Exact deflation for accurate SVD computation of nonnegative bidiagonal products of arbitrary rank

Huang Rong[1] and Jungong Xue[2†]

[1]School of Mathematics and Statistics, Hunan Normal University, Changsha, 410081, Hunan, China.
[2]School of Mathematical Science, Fudan University, Shanghai, 200433, Shanghai, China.

Contributing authors: rongh98@hunnu.edu.cn; xuej@fudan.edu.cn;
†These authors contributed equally to this work.

Abstract

Dealing with zero singular values can be quite challenging, as they have the potential to cause numerous numerical difficulties. This paper presents a method for computing the singular value decomposition (SVD) of a nonnegative bidiagonal product of arbitrary rank, regardless of whether the factors are of full rank or rank-deficient, square or rectangular. A key feature of our method is its ability to exactly deflate all zero singular values with a favorable complexity, irrespective of rank deficiency and ill conditioning. Furthermore, it ensures the computation of nonzero singular values, no matter how small they may be, with high relative accuracy. Additionally, our method is well-suited for accurately computing the SVDs of arbitrary submatrices, leveraging an approach to extract their representations from the original product. We have conducted error analysis and numerical experiments to validate the claimed high relative accuracy.

Keywords: Bidiagonal products, deflation, SVDs, rank deficiency, submatrices, high relative accuracy

MSC Classification: 65F15 , 15A18

1 Introduction

Computing the singular value decomposition (SVD) of matrix products or quotients is crucial in various applications, including statistical realization, control theory, canonical correlations, and source separation [17, 19, 33]. Many algorithms have been developed to compute SVDs for two matrices [2, 22, 31], three matrices [4, 5], and even long matrix products [3, 6, 16, 33]. While these algorithms are backward stable and can compute SVDs with high absolute accuracy, they often struggle to accurately compute tiny singular values, which are important in many applications [1, 33], especially with high relative accuracy. Efforts have been made to design high-accuracy SVD algorithms for a single full-rank matrix [8–13, 15, 21, 34]. However, these algorithms may not be directly applicable to scenarios involving multiple matrices. Though not abundant, there are several algorithms designed to accurately compute SVDs for two or three full-rank matrices, as warranted by the matrix entries [9, 14]. However, the challenge remains when it comes to computing SVDs involving multiple matrices with high relative accuracy. This is particularly true when dealing with rank deficiency, where the presence of zero singular values can cause numerous numerical difficulties. This paper specifically addresses the challenge of accurately computing the SVD of a nonnegative bidiagonal product of arbitrary rank, i.e.,

$$A = B_1 B_2 \dots B_K \in \mathbb{R}^{n_0 \times n_K}, \tag{1}$$

where $B_k \in \mathbb{R}^{n_{k-1} \times n_k}$, $1 \leq k \leq K$, are nonnegative lower or upper bidiagonal that are allowed to be rank-deficient and rectangular. The proposed method aims to overcome the issues caused by rank deficiency and ill conditioning, ensuring that zero singular values are exactly identified and returned, while the nonzero singular values, regardless of their small magnitude, are computed with high relative accuracy.

As shown in Section 2 later, there are numerous scenarios where numerical computations involving the product form (1) are necessary. Indeed, such bidiagonal products of full rank are explicitly accommodated by various structured matrices, including Cauchy, Vandermonde, Bernstein-Vandermonde and Cauchy-Vandermonde totally nonnegative (TN) matrices with distinct nodes [23, 27–30]. Utilizing the full-rank bidiagonal factors as input, one can apply the algorithms in the *TNTool* [26] to address a wide range of numerical linear algebra problems, including the SVD problem, with high relative accuracy. On the other hand, if the aforementioned structured matrices contain repeated nodes, they inherently become rank-deficient. Recently, an accurate algorithm has been provided for computing eigenvalues of a singular square TN matrix with high relative accuracy (returning zero eigenvalues exactly) [25]. However, as pointed out in [25], the accurate computation for the SVD problem of rank-deficient TN matrices remains an open question. This paper addresses the challenging issue stemming from rank deficiency through our proposed method.

Dealing with zero singular values can indeed be a challenging issue in numerical computations, as they can lead to various numerical difficulties. One attractive aspect of our method is its ability to exactly deflate zero singular values. Let us provide a brief explanation of our approach. We utilize nontrivial element pairs within the factors B_k's as input. Interestingly, these elements inherently reveal the presence of zero singular values. For instance, consider the factor B_1 as in (1) that is lower bidiagonal with its $(i, i-1)$th and (i, i)th elements being zero. This implies that the ith row of A is zero, indicating a potential zero singular value that can be deflated by removing this row. The key challenge then lies in extracting the representation of the resulting submatrix from the original one, a task which our proposed method adeptly addresses.

It is worth noting that a methodology was initially presented for handling submatrix representations of full-rank TN matrices, which can be decomposed as a product of nonsingular bidiagonal factors [24]. However, this approach encounters difficulties when confronted with matrices that exhibit rank deficiency. A subsequent approach addresses the computation of representations for matrices derived from square TN matrices by removing a column and appending a zero column at the end [25]. However, a gap exists in the ability to compute the representation for arbitrary submatrices. Moreover, this capability is particularly vital when dealing with rank deficiency. Therefore, different from the aforementioned methods, we introduce the updating and downdating procedures specifically tailored to extract the representation of arbitrary submatrices from the original one for any nonnegative bidiagonal product, including general TN matrices. The extraction method offers a comprehensive solution that extends the capabilities of existing methods and enables the computation of submatrix representations in a broader range of general TN matrices.

