbb{R}^{p \times p}, \tag{3.7}$$

where $D = \operatorname{diag}(w_{ii}) \in \mathbb{R}^{p\times p}$ and $U_i = \mathbf{biupp}(1, \ldots, 1; -w_{i,i+1}, \ldots, -w_{ip}) \in \mathbb{R}^{p\times p}$ $(1 \le i \le p)$ with

$$\begin{cases} w_{ii} = 1/u_{ii}, \ \forall\ 1 \le i \le p, \\ w_{ij} = -(u_{j-i,j} \cdot u_{j-1,j-1})/u_{jj}, \ \forall\ i+1 \le j \le p. \end{cases} \tag{3.8}$$

Taking into account the BF forms (3.1) and (3.7) of $A \in \mathbb{R}^{n\times p}$ and $W \in \mathbb{R}^{p\times p}$, respectively, we next show how to reduce AW into a bidiagonal matrix by orthogonal transformations under the assumption that no breakdown occurs. We illustrate the reduction for the case $n = 3$ and $p = 5$, i.e.,

$$\begin{aligned} AW &= (F_1^{-1}F_2^{-1}HC_3^{-1}C_2^{-1}C_1^{-1})(DU_4^{-1}U_3^{-1}U_2^{-1}U_1^{-1}) \\ &= \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \begin{bmatrix} y & y & y & y & y \\ 0 & y & y & y & y \\ 0 & 0 & y & y & y \\ 0 & 0 & 0 & y & y \\ 0 & 0 & 0 & 0 & y \end{bmatrix}, \end{aligned}$$

where x and y indicate the nontrivial entries of A and W, respectively.

- First, using Lemma 2.1, we get the QR factorization $F_1^{-1} = Q_1^T X_1^{-1}$ as in (2.1), thus, by the qd-type transformation (2.6),

  $$Q_1 A = X_1^{-1}F_2^{-1}HC_3^{-1}C_2^{-1}C_1^{-1} = \bar{F}_2^{-1}\bar{H}\bar{C}_3^{-1}\bar{C}_2^{-1}\bar{C}_1^{-1},$$

  and further, let

  $$A_1 = \bar{F}_2^{-1}\bar{H}\bar{C}_3^{-1}\bar{C}_2^{-1}, \quad W_1 = \bar{C}_1^{-1}W.$$

  Then by the qd-type transformation (2.6) again,

  $$W_1 = \bar{C}_1^{-1}(DU_4^{-1}U_3^{-1}U_2^{-1}U_1^{-1}) = \bar{D}\bar{U}_4^{-1}\bar{U}_3^{-1}\bar{U}_2^{-1}\bar{U}_1^{-1}.$$

  So,

  $$\begin{aligned} Q_1AW = A_1W_1 &=: (\bar{F}_2^{-1}\bar{H}\bar{C}_3^{-1}\bar{C}_2^{-1})(\bar{D}\bar{U}_4^{-1}\bar{U}_3^{-1}\bar{U}_2^{-1}\bar{U}_1^{-1}) \\ &= \begin{bmatrix} x & 0 & 0 & 0 & 0 \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} y & y & y & y & y \\ 0 & y & y & y & y \\ 0 & 0 & y & y & y \\ 0 & 0 & 0 & y & y \\ 0 & 0 & 0 & 0 & y \end{bmatrix}. \end{aligned}$$

  Consequently, the off-diagonal entries in the first row/column of A are eliminated to be zero with the BF forms being preserved.

- Second, rewrite $\bar{U}_1 = \bar{U}_2' E_2^T(-c_{12})$, where $\bar{U}_2'$ is obtained from $\bar{U}_1$ by setting its $(1,2)$th entry $-c_{12} = 0$. By the transposing version of Lemma 2.1, we have an orthogonal matrix P_1 such that $(\bar{U}_2')^{-1} = Y_2^{-1} P_1^T$ with $Y_2 = \mathrm{diag}(1, Y_2')$ being lower bidiagonal. So,
$$\bar{U}_1^{-1} P_1 = E_2^T(c_{12})(\bar{U}_2')^{-1} P_1 = Y_2^{-1} E_2^T(c_{12}),$$
thus, by the transposing version of (2.6),
$$W_1 P_1 = \bar{D}\bar{U}_4^{-1}\bar{U}_3^{-1}\bar{U}_2^{-1} Y_2^{-1} E_2^T(c_{12}) = \bar{Y}_2^{-1}\mathcal{D}\mathcal{U}_4^{-1}\mathcal{U}_3^{-1}\mathcal{U}_2^{-1} E_2^T(c_{12}),$$
where $\bar{Y}_2$ has the same pattern as that of Y_2; and further, let
$$W_2 = \mathcal{D}\mathcal{U}_4^{-1}\mathcal{U}_3^{-1}\mathcal{U}_2^{-1} E_2^T(c_{12}), \quad A_2 = A_1 \bar{Y}_2^{-1}.$$
Then by the transposing version of (2.6) again,
$$A_2 = (\bar{F}_2^{-1}\bar{H}\bar{C}_3^{-1}\bar{C}_2^{-1})\bar{Y}_2^{-1} = \mathcal{F}_2^{-1}\mathcal{H}\mathcal{C}_3^{-1}\mathcal{C}_2^{-1}.$$
So,
$$\begin{aligned} A_1 W_1 P_1 &= A_2 W_2 =: (\mathcal{F}_2^{-1}\mathcal{H}\mathcal{C}_3^{-1}\mathcal{C}_2^{-1})(\mathcal{D}\mathcal{U}_4^{-1}\mathcal{U}_3^{-1}\mathcal{U}_2^{-1} E_2^T(c_{12})) \\ &= \begin{bmatrix} x & 0 & 0 & 0 & 0 \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} y & y & 0 & 0 & 0 \\ 0 & y & y & y & y \\ 0 & 0 & y & y & y \\ 0 & 0 & 0 & y & y \\ 0 & 0 & 0 & 0 & y \end{bmatrix}. \end{aligned}$$
Consequently, the off-diagonal entries in the first row of W_1, except for the $(1,2)$th entry, are eliminated to be zero, and the zero entries of A_1 are not disturbed, with the BF forms being preserved.

Therefore, when proceeding with the analogous operations above on the remaining rows/columns 2 to $t = \min\{n, p\} = 3$, we conclude that there are orthogonal matrices Q_i $(1 \le i \le 2)$ and P_i $(1 \le i \le 3)$ such that
$$(Q_2 Q_1)(AW)(P_1 P_2 P_3) = \begin{bmatrix} x & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 \end{bmatrix} \begin{bmatrix} y & y & 0 & 0 & 0 \\ 0 & y & y & 0 & 0 \\ 0 & 0 & y & y & 0 \\ 0 & 0 & 0 & y & y \\ 0 & 0 & 0 & 0 & y \end{bmatrix}.$$

For the general case of $A \in \mathbb{R}^{n\times p}$ and $W \in \mathbb{R}^{p\times p}$, by performing the analogous reductions above on rows/columns 1 to $t = \min\{n, p\}$ in the procedure
$$(A_0, W_0) = (A, W) \to (A_1, W_1) \to \cdots \to (A_{2t}, W_{2t}) = (\mathcal{A}, \mathcal{W}), \tag{3.9}$$
we conclude Algorithm 2 to reduce AW into the following forms costing $O(2(np + p^2)t)$ operations. The complexity of the reduction is very preferable when t is small.