Basing on the extracting method, our deflation method is designed to exactly deflate zero singular values. Let us briefly outline the deflation method. Given the product $A = B_1 B_2 \dots B_K \in \mathbb{R}^{n_0 \times n_K}$ as in (1) having S nontrivial element pairs in the factors $B_k \in \mathbb{R}^{n_{k-1} \times n_k}$ $(1 \le k \le K)$, one key idea is to first split the product A into two parts

$$A = A_2 A_1, \text{where } A_1 = B_{T+1} \dots B_K \in \mathbb{R}^{r \times n_K},\ A_2 = B_1 \dots B_T \in \mathbb{R}^{n_0 \times r},$$

by determining the minimum dimension among the factors, denoted as $r = \min_{0 \le k \le K}\{n_k\} = n_T$. We then compute the representations of A_1 and A_2, on which we perform the deflation for $A = A_2 A_1$ in the following process. This deflation process is based on the minimum dimension among the factors, ensuring that the deflation is carried out effectively. In what follows, we denote by $A(\alpha|:]$ (or $A[:|\alpha)$) the submatrix of A deleting the rows (or columns) indexed in the integer array α.

- First, deleting the zero rows indexed in α of A_1, which are revealed by its representation, and removing the corresponding columns of A_2, we get that

$$A = A_2' A_1' := A_2[:|\alpha) \cdot A_1(\alpha|:],$$

  where the representations of A_1' and A_2' are extracted from those of A_1 and A_2, respectively. Then, a nonsingular bidiagonal matrix L is constructed such that zero rows of $A_1^{(1)} = L A_1'$ are further revealed by its representation, reformulating A as

$$A = A_2'' A_1^{(1)} := (A_2' L^{-1})(L A_1'),$$

where the representations of $A_1^{(1)}$ and A_2'' are extracted from those of A_1' and A_2', respectively.

- Second, we perform an orthogonal transformation to deflate some zero singular values of A by deleting zero rows of A_2'' revealed by its representation, arriving at

$$A^{(1)} = A_2^{(1)} A_1^{(1)} := (G^T A_2'') A_1^{(1)},$$

  where G are orthogonal, and the representation of $A_2^{(1)}$ is extracted from that of A_2''.

- Further, via the resulting representation of $A_2^{(1)}$, we delete the zero columns indexed in β of $A_2^{(1)}$ and remove the corresponding rows of $A_1^{(1)}$ to get that

$$A^{(1)} = A_2'^{(1)} A_1'^{(1)} := A_2^{(1)}[: |\beta) \cdot A_1^{(1)}(\beta| :],$$

  where the representations of $A_1'^{(1)}$ and $A_2'^{(1)}$ are extracted from those of $A_1^{(1)}$ and $A_2^{(1)}$, respectively. Then, a nonsingular bidiagonal matrix U is constructed such that zero columns of $A_2^{(2)} = A_2'^{(1)} U$ are further revealed by its representation, reformulating $A^{(1)}$ as

$$A^{(1)} = A_2^{(2)} A_1''^{(1)} := (A_2'^{(1)} U)(U^{-1} A_1'^{(1)}),$$

  where the representations of $A_1''^{(1)}$ and $A_2^{(2)}$ are extracted from those of $A_1'^{(1)}$ and $A_2'^{(1)}$, respectively.

- Finally, we perform an orthogonal transformation to deflate some zero singular values of $A^{(1)}$ by deleting zero columns of $A_1''^{(1)}$ revealed by its representation, arriving at

$$A^{(2)} = A_2^{(2)} A_1^{(2)} := A_2^{(2)} (A_1''^{(1)} V),$$

  where V is orthogonal, and the representation of $A_1^{(2)}$ is extracted from that of $A_1''^{(1)}$.

We subsequently perform the detection and deflation operations described above on the representations of $A_2^{(2)}$ and $A_1^{(2)}$ to yeild a product of a nonsingular diagonal matrix and a bidiagonal matrix of full rank. Thereby, all the zero singular values are deflated. Subsequently, the nonzero singular values are computed by applying the available SVD algorithm [11, 15, 32] to the resulting full-rank bidiagonal matrix. The whole process involves no subtraction of same-signed numbers, ensuring that zero singular values are exactly deflated and the nonzero one are computed with high relative accuracy. Moreover, the overall computational cost is at most $O(rS + \max\{n_0^2 r, n_K^2 r\})$, which is preferable when $r \ll \min\{n_0, n_K\}$.

Remark that a periodic reduction method has been developed to reduce a full-rank product of structured matrices into a tridiagonal form, facilitating the accurate eigenvalue computation [20]. In addition, a reduction method has been provided to transform a full-rank product of an upper triangular matrix and a BF matrix into a bidiagonal form, tailored for the accurate generalized singular value computation [21]. Notably, our method here focuses on exactly deflating zero singular values of a rank-deficient bidiagonal product.

The rest of the paper is organized as follows. In Section 2, we illustrate the crucial role played by the bidiagonal product form when dealing with rank deficiency and submatrices. Section 3 introduces a new method to extract the representation of an arbitrary submatrix from the original one, which is indispensable when deflating zero singular values. In Section 4, we establish a method to exactly deflate zero singular values of a rank-deficient nonnegative bidiagonal product in a preferable complexity. The overall SVD algorithm is designed to accurately compute the SVD of such a product as well as that of its arbitrary submatrix. Numerical experiments are performed in Section 5 to confirm the claimed high relative accuracy.

Before proceeding, we present some notation and symbols. Let α and β be integer arrays. $A[\alpha|\beta]$ (or $A(\alpha|\beta)$) represents the submatrix of $A \in \mathbb{R}^{n\times m}$ having (or deleting) the rows and columns indexed in α and β, respectively, and it is abbreviated to $A[\alpha]$ (or $A(\alpha)$) if $\alpha = \beta$. The $n \times m$ identity matrix is denoted by $I_{n,m}$, and $I_{n,n}$ is abbreviated to I_n. In the sequel, we mean by

$$L_{f:s} = \mathbf{bilow}(\{\bar{x}_i, x_i\}_{i=f}^{s}) \in \mathbb{R}^{n\times n},\ U_{f:s} = \mathbf{biupp}(\{\bar{y}_i, y_i\}_{i=f}^{s}) \in \mathbb{R}^{n\times n}$$

the lower (upper) bidiagonal matrix obtained from I_n by replacing its (i,i)th and $(i+1,i)$th ((i,i)th and $(i,i+1)$th) entries by $\bar{x}_i$ and x_i ($\bar{y}_i$ and y_i) for all $f \leq i \leq s$, respectively; with the convention $\{x_n\} = \emptyset$ ($\{y_n\} = \emptyset$) if $s = n$. All $\{\bar{x}_i, x_i\}$ ($\{\bar{y}_i, y_i\}$) are called nontrivial *element pairs* of $L_{f:s}$ ($U_{f:s}$). Set $L_{f:s} = I_n$ ($U_{f:s} = I_n$) if $f > s$ or $f > n$. Observe that $L_{f:s} = L_{f:f}L_{f+1:f+1}\ldots L_{s:s}$ and $U_{f:s} = U_{s:s}\ldots U_{f+1:f+1}U_{f:f}$.

2 Bidiagonal Product Forms

In this section, we underscore the crucial role played by the bidiagonal product form when dealing with the SVD computations for rank deficiency and submatrices.

2.1 Computing SVDs of rank-deficient structured matrices

Many structured matrices with distinct nodes, such as Cauchy, Vandermonde, Cauchy-Vandermonde and Bernstein-Vandermonde matrices, are full-rank and have been shown to have the explicit bidiagonal decomposition [23, 27–30]:

$$A = L_{n-1}\ldots L_1 D U_1 \ldots U_{m-1} \in \mathbb{R}^{n\times m}, \tag{2}$$

where $D \in \mathbb{R}^{n\times m}$ is diagonal, $L_k \in \mathbb{R}^{n\times n}$ (or $U_k \in \mathbb{R}^{m\times m}$) is nonsingular lower (or upper) bidiagonal. Based on the bidiagonal decomposition, accurate SVD computations have been performed via the available algorithms [24]. However, the presence of repeated nodes in these structured matrices often leads to rank deficiency, presenting a significant obstacle to accurately computing their bidiagonal decompositions [25]. Recent efforts have addressed this issue by exploiting special structures of the factors L_k in the bidiagonal decomposition in the context of Vandermonde and (q-, h-) Bernstein-Vandermonde matrices [7]. However, a general approach for representing structured matrices with repeated nodes in a bidiagonal product form has remained elusive. In response to this gap, we propose a systematic methodology that allows for the representation of these matrices with repeated nodes in a more manageable bidiagonal product form.

In fact, these structured matrices containing repeated nodes can be effectively expressed as a product of bidiagonal matrices and structured matrices of the same type but with distinct nodes. This process is demonstrated as follows. Let the (i,j)-th entry of the structured matrix $A \in \mathbb{R}^{n\times m}$ be a function of x_i and y_j. The nodes x_i, $1 \le i \le n$ and y_j, $1 \le j \le m$, are arranged such that the same nodes appear consecutively. More specifically, we let

$$\begin{cases} x_{i_k} = x_{i_k+1} = \ldots = x_{i_{k+1}-1} \neq x_{i_{k+1}}, \ \forall\, 1 \le k \le n_1, \\ y_{j_k} = y_{j_k+1} = \ldots = y_{j_{k+1}-1} \neq y_{j_{k+1}}, \ \forall\, 1 \le k \le m_1, \end{cases}$$

with the conventions $i_1 = j_1 = 1$, $i_{n_1+1} = n+1$ and $j_{m_1+1} = m+1$. Then the corresponding rows (and columns) of A are repeated, i.e.,

$$\begin{cases} A[i_k|\,:] = A[i_k+1|\,:] = \ldots = A[i_{k+1}-1|\,:] \neq A[i_{k+1}|\,:], \ \forall\, 1 \le k \le n_1, \\ A[:\,|j_k] = A[:\,|j_k+1] = \ldots = A[:\,|j_{k+1}-1] \neq A[:\,|j_{k+1}], \ \forall\, 1 \le k \le m_1. \end{cases}$$

A key observation is that A can be obtained by expanding its submatrix

$$A[i_1,\ldots,i_{n_1}|j_1,\ldots,j_{m_1}],$$

which is the structured matrix of the same type associated with distinct nodes, via multiplication by rectangular identity matrices and the bidiagonal matrices $E_i(x,y)$. Here, $E_i(x,y)$ is the matrix obtained from the identity I_n by replacing its (i,i) and $(i+1,i)$ entries with x and y, respectively. We have

$$\begin{aligned} A = &\left(\prod_{k=1}^{n_1}\prod_{i_{k+1}-2}^{t=i_k} E_t(1,1)\right)\left(\prod_{k=2}^{n_1}\prod_{i_k-1}^{t=k} E_t(0,1)\right) \\ &I_{n,n_1}\cdot A[i_1,\ldots,i_{n_1}|j_1,\ldots,j_{m_1}]\cdot I_{m_1,m} \\ &\left(\prod_{k=2}^{m_1}\prod_{j_k-1}^{t=k} E_t(0,1)\right)^T \left(\prod_{k=1}^{m_1}\prod_{j_{k+1}-2}^{t=j_k} E_t(1,1)\right)^T. \end{aligned}$$

As long as the full-rank bidiagonal decomposition

$$A[i_1,\ldots,i_{n_1}|j_1,\ldots,j_{m_1}] = L_{n_1-1}\ldots L_1 D U_1 \ldots U_{m_1-1}$$

is available, A can be expressed in the *bidiagonal product* form

$$A = \left(\prod_{k=1}^{n_1}\prod_{i_{k+1}-2}^{t=i_k} E_t(1,1)\right)\left(\prod_{k=2}^{n_1}\prod_{i_k-1}^{t=k} E_t(0,1)\right) \tag{3}$$

$$I_{n,n_1} \cdot L_{n_1-1} \dots L_1 D U_1 \dots U_{m_1-1} \cdot I_{m_1,m}$$
$$\left(\prod_{k=2}^{m_1} \prod_{j_k-1}^{t=k} E_t(0,1)\right)^T \left(\prod_{k=1}^{m_1} \prod_{j_{k+1}-2}^{t=j_k} E_t(1,1)\right)^T.$$

This approach offers a unified framework for accurately computing the SVDs involving structured matrices with repeated nodes, as demonstrated in Section 5.