- If $t = n$, then
$$\mathcal{A}\mathcal{W} = \begin{bmatrix} x & & & 0 & \dots & 0 \\ & \ddots & & \vdots & \dots & \vdots \\ & & x & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} y & y & & & & \\ & \ddots & \ddots & & & \\ & & y & y & & \\ & & & y & \cdots & y \\ & & & & \ddots & \vdots \\ & & & & & y \end{bmatrix}. \tag{3.10}$$

- If $t = p$, then

$$\mathcal{A}\mathcal{W} = \begin{bmatrix} x & & \\ & \ddots & \\ & & x \\ 0 & \dots & 0 \\ \vdots & \dots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} y & y & & \\ & \ddots & \ddots & \\ & & y & y \\ & & & y \end{bmatrix}. \tag{3.11}$$

Algorithm 2 The algorithm reduces AW into a bidiagonal matrix $\mathcal{A}\mathcal{W}$ of (3.10) or (3.11) for $A \in \mathbb{R}^{n\times p}$ of (3.1) and $W \in \mathbb{R}^{p\times p}$ of (3.7).

Set $t = \min\{n, p\}$. Denote $A_0 = A$ and $W_0 = W$.
for $i = 1 : t$ **do**
  Compute the QR factorization $F_i^{-1} = Q_i^T X_i^{-1}$ by (2.1).
  Compute the BF form $Q_i A_{2i-2} =: F_{i+1}^{-1} \cdots F_t^{-1} H C_t^{-1} \cdots C_i^{-1}$ by (2.6).
  Set $A_{2i-1} = F_{i+1}^{-1} \cdots F_t^{-1} H C_t^{-1} \cdots C_{i+1}^{-1}$ and $W_{2i-1} = C_i^{-1} W_{2i-2}$.
  Compute the BF form $W_{2i-1} =: DU_p^{-1} \cdots U_i^{-1} E_i^T(c_{i-1,i}) \cdots E_2^T(c_{12})$ by (2.6).
  Rewrite $U_i = U'_{i+1} E_{i+1}^T(-c_{i,i+1})$, where U'_{i+1} is obtained from U_i by setting its $(i, i+1)$th entry $-c_{i,i+1} = 0$.
  Compute the QR factorization $(U'_{i+1})^{-1} = Y_{i+1}^{-1} P_i^T$ by the transposing version of (2.1).
  Compute the BF form $W_{2i-1} P_i =: \bar{Y}_{i+1}^{-1} DU_p^{-1} \cdots U_{i+1}^{-1} E_{i+1}^T(c_{i,i+1}) \cdots E_2^T(c_{12})$ by the transposing version of (2.6).
  Set $W_{2i} = DU_p^{-1} \cdots U_{i+1}^{-1} E_{i+1}^T(c_{i,i+1}) \cdots E_2^T(c_{12})$, and $A_{2i} = A_{2i-1} \bar{Y}_{i+1}^{-1}$.
  Compute the BF form $A_{2i} =: F_{i+1}^{-1} \cdots F_t^{-1} H C_t^{-1} \cdots C_{i+1}^{-1}$ by the transposing version of (2.6).
end for

3.3. **The qd-type method.** All the singular values of a bidiagonal matrix can be accurately computed by the LAPACK routine `DLASQ1`. Now, we have the qd-type method, i.e., Algorithm 3, to compute the GSVs of a BF matrix pair (A, B) of full rank with one of them being full column rank in $O(2(np + p^2)t + mp^2)$ operations.

Algorithm 3 The algorithm computes the GSVs of a BF matrix pair (A, B) with $A \in \mathbb{R}^{n\times p}$ of full rank and $B \in \mathbb{R}^{m\times p}$ of full column rank.

1: Compute the factor R of (3.5) for the thin QR factorization of B by Algorithm 1, and obtain $W = R^{-1}$ of (3.7) by (3.8).
2: Reduce AW into the bidiagonal matrix $\mathcal{A}\mathcal{W}$ of (3.10) or (3.11) by Algorithm 2.
3: The GSVs of (A, B) are computed as the singular values of $\mathcal{A}\mathcal{W}$ by the `DLASQ1`.

The following result shows that stages 1 and 2 of Algorithm 3 can be performed without any subtraction of like-signed numbers.

Theorem 3.2. *Let $A \in \mathbb{R}^{n\times p}$ and $B \in \mathbb{R}^{m\times p}$ be as in* (3.1) *of full rank and full column rank satisfying the sign regularity* (1.5) *with the sign sequences*

$$\begin{cases}(s_1(A),\dots,s_t(A);r_2(A),\dots,r_n(A);c_2(A),\dots,c_p(A)),\\(s_1(B),\dots,s_p(B);r_2(B),\dots,r_m(B);c_2(B),\dots,c_p(B)),\end{cases} \quad t=\min\{n,p\}, \tag{3.12}$$

respectively, and

$$\begin{cases}c_i(A)c_i(B)=-1 \text{ or } 0,\ \forall\ 2\le i\le p,\\ r_i(A)c_i(B)=-s_{i-1}(A)s_i(A) \text{ or } 0,\ \forall\ 2\le i\le t,\\ c_i(A)r_i(B)=-s_{i-1}(B)s_i(B) \text{ or } 0,\ \forall\ 2\le i\le p,\\ r_i(A)r_i(B)=-s_{i-1}(A)s_i(A)s_{i-1}(B)s_i(B) \text{ or } 0,\ \forall\ 2\le i\le t.\end{cases} \tag{3.13}$$

Then stages 1 *and* 2 *of Algorithm* 3 *applied to* (A,B) *are subtraction-free.*

Proof. According to Theorem 3.1, stage 1 of Algorithm 3 is subtraction-free, i.e., the factor $R =: \mathcal{BF}(u_{ij}) \in \mathbb{R}^{p\times p}$ of (3.5) for B is computed in a subtraction-free way satisfying (3.6), and then, $W = R^{-1} =: \mathcal{BF}(w_{ij})$ is obtained by the formula (3.8) in a subtraction-free manner with

$$\begin{cases}\operatorname{sign}(w_{ii})=s_i(B),\ \forall\ 1\le i\le p,\\ \operatorname{sign}(w_{1i},\dots,w_{i-1,i})=c_i(W),\ \forall\ 2\le i\le p,\end{cases}$$

where

$$c_i(W)=-c_i(B)\cdot s_{i-1}(B)s_i(B) \text{ or } -r_i(B), \text{ whenever } c_i(W)\ne 0.$$

Thus, we have by the fact (3.13) that

$$\begin{cases}c_i(A)c_i(W)=s_{i-1}(B)s_i(B) \text{ or } 0,\ \forall\ 2\le i\le p,\\ r_i(A)c_i(W)=s_{i-1}(A)s_i(A)s_{i-1}(B)s_i(B) \text{ or } 0,\ \forall\ 2\le i\le t.\end{cases} \tag{3.14}$$

Now, we use the fact (3.14) of (A,W) to inductively prove that stage 2 of Algorithm 3 is subtraction-free. Consider that the operations of stage 2 are performed by Algorithm 2 in the procedure (3.9) as follows: first, because of the sign sequence (3.12) of A, we have by Lemma 2.1 and Theorem 2.5 that the orthogonal transformation in line 4 of Algorithm 2,

$$\bar{A}=Q_1A=F_2^{-1}\cdots F_t^{-1}HC_t^{-1}\cdots C_1^{-1}=:\mathcal{BF}(\bar{\alpha}_{ij}),$$

is performed in a subtraction-free manner with

$$\begin{cases}\operatorname{sign}(\bar{\alpha}_{ii})=s_i(A),\ \forall\ 1\le i\le t,\\ \operatorname{sign}(\bar{\alpha}_{i1},\dots,\bar{\alpha}_{i,i-1})=r_i(\bar{A}),\ \forall\ 2\le i\le n,\\ \operatorname{sign}(\bar{\alpha}_{1i},\dots,\bar{\alpha}_{i-1,i})=c_i(\bar{A}),\ \forall\ 2\le i\le p,\end{cases}$$

where Q_1 is as in (2.1), and

$$\begin{cases}r_i(\bar{A})=r_i(A), \text{ whenever } r_i(\bar{A})\ne 0,\\ c_i(\bar{A})=c_i(A) \text{ or } r_i(A)s_{i-1}(A)s_i(A), \text{ whenever } c_i(\bar{A})\ne 0.\end{cases} \tag{3.15}$$