2.2 Computing SVDs of submatrices

The SVD computation of submatrices has garnered substantial interest from various applications [18]. We interestingly find out that an arbitrary submatrix can be expressed as a product of rectangular bidiagonal matrices and the original matrix. Let us concretely demonstrate this fact as follows.

Consider the submatrices $A(r|:]$ and $A[:|r)$ of $A \in \mathbb{R}^{n\times m}$. Denote

$$\mathcal{U}_{r:n-1} = \mathbf{biupp}(\{0,1\}_{i=r}^{n-1}) \in \mathbb{R}^{(n-1)\times n}, \quad \mathcal{L}_{r:m-1} = \mathbf{bilow}(\{0,1\}_{i=r}^{m-1}) \in \mathbb{R}^{m\times(m-1)}.$$

Then

$$A(r|:] = \mathcal{U}_{r:n-1}A, \quad A[:|r) = A\mathcal{L}_{r:m-1}.$$

Thereby, for an arbitrary submatrix $A(\alpha|\beta)$ of A, it can be expressed as

$$A(\alpha|\beta) = \prod_{i=1} \mathcal{U}_{\alpha_i:n-k+i-1} \cdot A \cdot \prod_{l}^{i=1} \mathcal{L}_{\beta_i:m-k+i-1}.$$

Here, $\alpha = (\alpha_i) \in Q_{k,n}$ and $\beta = (\beta_i) \in Q_{l,m}$, and $Q_{k,n}$ denotes the set of the vectors of length k consisting of distinctive integers from $\{1, \dots, n\}$. In particular, if the original matrix A has a bidiagonal decomposition, e.g., the form (2) is available, then the submatrix $A(\alpha|\beta)$ is of the following bidiagonal product form

$$A(\alpha|\beta) = \prod_{i=1} \mathcal{U}_{\alpha_i:n-k+i-1} \cdot \prod_{n-1}^{i=1} L_i \cdot D \cdot \prod_{i=1}^{m-1} U_i \cdot \prod_{l}^{i=1} \mathcal{L}_{\beta_i:m-k+i-1},$$

As shown in Section 5, accurate SVD computations are achieved for arbitrary submatrices of many structured matrices as well as their products, by virtue of depicting submatrices as bidiagonal product forms.

Therefore, our primary focus in this paper is to conduct the SVD computation for a bidiagonal product of arbitrary rank. Our proposed method aims to compute with high relative accuracy the SVD of this product.

3 The extracting method for arbitrary submatrices

In this section, we present a method for extracting the representation of an arbitrary submatrix from the original one in the context of a nonnegative bidiagonal product, specifically designed to handle rank deficiency by deflating zero singular values later.

Before proceeding, we introduce some notations that are frequently utilized later. By extending the representation of a square TN matrix in [25] to a rectangular bidiagonal product, throughout the paper we always mean by

$$A =: (\{\bar{g}_{ij}, g_{ij}\}) \in \mathbb{R}^{n\times m} \tag{4}$$

the matrix generated from the $n \times m$ element pairs $\{\bar{g}_{ij}, g_{ij}\}_{i,j=1}^{n,m}$, which are stored as the matrix $\mathcal{BD}(A) = (\{\bar{g}_{ij}, g_{ij}\}) \in \mathbb{R}^{n\times m}$, by the following bidiagonal product form

$$A = L_{n-1} \dots L_1 D U_1 \dots U_{m-1}, \tag{5}$$

where $D = \mathbf{diag}(\{g_{ii}\}_{i=1}^{\min\{n,m\}}) \in \mathbb{R}^{n\times m}$, and

$$\begin{cases} L_k = \mathbf{bilow}(\{\bar{g}_{i+1,i+1-k}, g_{i+1,i+1-k}\}_{i=k}^{\min\{n-1,m+k-1\}}) \in \mathbb{R}^{n\times n}, \ \forall\ 1 \le k \le n-1, \\ U_l = \mathbf{biupp}(\{\bar{g}_{i+1-l,i+1}, g_{i+1-l,i+1}\}_{i=l}^{\min\{m-1,n+l-1\}}) \in \mathbb{R}^{m\times m}, \ \forall\ 1 \le l \le m-1. \end{cases}$$

We call $\mathcal{BD}(A)$ to be the representation of A. Notice that all $\bar{g}_{ii}$ has no impact on forming A. In addition, by the form (5), without loss of generality, we always assume that $g_{ii} \ne 0$ for all $i \ne \min\{n, m\}$. If all $\bar{g}_{ij} \ge 0$ and $g_{ij} \ge 0$, we denote $\mathcal{BD}(A) \ge 0$. Clearly, the representation of A^T is the transpose of $\mathcal{BD}(A)$, i.e.,

$$A^T =: (\{\bar{g}_{ji}, g_{ji}\}) \in \mathbb{R}^{m\times n}. \tag{6}$$

Later, we will show that any nonnegative bidiagonal product has the representation.

Given the representation $\mathcal{BD}(A) = (\{\bar{g}_{ij}, g_{ij}\}) \in \mathbb{R}^{n\times m}$, we provide the updating and downdating procedures when appending and deleting rows (columns) of A.

- Suppose $A' \in \mathbb{R}^{t\times m}$ $(t > n)$ is obtained from $A \in \mathbb{R}^{n\times m}$ by attaching $(t - n)$ zero rows to the last rows of A. Then

$$A' = I_{t,n}A = \begin{bmatrix} A \\ 0_{(t-n)\times m} \end{bmatrix} = \begin{bmatrix} L_{n-1} & \\ & I_{t-n} \end{bmatrix} \cdots \begin{bmatrix} L_1 & \\ & I_{t-n} \end{bmatrix} \begin{bmatrix} D \\ 0_{(t-n)\times m} \end{bmatrix} U_1 \dots U_{m-1}.$$

  Thus, $\mathcal{BD}(A') = (\{\bar{g}'_{ij}, g'_{ij}\}) \in \mathbb{R}^{t\times m}$ is obtained as

$$\{\bar{g}'_{ij}, g'_{ij}\} = \begin{cases} (1, 0), \ \forall\ n+1 \le i \le t,\ 1 \le j \le m, \\ \{\bar{g}_{ij}, g_{ij}\}, \ \text{otherwise}. \end{cases} \tag{7}$$

- Suppose $A' \in \mathbb{R}^{t\times m}$ $(t < n)$ is obtained from $A \in \mathbb{R}^{n\times m}$ by deleting the last $(n - t)$ rows. Then

$$A' = I_{t,n}A = A[1:t|:] = L_{n-1}[1:t] \dots L_1[1:t] D[1:t|:] U_1 \dots U_{m-1}. \tag{8}$$

Thus, $\mathcal{BD}(A') = (\{\bar{g}'_{ij}, g'_{ij}\}) \in \mathbb{R}^{t\times m}$ is obtained as

$$\{\bar{g}'_{ij}, g'_{ij}\} = \begin{cases} \{\bar{g}_{tj}, g_{tj} \prod_{k=1}^{j} \bar{g}_{t+1,k}\}, \ \forall\ i = t,\ 1 \le j \le \min\{t, m\}, \\ \{\bar{g}_{ij}, g_{ij}\}, \ \text{otherwise}, \end{cases} \tag{9}$$

costing $2\cdot\min\{t,m\}$ subtraction-free operations. We illustrate the downdating procedure (9) by treating the cases $t > m$ and $t \le m$ separately with the aid of the example $\mathcal{BD}(A) = (\{\bar{g}_{ij}, g_{ij}\}) \in \mathbb{R}^{7\times 4}$ and $t = 5, 2$ as follows.

– For $t = 5 > m = 4$, we have by (8) that

$$\begin{aligned} A' =& \begin{bmatrix} 1 &&&& \\ & 1 &&& \\ && 1 && \\ &&& 1 & \\ &&&& \bar{g}_{61} \end{bmatrix} \begin{bmatrix} 1 &&&& \\ & 1 &&& \\ && 1 && \\ &&& \bar{g}_{51} & \\ &&& g_{51} & \bar{g}_{62} \end{bmatrix} \begin{bmatrix} 1 &&&& \\ & 1 &&& \\ && \bar{g}_{41} && \\ && g_{41} & \bar{g}_{52} & \\ &&& g_{52} & \bar{g}_{63} \end{bmatrix} \begin{bmatrix} 1 &&&& \\ & \bar{g}_{31} &&& \\ & g_{31} & \bar{g}_{42} && \\ && g_{42} & \bar{g}_{53} & \\ &&& g_{53} & \bar{g}_{64} \end{bmatrix} \begin{bmatrix} \bar{g}_{21} &&&& \\ g_{21} & \bar{g}_{32} &&& \\ & g_{32} & \bar{g}_{43} && \\ && g_{43} & \bar{g}_{54} & \\ &&& g_{54} & \bar{g}_{65} \end{bmatrix} \\ & \begin{bmatrix} g_{11} & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 \\ 0 & 0 & g_{33} & 0 \\ 0 & 0 & 0 & g_{44} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{g}_{12} & g_{12} && \\ & \bar{g}_{23} & g_{23} & \\ && \bar{g}_{34} & g_{34} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & \bar{g}_{13} & g_{13} & \\ && \bar{g}_{24} & g_{24} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & 1 && \\ && \bar{g}_{14} & g_{14} \\ &&& 1 \end{bmatrix} \\ =& \begin{bmatrix} 1 &&&& \\ & 1 &&& \\ && 1 && \\ &&& \bar{g}_{51} & \\ &&& g_{51}\bar{g}_{61} & 1 \end{bmatrix} \begin{bmatrix} 1 &&&& \\ & 1 &&& \\ && \bar{g}_{41} && \\ && g_{41} & \bar{g}_{52} & \\ &&& g_{52} \prod\limits_{k=1}^{2} \bar{g}_{6k} & 1 \end{bmatrix} \begin{bmatrix} 1 &&&& \\ & \bar{g}_{31} &&& \\ & g_{31} & \bar{g}_{42} && \\ && g_{42} & \bar{g}_{53} & \\ &&& g_{53} \prod\limits_{k=1}^{3} \bar{g}_{6k} & 1 \end{bmatrix} \begin{bmatrix} \bar{g}_{21} &&&& \\ g_{21} & \bar{g}_{32} &&& \\ & g_{32} & \bar{g}_{43} && \\ && g_{43} & \bar{g}_{54} & \\ &&& g_{54} \prod\limits_{k=1}^{4} \bar{g}_{6k} & 1 \end{bmatrix} \\ & \begin{bmatrix} g_{11} & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 \\ 0 & 0 & g_{33} & 0 \\ 0 & 0 & 0 & g_{44} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{g}_{12} & g_{12} && \\ & \bar{g}_{23} & g_{23} & \\ && \bar{g}_{34} & g_{34} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & \bar{g}_{13} & g_{13} & \\ && \bar{g}_{24} & g_{24} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & 1 && \\ && \bar{g}_{14} & g_{14} \\ &&& 1 \end{bmatrix} \\ =:& \begin{bmatrix} \{\bar{g}_{11}, g_{11}\} & \{\bar{g}_{12}, g_{12}\} & \{\bar{g}_{13}, g_{13}\} & \{\bar{g}_{14}, g_{14}\} \\ \{\bar{g}_{21}, g_{21}\} & \{\bar{g}_{22}, g_{22}\} & \{\bar{g}_{23}, g_{23}\} & \{\bar{g}_{24}, g_{24}\} \\ \{\bar{g}_{31}, g_{31}\} & \{\bar{g}_{32}, g_{32}\} & \{\bar{g}_{33}, g_{33}\} & \{\bar{g}_{34}, g_{34}\} \\ \{\bar{g}_{41}, g_{41}\} & \{\bar{g}_{42}, g_{42}\} & \{\bar{g}_{43}, g_{43}\} & \{\bar{g}_{44}, g_{44}\} \\ \{\bar{g}_{51}, g_{51}\bar{g}_{61}\} & \{\bar{g}_{52}, g_{52} \prod\limits_{k=1}^{2} \bar{g}_{6k}\} & \{\bar{g}_{53}, g_{53} \prod\limits_{k=1}^{3} \bar{g}_{6k}\} & \{\bar{g}_{54}, g_{54} \prod\limits_{k=1}^{4} \bar{g}_{6k}\} \end{bmatrix}. \end{aligned}$$