Thus, for line 5 of Algorithm 2, let

$$A_1=F_2^{-1}\cdots F_t^{-1}HC_t^{-1}\cdots C_2^{-1},\quad W_1=C_1^{-1}W.$$

Then by the facts (3.14) and (3.15),

$$\begin{cases} r_i(A_1) = r_i(A), \text{ if } r_i(A_1) \neq 0, \ \forall\, 2 \le i \le n, \\ c_i(A_1) = c_i(A) \text{ or } r_i(A)s_{i-1}(A)s_i(A), \text{ if } c_i(A_1) \neq 0, \ \forall\, 2 \le i \le p, \\ c_i(A_1)c_i(W) = s_{i-1}(B)s_i(B), \text{ if } c_i(A_1)c_i(W) \neq 0, \ \forall\, 2 \le i \le p, \end{cases} \tag{3.16}$$

and further, for line 6 of Algorithm 2, by Theorem 2.5, the BF form of

$$W_1 = C_1^{-1} W =: \mathcal{BF}(\bar{w}_{ij})$$

is computed in a subtraction-free way with

$$\begin{cases} \operatorname{sign}(\bar{w}_{ii}) = \operatorname{sign}(w_{ii}) = s_i(B), \ \forall\, 1 \le i \le p, \\ \operatorname{sign}(\bar{w}_{1i}, \ldots, \bar{w}_{i-1,i}) = c_i(W_1), \ \forall\, 2 \le i \le p, \end{cases} \tag{3.17}$$

where

$$c_i(W_1) = c_i(W) \text{ or } c_i(\bar{A})s_{i-1}(B)s_i(B), \text{ whenever } c_i(W_1) \neq 0.$$

Hence, we get that (A, W) is reduced to be (A_1, W_1) in a subtraction-free way, and by (3.15), (3.16), and (3.17), the fact (3.14) is true for (A_1, W_1), i.e.,

$$\begin{cases} c_i(A_1)c_i(W_1) = s_{i-1}(B)s_i(B) \text{ or } 0, \ \forall\, 2 \le i \le p, \\ r_i(A_1)c_i(W_1) = s_{i-1}(A)s_i(A)s_{i-1}(B)s_i(B) \text{ or } 0, \ \forall\, 2 \le i \le t. \end{cases}$$

Therefore, applying an induction argument on the remaining reductions of (A_1, W_1) in the procedure (3.9), we conclude that stage 2 is subtraction-free. □

Remark 3.3. As shown in numerical experiments later, the sign regularity (1.5) with (3.13) is satisfied for many structured matrix pairs.

4. Perturbation and error analysis

In this section, we provide a strong perturbation result to show that all the GSVs of a BF matrix pair are accurately determined by the BF generators. We then present an error analysis to illustrate that Algorithm 3 computes the GSVs to high relative accuracy as predicted by the strong perturbation result.

In what follows, let $\langle n \rangle =: \{1, 2, \ldots, n\}$, and denote by $Q_{k,n}$ the set of strictly increasing sequences of k positive integer numbers less than or equal to n. The absolute value operation $|\cdot|$ is interpreted componentwise.

Lemma 4.1. *Let $A =: \mathcal{BF}(\alpha_{ij}) \in \mathbb{R}^{n\times p}$ and $B =: \mathcal{BF}(\beta_{ij}) \in \mathbb{R}^{m\times p}$ be as in Theorem 3.2, and let $\tilde{A} =: \mathcal{BF}(\tilde{\alpha}_{ij})$ and $\tilde{B} =: \mathcal{BF}(\tilde{\beta}_{ij})$ be obtained from A and B only by replacing one of the generators α_{ij} or β_{ij} with*

$$\tilde{\alpha}_{ij} = \alpha_{ij}(1+\epsilon_{ij}) \text{ or } \tilde{\beta}_{ij} = \beta_{ij}(1+\epsilon'_{ij}); \tag{4.1}$$

here, $|\epsilon_{ij}|, |\epsilon'_{ij}| \le \epsilon$ and $8\epsilon < 1$. Then for the descending-ordered GSVs $\sigma_i(A, B)$ and $\sigma_i(\tilde{A}, \tilde{B})$ of (A, B) and $(\tilde{A}, \tilde{B})$, respectively,

$$|\sigma_i(\tilde{A}, \tilde{B}) - \sigma_i(A, B)| \le \frac{8\epsilon}{1-8\epsilon}\sigma_i(A, B), \quad \forall\, 1 \le i \le p.$$

Proof. By the sign sequence (3.12) of A and B, let

$$\begin{cases} Z = \operatorname{diag}(1, \prod_{j=2}^{2} z_j, \ldots, \prod_{j=2}^{p} z_j) \in \mathbb{R}^{p\times p}, \\ \bar{Z} = \operatorname{diag}(1, \prod_{j=2}^{2} \bar{z}_j, \ldots, \prod_{j=2}^{n} \bar{z}_j) \in \mathbb{R}^{n\times n}, \end{cases}$$

where for all $2 \le j \le p$,

$$z_j = \begin{cases} c_j(A), \text{ if } c_j(A) \neq 0, \\ r_j(A)s_{j-1}(A)s_j(A), \text{ if } c_j(A) = 0 \text{ but } r_j(A) \neq 0, \\ z'_j = \begin{cases} -c_j(B), \text{ if } c_j(B) \neq 0, \\ -r_j(B)s_{j-1}(B)s_j(B), \text{ if } c_j(B) = 0 \text{ but } r_j(B) \neq 0, \\ 1, \text{ otherwise}, \end{cases} & \text{otherwise}, \end{cases}$$

and for all $2 \le j \le n$,

$$\bar{z}_j = \begin{cases} r_j(A), \text{ if } r_j(A) \neq 0, \\ c_j(A)s_{j-1}(A)s_j(A), \text{ if } r_j(A) = 0 \text{ but } c_j(A) \neq 0, \\ \bar{z}'_j = \begin{cases} z'_j s_{j-1}(A)s_j(A), \text{ if } j \le t = \min\{n,p\}, \\ 1, \text{ if } j > t, \end{cases} & \text{otherwise}. \end{cases}$$

Consider that

$$A = F_1^{-1} \cdots F_t^{-1} H C_t^{-1} \cdots C_1^{-1}$$

is as in (3.1). Then

$$\bar{Z} F_i^{-1} \bar{Z} = |F_i^{-1}|, \ \bar{Z} H Z = s_1(A)|H|, \ Z C_i^{-1} Z = |C_i^{-1}|, \ \forall\ 1 \le i \le t,$$

and so,

$$\begin{aligned} ZA^TAZ &= (|F_1^{-1}| \cdots |F_t^{-1}||H||C_t^{-1}| \cdots |C_1^{-1}|)^T \\ &\quad (|F_1^{-1}| \cdots |F_t^{-1}||H||C_t^{-1}| \cdots |C_1^{-1}|) \end{aligned}$$

is a product of nonnegative bidiagonal matrices. Thus, if the perturbed parameter is α_{ij} as in (4.1), then by the Cauchy-Binet identity, for any $\alpha, \beta \in Q_{k,p}$ and $k \in \langle p \rangle$,

$$\begin{cases} \det((ZA^TAZ)[\alpha|\beta]) \ge 0, \\ \det((Z\tilde{A}^T\tilde{A}Z)[\alpha|\beta]) = \det((ZA^TAZ)[\alpha|\beta]) \cdot (1+\delta_1) \ge 0, \ |\delta_1| \le \frac{2\epsilon}{1-2\epsilon}. \end{cases}$$

Similarly, by the sign sequence (3.12) of B, let

$$\begin{cases} V = \mathrm{diag}(1, \prod_{j=2}^{2} v_j, \ldots, \prod_{j=2}^{p} v_j) \in \mathbb{R}^{p\times p}, \\ \bar{V} = \mathrm{diag}(1, \prod_{j=2}^{2} \bar{v}_j, \ldots, \prod_{j=2}^{m} \bar{v}_j) \in \mathbb{R}^{m\times m}, \end{cases}$$

where

$$v_j = \begin{cases} c_j(B), \text{ if } c_j(B) \neq 0, \\ r_j(B)s_{j-1}(B)s_j(B), \text{ if } c_j(B) = 0 \text{ but } r_j(B) \neq 0, \\ -z_j, \text{ otherwise}, \end{cases} \quad \forall\ 2 \le j \le p,$$

and

$$\bar{v}_j = \begin{cases} r_j(B), \text{ if } r_j(B) \neq 0, \\ c_j(B)s_{j-1}(B)s_j(B), \text{ if } r_j(B) = 0 \text{ but } c_j(B) \neq 0, \\ \bar{v}'_j = \begin{cases} -z_j \cdot s_{j-1}(B)s_j(B), \text{ if } j \le p, \\ 1, \text{ if } j > p, \end{cases} & \text{otherwise}, \end{cases} \quad \forall\ 2 \le j \le m.$$

Consider that

$$B = L_1^{-1} \cdots L_p^{-1} D U_p^{-1} \cdots U_1^{-1}$$

is as in (3.1). Then

$$\bar{V} B V = s_1(B) \cdot |L_1^{-1}| \cdots |L_p^{-1}||D||U_p^{-1}| \cdots |U_1^{-1}|,$$

and so,

$$\begin{aligned} VB^TBV &= (|L_1^{-1}|\cdots|L_p^{-1}||D||U_p^{-1}|\cdots|U_1^{-1}|)^T \\ &\quad (|L_1^{-1}|\cdots|L_p^{-1}||D||U_p^{-1}|\cdots|U_1^{-1}|) \end{aligned}$$

is a product of nonnegative bidiagonal matrices. Thus, if the perturbed parameter is β_{ij} as in (4.1), then by the Cauchy-Binet identity, for any $\alpha, \beta \in Q_{k,p}$ and $k \in \langle p \rangle$,

$$\begin{cases} \det((VB^TBV)[\alpha|\beta]) \geq 0, \\ \det((V\tilde{B}^T\tilde{B}V)[\alpha|\beta]) = \det((VB^TBV)[\alpha|\beta]) \cdot (1+\delta_2) \geq 0, \ |\delta_2| \leq \frac{2\epsilon}{1-2\epsilon}. \end{cases}$$

Moreover, by the fact (3.13), we have $ZV = J = \mathrm{diag}(1, -1, \ldots, (-1)^{p-1}) \in \mathbb{R}^{p\times p}$ because $z_j v_j = -1$ for all $2 \leq j \leq p$, and thus, $ZB^TBZ = J(VB^TBV)J$ with for any $\alpha, \beta \in Q_{k,p}$ and $k \in \langle p \rangle$, by [2, the equality (1.32)],

$$\det((ZB^TBZ)^{-1}[\alpha|\beta]) = \frac{\det((VB^TBV)[\beta'|\alpha'])}{\det(VB^TBV)} \geq 0, \ \alpha' = \langle p \rangle \setminus \alpha, \ \beta' = \langle p \rangle \setminus \beta.$$

Now, denote $M = (ZB^TBZ)^{-1}(ZA^TAZ)$ and $\tilde{M} = (Z\tilde{B}^T\tilde{B}Z)^{-1}(Z\tilde{A}^T\tilde{A}Z)$. Then by the Cauchy-Binet identity, we have that for any $\alpha, \beta \in Q_{k,p}$ and $k \in \langle p \rangle$,

$$\begin{cases} \det(M[\alpha|\beta]) \geq 0, \\ \det(\tilde{M}[\alpha|\beta]) = \det(M[\alpha|\beta]) \cdot (1+\delta_3) \geq 0, \ |\delta_3| \leq \frac{4\epsilon}{1-4\epsilon}. \end{cases}$$

In particular, M has the same eigenvalues as those of $(B^TB)^{-1}(A^TA)$. Denote by $\lambda_i(M)$ and $\lambda_i(\tilde{M})$ $(1 \leq i \leq p)$ the descending-ordered eigenvalues of M and $\tilde{M}$, respectively. Then by the analogous argument as that of [27, Theorem 7.2],

$$\begin{cases} \lambda_i(M) \geq 0, \\ \lambda_i(\tilde{M}) = \lambda_i(M)(1+\delta_4) \geq 0, \ |\delta_4| \leq \frac{8\epsilon}{1-8\epsilon}, \end{cases} \qquad \forall\ 1 \leq i \leq p.$$

Notice that $\sigma_i(A, B) = \sqrt{\lambda_i(M)}$ for all $1 \leq i \leq p$. So,

$$\sigma_i(\tilde{A}, \tilde{B}) = \sigma_i(A, B)(1+\delta_5), \ |\delta_5| = |\sqrt{1+\delta_4} - 1| \leq \frac{8\epsilon}{1-8\epsilon}.$$

The result is true. $\square$

Using Lemma 4.1, we immediately get the following result.

Theorem 4.2. *Let $A =: \mathcal{BF}(\alpha_{ij}) \in \mathbb{R}^{n\times p}$ and $B =: \mathcal{BF}(\beta_{ij}) \in \mathbb{R}^{m\times p}$ be as in Theorem 3.2, and let $\tilde{A} =: \mathcal{BF}(\tilde{\alpha}_{ij})$ and $\tilde{B} =: \mathcal{BF}(\tilde{\beta}_{ij})$ be obtained from A and B by replacing each of the generators α_{ij} and β_{ij} with*

$$\tilde{\alpha}_{ij} = \alpha_{ij}(1+\epsilon_{ij}), \ \tilde{\beta}_{ij} = \beta_{ij}(1+\epsilon'_{ij}),$$

with $|\epsilon_{ij}|, |\epsilon'_{ij}| \leq \epsilon$ and $8(n+m)p\epsilon < 1$. Then for the descending-ordered GSVs $\sigma_i(A, B)$ and $\sigma_i(\tilde{A}, \tilde{B})$ of (A, B) and $(\tilde{A}, \tilde{B})$, respectively,

$$|\sigma_i(\tilde{A}, \tilde{B}) - \sigma_i(A, B)| \leq \frac{8(n+m)p\epsilon}{1-8(n+m)p\epsilon}\sigma_i(A, B), \quad \forall\ 1 \leq i \leq p.$$

Proof. Consider that the perturbed matrix pair $(\tilde{A}, \tilde{B})$ is obtained from the matrix pair (A, B) through a sequence of $(n+m)p$ "only-one-generator" perturbations of (4.1). Therefore, we apply Lemma 4.1 repeatedly and accumulate the relative perturbation errors to conclude that the result is true. $\square$

Now, we are ready to illustrate the high relative accuracy of Algorithm 3. Recall that μ is the unit roundoff.