– For $t = 2 < m = 4$, we have by (8) that

$$\begin{aligned} A' =& \begin{bmatrix} 1 & \\ & \bar{g}_{31} \end{bmatrix} \begin{bmatrix} \bar{g}_{21} & \\ g_{21} & \bar{g}_{32} \end{bmatrix} \begin{bmatrix} g_{11} & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{g}_{12} & g_{12} && \\ & \bar{g}_{23} & g_{23} & \\ && \bar{g}_{34} & g_{34} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & \bar{g}_{13} & g_{13} & \\ && \bar{g}_{24} & g_{24} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & 1 && \\ && \bar{g}_{14} & g_{14} \\ &&& 1 \end{bmatrix} \\ =& \begin{bmatrix} \bar{g}_{21} & \\ g_{21}\bar{g}_{31} & 1 \end{bmatrix} \begin{bmatrix} g_{11} & 0 & 0 & 0 \\ 0 & g_{22} \prod\limits_{r=1}^{2} \bar{g}_{3r} & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{g}_{12} & g_{12} && \\ & \bar{g}_{23} & g_{23} & \\ && 1 & 0 \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & \bar{g}_{13} & g_{13} & \\ && \bar{g}_{24} & g_{24} \\ &&& 1 \end{bmatrix} \begin{bmatrix} 1 &&& \\ & 1 && \\ && \bar{g}_{14} & g_{14} \\ &&& 1 \end{bmatrix} \\ =:& \begin{bmatrix} \{\bar{g}_{11}, g_{11}\} & \{\bar{g}_{12}, g_{12}\} & \{\bar{g}_{13}, g_{13}\} & \{\bar{g}_{14}, g_{14}\} \\ \{\bar{g}_{21}, g_{21}\bar{g}_{31}\} & \{\bar{g}_{22}, g_{22} \prod\limits_{k=1}^{2} \bar{g}_{3k}\} & \{\bar{g}_{23}, g_{23}\} & \{\bar{g}_{24}, g_{24}\} \end{bmatrix}. \end{aligned}$$

For the case of appending (or deleting) the columns of $A \in \mathbb{R}^{n\times m}$, i.e., $A' = AI_{m\times t}$ for $t > m$ and $t < m$, the representation of A' is obtained by applying (7) (or (9)) to $I_{t\times m}A^T$ with the transposing operation (6).

In addition, the command `STNAddToPrevious` (`STNAddToNext`) has been provided in [25] to compute the representation of $AE_i(x, y)$ ($E_i(x, y)A$), where A is a square TN matrix and $E_i(x, y)$ is an elementary matrix. Now, given a rectangular representation $\mathcal{BD}(A) \in \mathbb{R}^{n\times m}$ and an upper (lower) bidiagonal matrix U (L), we extend the commands to compute the representation of UA (LA) by using the following passing-through operations, whose proofs are given in the appendix section.

- Given

$$U_{1:n} = \mathbf{biupp}(\{\bar{y}_i, y_i\}_{i=1}^n) \geq 0,\ L_{1:n} = \mathbf{bilow}(\{\bar{x}_i, x_i\}_{i=1}^n) \geq 0,$$

then

$$U_{1:n}L_{1:n} = \bar{L}_{1:n}\bar{U}_{1:n}, \tag{10}$$

where

$$\bar{U}_{1:n} = \mathbf{biupp}(\{\bar{y}'_i, y'_i\}_{i=1}^n),\ \bar{L}_{1:n} = \mathbf{bilow}(\{\bar{x}'_i, x'_i\}_{i=1}^n)$$

satisfies that for $1 \leq i \leq n$,

$$\begin{cases} z_1 = \bar{y}_1\bar{x}_1,\ w_i = z_i + x_i y_i, \\ \begin{cases} \bar{x}'_i = 1,\ \bar{y}'_i = w_i,\ x'_i = \frac{\bar{y}_{i+1}x_i}{\bar{y}'_i},\ y'_i = \frac{y_i\bar{x}_{i+1}}{\bar{x}'_i},\ z_{i+1} = \frac{\bar{y}_{i+1}\bar{x}_{i+1}z_i}{\bar{x}'_i\bar{y}'_i},\ \text{if } w_i \neq 0, \\ \bar{x}'_i = 0,\ \bar{y}'_i = 1,\ x'_i = \bar{y}_{i+1}x_i,\ y'_i = 0,\ z_{i+1} = \bar{y}_{i+1}\bar{x}_{i+1},\ \text{if } w_i = 0 \text{ and } y_i = 0, \\ \bar{x}'_i = 1,\ \bar{y}'_i = 0,\ x'_i = 0,\ y'_i = y_i\bar{x}_{i+1},\ z_{i+1} = \bar{y}_{i+1}\bar{x}_{i+1},\ \text{otherwise}, \end{cases} \end{cases}$$

which costs at most $8n$ subtraction-free arithmetic operations.

- Given

$$D = \operatorname{diag}(d_i) \in \mathbb{R}^{n\times m},\ U_{1:n} = \mathbf{biupp}(\{\bar{y}_i, y_i\}_{i=1}^n),$$

then

$$U_{1:n}D = \bar{D}\bar{U}_{1:m},\ \text{where } \begin{cases} \bar{D} = \mathbf{diag}(\bar{d}_i) \in \mathbb{R}^{n\times m}, \\ \bar{U}_{1:m} = \mathbf{biupp}(\{\bar{y}'_i, y'_i\}_{i=1}^m) \in \mathbb{R}^{m\times m}, \end{cases} \tag{11}$$

here, by a straightforward calculation,

$$\begin{cases} \begin{cases} \bar{d}_i = d_i\bar{y}_i,\ \bar{y}'_i = 1,\ y'_i = \frac{d_{i+1}y_i}{d_i},\ \text{if } d_i\bar{y}_i \neq 0, \\ \bar{d}_i = 1,\ \bar{y}'_i = 0,\ y'_i = d_{i+1}y_i,\ \text{if } d_i\bar{y}_i = 0, \end{cases} & \forall\ 1 \leq i \leq \min\{n, m\}, \\ \bar{y}'_i = 1,\ y'_i = 0,\ \forall\ \min\{n, m\} < i \leq m, \end{cases}$$

which costs at most $3\min\{n, m\}$ subtraction-free arithmetic operations.