Theorem 4.3. *Let $\sigma_i(A,B)$ and $\hat{\sigma}_i(A,B)$ $(1 \le i \le p)$ be the exact and computed descending-ordered GSVs of a BF matrix pair $(A,B) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{m\times p}$ as in Theorem 3.2 by Algorithm 3. Then for all $1 \le i \le p$,*

$$|\hat{\sigma}_i(A,B) - \sigma_i(A,B)| \le \frac{O(16(n+m+p)p^2)\mu}{1 - O(16(n+m+p)p^2)\mu}\sigma_i(A,B).$$

Proof. Denote by $\hat{R}$ the computed quantity of the factor R by applying stage 1 of Algorithm 3 to B in the procedure (3.4) as follows:

$$B_0 = B \to B_1 \to \cdots \to B_p = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix},$$

costing $O(mp^2)$ subtraction-free arithmetic operations by Theorem 3.2, where every single operation causes at most μ relative perturbations in at most one generator of the intermediate BF matrices B_i $(1 \le i \le p)$. So, by using Lemma 4.1 repeatedly, the descending-ordered GSVs $\sigma_i(A,\hat{R})$ and $\sigma_i(A,B)$ satisfy that for all $1 \le i \le p$,

$$|\sigma_i(A,\hat{R}) - \sigma_i(A,B)| \le \frac{O(8mp^2)\mu}{1 - O(8mp^2)\mu}\sigma_i(A,B). \tag{4.2}$$

Further, let $\hat{\mathcal{A}}$ and $\hat{\mathcal{W}}$ be the computed quantities of $\mathcal{A}$ and $\mathcal{W}$ by applying stage 2 of Algorithm 3 to A and $W = \hat{R}^{-1}$ in the procedure (3.9) as follows:

$$(A_0, W_0) = (A, W) \to (A_1, W_1) \to \cdots \to (A_{2t}, W_{2t}) = (\hat{\mathcal{A}}, \hat{\mathcal{W}}),\ t = \min\{n, p\},$$

costing $O(2(np+p^2)t)$ subtraction-free arithmetic operations by Theorem 3.2, where every single operation causes at most μ relative perturbations in at most one generator of the intermediate BF pairs (A_i, W_i) and so (A_i, W_i^{-1}) (by the converting formula (3.8) for the generators between W_i and W_i^{-1}) for all $1 \le i \le 2t$. So, by using Lemma 4.1 repeatedly, the descending-ordered GSVs $\sigma_i(\hat{\mathcal{A}}, \hat{\mathcal{W}}^{-1})$ and $\sigma_i(A, W^{-1}) = \sigma_i(A, \hat{R})$ satisfy that for all $1 \le i \le p$,

$$|\sigma_i(\hat{\mathcal{A}}, \hat{\mathcal{W}}^{-1}) - \sigma_i(A,\hat{R})| \le \frac{O(16(np+p^2)t)\mu}{1 - O(16(np+p^2)t)\mu}\sigma_i(A,\hat{R}). \tag{4.3}$$

Finally, the LAPACK routine `DLASQ1` computes all the singular values of $\hat{\mathcal{A}}\hat{\mathcal{W}}$ with a relative error not exceeding $O(t^2)\mu$. Notice that the singular values of $\hat{\mathcal{A}}\hat{\mathcal{W}}$ are just the GSVs of $(\hat{\mathcal{A}}, \hat{\mathcal{W}}^{-1})$. Therefore, combining with the facts (4.2) and (4.3), we conclude that the result is true. □

5. Numerical experiments

In this section, we provide numerical experiments to confirm the theoretical high relative accuracy of our qd-type method by measuring the relative errors

$$\texttt{Rel. error}(\hat{\sigma}_i(A,B)) = |\hat{\sigma}_i(A,B) - \sigma_i(A,B)|/\sigma_i(A,B)$$

of the computed GSVs $\hat{\sigma}_i(A,B)$, where $\sigma_i(A,B)$ are the exact descending-ordered GSVs of (A,B) by *Mathematica*. All the tests are conducted in MATLAB 7.0 with double precision arithmetic. We have done tests for several important classes of structured matrices that may have huge condition numbers: Cauchy, Vandermonde, Bernstein-Vandermonde, and Cauchy-Vandermonde matrices, whose accurate generators have been available in the literature [11, 27, 28, 30].

Example 5.1. In this example, consider the Cauchy matrix $A = [1/(x_i + y_j)]_{i,j=1}^{n,n}$, whose nodes $0 < x_1 < \cdots < x_n$ and $0 < y_1 < \cdots < y_n$ are randomly chosen by the Matlab command `rand`. Accurate generators of this class are referred to [11, 27]. Set $n = 30$. We test our qd-type method for the matrix pair (A, A^{-1}). Clearly, the GSVs of (A, A^{-1}) are the singular values of A^2. Thus we compute the GSVs by Algorithm 3 and Matlab command $\texttt{svd}(A^2)$, respectively. The relative errors for all the computed GSVs are reported in Table 1. As observed, all the GSVs are computed by Algorithm 3 to high relative accuracy, whereas only the larger GSVs are accurately computed by $\texttt{svd}(A^2)$.

TABLE 1. The relative errors of the computed GSVs in Example 5.1.

i	$\sigma_i(A, A^{-1})$	Rel. error$(\hat{\sigma}_i(A, A^{-1}))$ by Alg. 3	Rel. error$(\hat{\sigma}_i(A, A^{-1}))$ by $\texttt{svd}(A^2)$
1	3.837366680349300e+003	1.185050553587103e-015	3.555151660761309e-016
2	2.396907010833898e+002	1.660072460997237e-015	2.252955482781965e-015
3	5.913144813720718e+000	4.506122111059503e-016	6.158366885114654e-015
4	9.409100772278690e-002	2.064905393316875e-015	5.883505438529296e-013
5	9.663456193152294e-004	2.243922155416383e-016	5.001612727536901e-011
6	4.929144211589063e-006	1.718417865037941e-016	1.124449256061839e-008
7	4.239837425333318e-008	1.716855780123311e-015	1.653600242308265e-006
8	1.596870650434085e-010	9.712518970245468e-016	1.536763088532686e-004
9	2.887869565574453e-013	2.447546037064270e-015	3.129882760473304e-002
10	5.878728336329913e-016	1.677362985856005e-016	2.516103525102535e+002
11	1.035646302272118e-018	1.859640632291077e-016	1.205803073286590e+005
12	1.789565746629741e-021	2.312147582555705e-015	6.747057190093108e+007
13	2.560428927844945e-024	1.004282471760713e-015	4.210220160176415e+010
14	3.367196077234422e-027	2.769973106587413e-015	2.390646285428661e+013
15	4.873617474945300e-030	4.312910701943872e-015	1.378481105657451e+016
16	6.393941600712662e-033	5.992665757161587e-015	9.983774927579791e+018
17	1.367190549613422e-035	6.255782761746982e-015	4.369652449293529e+021
18	8.537046536547042e-039	5.503330919945078e-015	6.468095556006819e+024
19	4.468708203359809e-042	7.272588298038238e-015	1.101053890336203e+028
20	2.378900570143876e-045	6.147413677776880e-015	1.712292493532668e+031
21	6.632531314286597e-049	3.550530995422411e-015	5.417006824232429e+034
22	9.575497134856062e-053	3.098914739108783e-015	3.307246372772586e+038
23	1.426945575823943e-056	4.600993509061067e-015	2.049450887374576e+042
24	7.695272554693393e-061	8.978154828137640e-016	2.591410543536386e+046
25	1.179164531573111e-064	4.863575461743630e-015	1.480623720467055e+050
26	2.129762726634752e-069	8.217674009088239e-015	7.636521162694257e+054
27	8.592193561200744e-074	1.755167061586437e-014	1.578183937652539e+059
28	2.087072464764095e-078	1.079597011616434e-014	4.772265262669524e+063
29	1.181300663762475e-082	9.412490556789930e-015	7.215855358771275e+067
30	7.669643627615243e-090	1.091509604449015e-014	4.740003201850433e+074

Example 5.2. Consider the following subclasses of Bernstein-Vandermonde matrices:

$$\begin{cases} \mathcal{BV}_{n,p}^{-} = \{A \mid A = [\binom{p-1}{j-1}(1-x_i)^{p-j} x_i^{j-1}]_{i,j=1}^{n,p},\ x_1 > \cdots > x_n > 1\}, \\ \mathcal{BV}_{m,p}^{+} = \{A \mid A = [\binom{p-1}{j-1}(1-x_i)^{p-j} x_i^{j-1}]_{i,j=1}^{m,p},\ 0 < x_1 < \cdots < x_m < 1\}. \end{cases}$$

Accurate generators of this class are referred to [28]. Test the matrix pair $(A, B) \in \mathcal{BV}_{n,p}^{-} \times \mathcal{BV}_{m,p}^{+}$ whose nodes are randomly chosen by the commands `100*rand` and `rand`, respectively. Set $n = 50$ and $m = p = 30$. We compute the GSVs by Algorithm 3 and Matlab command `gsvd`, respectively. The relative errors for all the computed GSVs are reported in Table 2, which confirms the high relative accuracy of Algorithm 3 in contrast to the poor results of the `gsvd`.

TABLE 2. The relative errors of the computed GSVs in Example 5.2.

i	$\sigma_i(A,B)$	`Rel. error`$(\hat{\sigma}_i(A,B))$ by Alg. 3	`Rel. error`$(\hat{\sigma}_i(A,B))$ by `gsvd`
1	4.937608271584825e+078	8.048215426079582e-014	9.999999999999973e-001
2	1.731124511681691e+073	2.882690324097263e-014	9.999999999999997e-001
3	1.241071088585855e+069	9.523525257844666e-014	9.999999999998999e-001
4	1.561008155625254e+065	6.651155276917867e-014	9.999999998030380e-001
5	1.540478266557079e+061	2.260651413139237e-014	9.999997677042901e-001
6	2.774133526921384e+057	7.913208157346814e-015	9.994937343919852e-001
7	8.129536082257777e+053	9.836521471494824e-015	8.784194767320852e-001
8	2.620722341180145e+050	7.766486119354005e-015	1.350711309376432e+002
9	5.506603261857968e+046	5.893261929840426e-015	1.024610024928824e+005
10	1.129857243619833e+043	1.533924807568592e-015	8.216607171535759e+007
11	6.408133061305931e+039	7.546196734986392e-016	1.806319188299874e+010
12	3.583742285853278e+036	8.235745810864829e-016	8.294514756066526e+012
13	3.618158076344211e+033	1.911892427494643e-015	2.069551378994854e+015
14	3.827323378921773e+030	1.088444649924329e-014	2.464247466921457e+017
15	5.208150484790644e+027	1.688909152662778e-015	1.004818477252024e+020
16	7.496948905588700e+024	1.718686101541226e-015	8.455405028440791e+021
17	7.697330104016026e+021	8.173556174650063e-016	2.043266652484557e+024
18	1.609530600608101e+019	0	7.639040026385489e+026
19	6.583508052634440e+016	1.336673385928132e-015	1.427562886206579e+028
20	2.096740625617186e+014	1.788490170974853e-015	1.130763335846495e+030
21	1.163538056860335e+012	1.888434685951787e-015	1.521552546972157e+031
22	6.833257775524699e+009	6.978181913626265e-016	7.162303966193643e+032
23	4.042177600952068e+007	7.372838437548085e-016	4.271709787681572e+034
24	4.708130589378171e+005	6.181610706039828e-015	1.933467885390162e+035
25	7.198922124946288e+003	1.010700975493288e-015	9.017458198313638e+035
26	1.066174793277250e+002	1.199594036955996e-015	5.805921846782484e+036
27	4.793260604068320e+000	3.705946715879693e-016	1.892518792628565e+037
28	1.000356867664248e-001	1.109826963269055e-015	1.866284243690280e+038
29	8.563046833214733e-004	5.064562619370722e-016	2.576151990567913e+038
30	4.856169584085631e-006	0	5.082430545043722e+039

Example 5.3. Consider the following subclasses of Cauchy-Vandermonde matrices:

$$\begin{cases} \mathcal{CV}^-_{n,p,l} = \{A |\ A = (a_{ij}) \in \mathbb{R}^{n\times p},\ a_{ij} = \begin{cases} (-1)^{j-1}/(x_i + y_j),\ \forall\ 1 \le j \le l, \\ (-1)^{j-1}(x_i)^{j-l-1},\ \forall\ l+1 \le j \le p, \end{cases} \}, \\ \mathcal{CV}^+_{m,p,l} = \{A |\ A = (a_{ij}) \in \mathbb{R}^{m\times p},\ a_{ij} = \begin{cases} 1/(x_i + y_j),\ \forall\ 1 \le j \le l, \\ (x_i)^{j-l-1},\ \forall\ l+1 \le j \le p, \end{cases} \}, \end{cases}$$

where the distinct nodes x_i and y_j are ascending-ordered, respectively. Accurate generators of this class are referred to [30]. Test the matrix pair $(A,B) \in \mathcal{CV}^-_{n,p,l} \times \mathcal{CV}^+_{m,p,l}$ whose nodes are chosen by the Matlab command `rand`. Set $n = 30$, $m = p = 20$, and $l = 15$. We compute the GSVs by Algorithm 3 and Matlab command `gsvd`, respectively. All the computed GSVs are reported in Table 3. As expected, Algorithm 3 is able to compute each GSV, no matter how tiny it is, to nearly full relative accuracy with $\max_i$ `Rel. error`$(\hat{\sigma}_i(A,B)) = 7.6465e-014$, whereas the GSVs computed by the command `gsvd` have no relative accuracy at all with $\max_i$ `Rel. error`$(\hat{\sigma}_i(A,B)) = 1.5505e+011$.

Example 5.4. Consider the matrix pair $(A,B) \in \mathcal{V}^-_{n,p} \times \mathcal{C}^-_{m,p}$, where

$$A = \left[\left(-\frac{i^2}{n} \right)^{j-1} \right]_{i,j=1}^{n,p}, \quad B = \left[\frac{-m}{i+j} \right]_{i,j=1}^{m,p}.$$

Then A and B are Vandermonde and Cauchy matrices, respectively, whose accurate generators are referred to [11, 27]. Set $n = 70$ and $m = p = 30$. We compute the GSVs by Algorithm 3 and Matlab command `gsvd`, respectively. All the computed GSVs are reported in Table 4. As observed, all the GSVs are computed by Algorithm 3 to high relative accuracy with $\max_i$ `Rel. error`$(\hat{\sigma}_i(A,B)) = 8.1858e-015$, whereas the command `gsvd` even fails to provide a correct digit.