- Given

$$U_{1:m} = \mathbf{biupp}(\{\bar{y}_i, y_i\}_{i=1}^m) \geq 0,\ U'_{1:m} = \mathbf{biupp}(\{\bar{x}_i, x_i\}_{i=1}^m) \geq 0,$$

then

$$U_{1:m}U'_{1:m} = \bar{U}'_{1:m}\bar{U}_{2:m}, \tag{12}$$

where

$$\bar{U}'_{1:m} = \mathbf{biupp}(\{\bar{x}'_i, x'_i\}_{i=1}^m),\ \bar{U}_{2:m} = \mathbf{biupp}(\{\bar{y}'_i, y'_i\}_{i=2}^m),$$

satisfies that for all $1 \le i \le m-1$,

$$\begin{cases} \bar{x}'_1 = \bar{x}_1\bar{y}_1,\ z_1 = \bar{y}_1 x_1,\ w_i = z_i + \bar{x}_{i+1}y_i, \\ \begin{cases} \bar{y}'_{i+1} = 1,\ x'_i = w_i,\ \bar{x}'_{i+1} = \frac{\bar{x}_{i+1}\bar{y}_{i+1}}{\bar{y}'_{i+1}},\ y'_{i+1} = \frac{x_{i+1}y_i}{x'_i},\ z_{i+1} = \frac{x_{i+1}\bar{y}_{i+1}z_i}{\bar{y}'_{i+1}x'_i},\ \text{if } w_i \neq 0, \\ \bar{y}'_{i+1} = 0,\ x'_i = 1,\ \bar{x}'_{i+1} = 0,\ y'_{i+1} = x_{i+1}y_i,\ z_{i+1} = x_{i+1}\bar{y}_{i+1},\ \text{if } w_i = 0 \text{ and } y_i \neq 0, \\ \bar{y}'_{i+1} = 1,\ x'_i = 0,\ \bar{x}'_{i+1} = \bar{x}_{i+1}\bar{y}_{i+1},\ y'_{i+1} = 0,\ z_{i+1} = x_{i+1}\bar{y}_{i+1},\ \text{otherwise}, \end{cases} \end{cases}$$

which costs at most $8m$ subtraction-free arithmetic operations.

Equipping with the preliminaries above, we are ready to extract the representation of an arbitrary submatrix $A(\alpha|\beta)$ from the original $\mathcal{BD}(A)$. We first consider the submatrix $A(r|:]$ ($A[:|r)$) by deleting the rth row (column) of $A \in \mathbb{R}^{n\times m}$. Denote

$$\mathcal{U}_r = \mathbf{biupp}(\{0,1\}_{i=r}^{n}) \in \mathbb{R}^{n\times n},\quad \mathcal{L}_r = \mathbf{bilow}(\{0,1\}_{i=r}^{m}) \in \mathbb{R}^{m\times m}. \tag{13}$$

Then it is interesting to observe that

$$\mathcal{U}_r A = \begin{bmatrix} A(r|:] \\ 0 \end{bmatrix},\quad A\mathcal{L}_r = \begin{bmatrix} A[:|r) & 0 \end{bmatrix}.$$

Further, deleting the last row (column) of $\mathcal{U}_r A$ ($A\mathcal{L}_r$) is equivalent to left-multiplying $I_{n-1,n}$ (right-multiplying $I_{m,m-1}$), i.e.,

$$A(r|:] = I_{n-1,n}\mathcal{U}_r A,\quad A[:|r) = A\mathcal{L}_r I_{m,m-1}.$$

Thus, when deleting the rth row (or column) of A, we compute the representation of the resulting submatrix $A(r|:]$ (or $A[:|r)$) as follows:

- first, compute the representation of $\bar{A} = \mathcal{U}_r A$ (or $\bar{A} = A\mathcal{L}_r$) by using (10), (11) and (12), costing at most $O((n-r)m)$ (or $O(n(m-r))$) subtraction-free operations;
- then, compute the representation of $A' = I_{n-1,n}\bar{A}$ (or $A' = \bar{A}I_{m,m-1}$) by using (9), costing at most $O(m)$ (or $O(n)$) subtraction-free operations.

Now, given any submatrix $A(\alpha|\beta)$ of $A \in \mathbb{R}^{n\times m}$ with $\alpha = (\alpha_i) \in Q_{t,n}$ and $\beta = (\beta_i) \in Q_{l,m}$, using the argument above, it can be depicted as

$$A(\alpha|\beta) = \prod_{i=1}^{t} I_{n-t+i-1,n-t+i}\mathcal{U}_{\alpha_i} \cdot A \cdot \prod_{l}^{i=1} \mathcal{L}_{\beta_i} I_{m-l+i,m-l+i-1}.$$

Therefore, we summarize Algorithm 1 to compute the representation of any submatrix $A(\alpha|\beta)$ in $t+l$ extracting steps, each step corresponding to the deletion of one row indexed in α or one column indexed in β. The cost of the whole procedure is at most $O((t+l)nm)$. By now, the issue of computing representations for arbitrary submatrices in general TN matrices has been addressed affirmatively, and we enhance the existing capabilities and broaden the scope of extracting submatrix representations in nonnegative bidiagonal products, including general TN matrices.

Algorithm 1 Given the representation $\mathcal{BD}(A)$ of $A \in \mathbb{R}^{n\times m}$, this algorithm computes the representation of an arbitrary submatrix $A(\alpha|\beta)$.