TABLE 3. The computed GSVs in Example 5.3.

i	$\sigma_i(A,B)$	$\hat{\sigma}_i(A,B)$ by Algorithm 3	$\hat{\sigma}_i(A,B)$ by `gsvd`
1	2.431749245122608e+036	2.431749245122794e+036	1.346449381280061e+017
2	4.016675851236368e+030	4.016675851236356e+030	6.212733620493904e+016
3	6.056444061916870e+026	6.056444061916883e+026	3.888308180611378e+016
4	6.335863677017518e+023	6.335863677017476e+023	3.462798846815255e+015
5	5.429394208672166e+021	5.429394208672107e+021	7.395069893099605e+014
6	3.340941485378802e+019	3.340941485378799e+019	7.402224167195766e+013
7	1.917636094373130e+017	1.917636094373124e+017	2.479271564337629e+012
8	2.752930584765908e+015	2.752930584765910e+015	1.131229792568784e+012
9	3.562810606750646e+013	3.562810606750635e+013	1.155796434470147e+011
10	6.520162731980753e+011	6.520162731980757e+011	8.466225438598619e+009
11	1.258610950539334e+010	1.258610950539333e+010	5.214169415646864e+008
12	2.274978133354873e+008	2.274978133354870e+008	1.375693191860078e+008
13	5.701289037580025e+006	5.701289037580024e+006	4.943355724976541e+006
14	1.697547585661229e+005	1.697547585661231e+005	1.693886212706910e+005
15	5.025103266936044e+003	5.025103266936041e+003	5.017352593971893e+003
16	3.523768112989028e+002	3.523768112989026e+002	3.524640026065235e+002
17	7.375226774344139e+000	7.375226774344138e+000	7.375778072961789e+000
18	2.028150951959962e-001	2.028150951959966e-001	2.028285428486759e-001
19	5.274266569134978e-003	5.274266569134989e-003	5.274695662875262e-003
20	6.255314115873677e-005	6.255314115873677e-005	6.233364260750291e-005
21	1.943064353891533e-006	1.943064353891533e-006	1.905367461381455e-006
22	3.421700223198902e-008	3.421700223198894e-008	3.675714171087664e-008
23	8.061038131618250e-010	8.061038131618245e-010	1.319231164292686e-009
24	1.644745480143042e-011	1.644745480143039e-011	1.745345952001044e-010
25	1.329164805016835e-013	1.329164805016833e-013	2.310157137531839e-011
26	2.113016554888170e-015	2.113016554888163e-015	3.826283873157563e-013
27	2.226367423689541e-017	2.226367423689532e-017	6.512521928407069e-014
28	1.351531442821981e-019	1.351531442821977e-019	2.760664483720381e-015
29	2.677071111253502e-023	2.677071111253506e-023	6.048188612842263e-016
30	3.216405652780558e-028	3.216405652780558e-028	4.987010698694678e-017

TABLE 4. The computed GSVs in Example 5.4.

i	$\sigma_i(A,B)$	$\hat{\sigma}_i(A,B)$ by Alg. 3	$\hat{\sigma}_i(A,B)$ by `gsvd`
1	7.279538114138588e+091	7.279538114138585e+091	2.039659673822689e+018
2	1.525428576942243e+086	1.525428576942251e+086	3.874246367841285e+017
3	9.795043914718496e+080	9.795043914718475e+080	1.116439611909831e+017
4	1.254955773778056e+076	1.254955773778054e+076	1.676463393469303e+016
5	2.695891269761148e+071	2.695891269761126e+071	1.433633439307463e+015
6	8.840008901115421e+066	8.840008901115382e+066	2.618254107753066e+014
7	4.175036531392758e+062	4.175036531392748e+062	5.366967816916211e+013
8	2.731920967011077e+058	2.731920967011075e+058	2.029938031607849e+013
9	2.410328334147611e+054	2.410328334147608e+054	2.295078949611471e+012
10	2.811998619588474e+050	2.811998619588462e+050	2.475362159749992e+011
11	4.277239425666351e+046	4.277239425666361e+046	5.565475778882438e+010
12	8.397621644657097e+042	8.397621644657097e+042	9.837824968075697e+009
13	2.113563454018201e+039	2.113563454018197e+039	1.533458432169387e+009
14	6.790772383529617e+035	6.790772383529611e+035	1.538787963725168e+008
15	2.780131637033885e+032	2.780131637033884e+032	6.538078397799131e+007
16	1.450775600079850e+029	1.450775600079858e+029	6.652659651612685e+006
17	9.673992979771721e+025	9.673992979771762e+025	8.280445435628118e+005
18	8.282080389830667e+022	8.282080389830660e+022	8.759852303920349e+004
19	9.169232413788435e+019	9.169232413788439e+019	3.524335419755443e+004
20	1.326064178592266e+017	1.326064178592265e+017	4.629241793432614e+003
21	2.539419187711274e+014	2.539419187711273e+014	4.425904983731892e+002
22	6.556937633182509e+011	6.556937633182512e+011	1.598115666850363e+002
23	2.338793992297006e+009	2.338793992297006e+009	7.139467970285348e+001
24	1.191362207523464e+007	1.191362207523465e+007	6.229197119065474e+001
25	9.092248270095048e+004	9.092248270095041e+004	2.484806258240705e+001
26	1.123753181866915e+003	1.123753181866913e+003	1.981694542241248e+001
27	2.626365441973276e+001	2.626365441973278e+001	9.463339926165990e+000
28	1.114303707840759e+000	1.114303707840757e+000	2.606098947208910e+000
29	3.549612570503446e-002	3.549612570503446e-002	1.080309561875296e-001
30	5.249716399554590e-004	5.249716399554601e-004	1.786361973098763e-003

ACKNOWLEDGMENTS

The author would like to thank the editor and the anonymous referees for their valuable comments and suggestions which have helped to improve the overall presentation of the paper greatly.

References

[1] O. Alter, P. O. Brown, and D. Botstein, *Generalized singular decomposition for comparative analysis of genome-scale expression data sets of two different organisms*, Proc. Nat. Acad. Sci. USA **100** (2003), 3351–3356.

[2] T. Ando, *Totally positive matrices*, Linear Algebra Appl. **90** (1987), 165–219, DOI 10.1016/0024-3795(87)90313-2. MR884118

[3] Z. Bai and J. W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput. **14** (1993), no. 6, 1464–1486, DOI 10.1137/0914085. MR1241595

[4] A. Berenstein, S. Fomin, and A. Zelevinsky, *Parametrizations of canonical bases and totally positive matrices*, Adv. Math. **122** (1996), no. 1, 49–149, DOI 10.1006/aima.1996.0057. MR1405449

[5] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[6] D. Chu, L. De Lathauwer, and B. De Moor, *A QR-type reduction for computing the SVD of a general matrix product/quotient*, Numer. Math. **95** (2003), no. 1, 101–121, DOI 10.1007/s00211-002-0431-z. MR1993940

[7] J. Delgado and J. M. Peña, *Accurate computations with collocation matrices of q-Bernstein polynomials*, SIAM J. Matrix Anal. Appl. **36** (2015), no. 2, 880–893, DOI 10.1137/140993211. MR3361436

[8] J. Demmel, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl. **21** (1999), no. 2, 562–580, DOI 10.1137/S0895479897328716. MR1742810

[9] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl. **299** (1999), no. 1-3, 21–80, DOI 10.1016/S0024-3795(99)00134-2. MR1723709

[10] J. W. Demmel and W. Gragg, *On computing accurate singular values and eigenvalues of matrices with acyclic graphs*, Linear Algebra Appl. **185** (1993), 203–217, DOI 10.1016/0024-3795(93)90213-8. MR1213179

[11] J. Demmel, I. Dumitriu, O. Holtz, and P. Koev, *Accurate and efficient expression evaluation and linear algebra*, Acta Numer. **17** (2008), 87–145, DOI 10.1017/S0962492906350015. MR2436010

[12] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput. **11** (1990), no. 5, 873–912, DOI 10.1137/0911052. MR1057146

[13] J. Demmel and P. Koev, *The accurate and efficient solution of a totally positive generalized Vandermonde linear system*, SIAM J. Matrix Anal. Appl. **27** (2005), no. 1, 142–152, DOI 10.1137/S0895479804440335. MR2176813

[14] J. Demmel and P. Koev, *Accurate SVDs of weakly diagonally dominant M-matrices*, Numer. Math. **98** (2004), no. 1, 99–104, DOI 10.1007/s00211-004-0527-8. MR2076055

[15] F. M. Dopico and P. Koev, *Perturbation theory for the LDU factorization and accurate computations for diagonally dominant matrices*, Numer. Math. **119** (2011), no. 2, 337–371, DOI 10.1007/s00211-011-0382-3. MR2836090

[16] Z. Drmač, *A tangent algorithm for computing the generalized singular value decomposition*, SIAM J. Numer. Anal. **35** (1998), no. 5, 1804–1832, DOI 10.1137/S0036142995289883. MR1639946

[17] Z. Drmač and E. R. Jessup, *On accurate quotient singular value computation in floating-point arithmetic*, SIAM J. Matrix Anal. Appl. **22** (2000), no. 3, 853–873, DOI 10.1137/S0895479896310548. MR1799528

[18] K. V. Fernando and B. N. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math. **67** (1994), no. 2, 191–229, DOI 10.1007/s002110050024. MR1262781

[19] M. Gasca and J. M. Peña, *A matricial description of Neville elimination with applications to total positivity*, Linear Algebra Appl. **202** (1994), 33–53, DOI 10.1016/0024-3795(94)90183-X. MR1288481

[20] G. Golub, K. Sølna, and P. Van Dooren, *Computing the SVD of a general matrix product/quotient*, SIAM J. Matrix Anal. Appl. **22** (2000), no. 1, 1–19, DOI 10.1137/S0895479897325578. MR1779712

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996. MR1417720

[22] P. Howland, M. Jeon, and H. Park, *Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl. **25** (2003), no. 1, 165–179, DOI 10.1137/S0895479801393666. MR2002905

[23] R. Huang, *Rank structure properties of rectangular matrices admitting bidiagonal-type factorizations*, Linear Algebra Appl. **465** (2015), 1–14, DOI 10.1016/j.laa.2014.09.016. MR3274659

[24] R. Huang, *A periodic qd-type reduction for computing eigenvalues of structured matrix products to high relative accuracy*, J. Sci. Comput. **75** (2018), no. 3, 1229–1261, DOI 10.1007/s10915-017-0584-7. MR3798100

[25] R. Huang and D. Chu, *Relative perturbation analysis for eigenvalues and singular values of totally nonpositive matrices*, SIAM J. Matrix Anal. Appl. **36** (2015), no. 2, 476–495, DOI 10.1137/140995702. MR3340200

[26] R. Huang and D. Chu, *Computing singular value decompositions of parameterized matrices with total nonpositivity to high relative accuracy*, J. Sci. Comput. **71** (2017), no. 2, 682–711, DOI 10.1007/s10915-016-0315-5. MR3627537

[27] P. Koev, *Accurate eigenvalues and SVDs of totally nonnegative matrices*, SIAM J. Matrix Anal. Appl. **27** (2005), no. 1, 1–23, DOI 10.1137/S0895479803438225. MR2176803

[28] A. Marco and J.-J. Martínez, *A fast and accurate algorithm for solving Bernstein-Vandermonde linear systems*, Linear Algebra Appl. **422** (2007), no. 2-3, 616–628, DOI 10.1016/j.laa.2006.11.020. MR2305145

[29] A. Marco and J.-J. Martínez, *Accurate computations with Said-Ball-Vandermonde matrices*, Linear Algebra Appl. **432** (2010), no. 11, 2894–2908, DOI 10.1016/j.laa.2009.12.034. MR2639252

[30] J. J. Martínez and J. M. Peña, *Fast algorithms of Björck-Pereyra type for solving Cauchy-Vandermonde linear systems*, Appl. Numer. Math. **26** (1998), no. 3, 343–352, DOI 10.1016/S0168-9274(97)00102-5. MR1609143

[31] P. Mongelli, *Total positivity properties of Jacobi-Stirling numbers*, Adv. in Appl. Math. **48** (2012), no. 2, 354–364, DOI 10.1016/j.aam.2011.06.008. MR2873882

[32] S. Ostrovska, *On the Lupaş q-transform*, Comput. Math. Appl. **61** (2011), no. 3, 527–532, DOI 10.1016/j.camwa.2010.11.025. MR2764046

[33] V. Y. Pan, *Structured Matrices and Polynomials*: *Unified Superfast Algorithms*, Birkhäuser Boston, Inc., Boston, MA; Springer-Verlag, New York, 2001. MR1843842

[34] V. Y. Pan, *How bad are Vandermonde matrices?*, SIAM J. Matrix Anal. Appl. **37** (2016), no. 2, 676–694, DOI 10.1137/15M1030170. MR3504989

[35] C. C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput. **7** (1986), no. 4, 1126–1146, DOI 10.1137/0907077. MR857786

[36] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal. **18** (1981), no. 3, 398–405, DOI 10.1137/0718026. MR615522

[37] B. N. Parlett, *The new qd algorithms*, Acta Numerica, 1995, Acta Numer., Cambridge Univ. Press, Cambridge, 1995, pp. 459–491, DOI 10.1017/s0962492900002580. MR1352476

[38] C. H. Park and H. Park, *Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl. **27** (2005), no. 1, 87–102, DOI 10.1137/S0895479804442334. MR2176809

[39] G. M. Phillips, *Interpolation and Approximation by Polynomials*, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, vol. 14, Springer-Verlag, New York, 2003. MR1975918

[40] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Code CD-ROM v 2.06 with UNIX single-screen license, Cambridge University Press, Cambridge, 1996. MR1414682

[41] K. Rietsch, *Totally positive Toeplitz matrices and quantum cohomology of partial flag varieties*, J. Amer. Math. Soc. **16** (2003), no. 2, 363–392, DOI 10.1090/S0894-0347-02-00412-5. MR1949164

[42] G. W. Stewart, *Computing the CS decomposition of a partitioned orthonormal matrix*, Numer. Math. **40** (1982), no. 3, 297–306, DOI 10.1007/BF01396447. MR695598

[43] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal. **13** (1976), no. 1, 76–83, DOI 10.1137/0713009. MR0411152

[44] C. Van Loan, *Computing the CS and the generalized singular value decompositions*, Numer. Math. **46** (1985), no. 4, 479–491, DOI 10.1007/BF01389653. MR796639

[45] Q. Ye, *Computing singular values of diagonally dominant matrices to high relative accuracy*, Math. Comp. **77** (2008), no. 264, 2195–2230, DOI 10.1090/S0025-5718-08-02112-1. MR2429881

School of Mathematics and Computational Science, Hunan University of Science and Technology, Xiangtan 411201, Hunan, People's Republic of China

Email address: `rongh98@aliyun.com`