Input: the representation $\mathcal{BD}(A)$, the index sets $\alpha = (\alpha_i) \in Q_{t,n}$ and $\beta = (\beta_i) \in Q_{l,m}$.
Output: the representation $\mathcal{BD}(A(\alpha|\beta))$.

```
for i = t : −1 : 1 do
    Set A := U_{α_i} A, where U_{α_i} is as in (13).
    Compute BD(A) by the procedures (10), (11) and (12).
    Compute BD(A) of A := I_{n−1,n} A by the downdating procedure (9).
    Set n := n − 1.
end for
for i = l : −1 : 1 do
    Set A := A L_{β_i}, where L_{β_i} is as in (13).
    Compute BD(A) by the procedures (10), (11) and (12).
    Compute BD(A) of A := A I_{m,m−1} by the downdating procedure (9).
    Set m := m − 1.
end for
```

Finally, we conclude this section with how to compute the representation $\mathcal{BD}(A)$ of a product $A = B_1B_2\dots B_K \in \mathbb{R}^{n_0\times n_K}$ as in (1). Suppose each $B_i \in \mathbb{R}^{n_{i-1}\times n_i}$ $(1 \le i \le K)$ has s_i nontrivial element pairs. Then the total number of nontrivial element pairs is $S = \sum_{i=1}^{K} s_i$. We consider the case $n_0 \ge n_K$ (the case $n_0 < n_K$ can be treated alike). Observe that the lower (upper) bidiagonal matrix B_i can be rewritten as

$$B_i = B_i' I_{n_{i-1}\times n_i} \quad (B_i = I_{n_{i-1}\times n_i} B_i'), \tag{14}$$

where B_i' is square lower (or upper) bidiagonal having the same nontrivial element pairs as those of B_i. Set $A_K = I_{n_K}$. Then we derive the representation of A by sequentially computing the representation of $A_{i-1} = B_iA_i$ $(i = K,\dots,1)$ as follows:

- if B_i is lower bidiagonal, then by (14), we first perform $A_i := I_{n_{i-1}\times n_i}A_i$ by (7) or (9), then, compute $A_{i-1} := B_i'A_i$ by the procedure (12);
- if B_i is upper bidiagonal, then by (14), we first compute $A_{i-1} := B_i'A_i$ by the procedures (10), (11) and (12), then, perform $A_{i-1} := I_{n_{i-1}\times n_i}A_{i-1}$ by (7) or (9);

which costs $O(s_in_K)$ operations. Hence, the total cost is $O(\sum_{i=1}^{K} s_in_K) = O(Sn_K)$.

4 The deflation method for zero singular values

In this section, we develop a method to exactly deflate all the zero singular values for a nonnegative bidiagonal product of arbitrary rank.

It is important to note that deflating zero singular values exactly may not always be possible due to numerical precision limitations. The idea behind our exact deflation goes as follows: given $\mathcal{BD}(A) = (\{\bar{g}_{ij}, g_{ij}\}) \in \mathbb{R}^{n\times m}$,

- if $\bar{g}_{k1} = 0$ for some $2 \le k \le n$, then $A[k-1|:] = 0$, which means that zero rows of A are implicitly revealed by zero elements in the first column of $\mathcal{BD}(A)$. Thus, by the position of the zero element, we have a permutation matrix

$$P = \mathbf{bilow}(\{0,1\}_{i=k-1}^{n}) + e_{k-1}e_n^T \in \mathbb{R}^{n\times n}, \tag{15}$$

such that

$$P^T A = \begin{bmatrix} A(k-1|\,:] \\ 0 \end{bmatrix}.$$

So, we can deflate a possible zero singular value by deleting the zero row. It then remains to compute the submatrix representation $\mathcal{BD}(A(k-1|\,:])$ by Algorithm 1, on which we go on to implicitly deflate zero singular values. Similar treatment applies to the case where $\bar{g}_{1k} = 0$ for some $2 \le k \le m$, which implies $A[:\,|k-1] = 0$;

- if all $\bar{g}_{i1} \neq 0$, i.e., $\bar{g}_{i1} = 1$ for all $2 \le i \le n$, then no zero row of A is revealed. In this case, denote $L = \mathbf{bilow}(\{1, -g_{i1}\}_{i=2}^{n})$, and set $A' = LA$, then $\mathcal{BD}(A')$ is obtained simply by setting the elements $g_{i1} = 0$ $(2 \le i \le n)$ in $\mathcal{BD}(A)$. Now, zero rows of A' are revealed again by zero elements $\bar{g}_{k2} = 0$ in the second column of $\mathcal{BD}(A')$. Similar treatment applies to the case that $\bar{g}_{1j} \neq 0$ for all $2 \le j \le m$.

The other technique is to express the inverse of a lower bidiagonal matrix as the product of an orthogonal matrix and the inverse of an upper bidiagonal matrix. More detailed, given $L_k = \mathbf{bilow}(\{\bar{x}_i, -x_i\}_{i=k}^{n}) \in \mathbb{R}^{n\times n}$ $(1 \le k \le n)$, it is easy to find Givens rotations $G_i \in \mathbb{R}^{n\times n}$ $(k+1 \le i \le n)$ such that

$$L_k G_n G_{n-1} \dots G_{k+1} = U_k = \mathbf{biupp}(\{\bar{y}_i, -y_i\}_{i=k}^{n}) \in \mathbb{R}^{n\times n},$$

where

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

and

$$\begin{cases} z_n = \bar{x}_n,\ \bar{y}_i = \sqrt{z_i^2 + x_{i-1}^2},\ c_i = z_i/\bar{y}_i,\ s_i = x_{i-1}/\bar{y}_i, \\ y_{i-1} = s_i\bar{x}_{i-1},\ z_{i-1} = c_i\bar{x}_{i-1},\ \bar{y}_k = z_k, \end{cases} \quad i = n, n-1, \dots, k+1,$$

which costs at most $8(n-k)$ subtraction-free operations. If all $\bar{x}_i$ and x_i are nonnegative, then so are all $c_i, s_i, \bar{y}_i, y_i$. Set $G = G_n \dots G_{k+1}$. Then G is orthogonal. Moreover, if L_k is nonsingular, then L_k^{-1} is expressed as

$$L_k^{-1} = G \cdot U_k^{-1}, \quad \text{where } U_k^{-1} = \prod_{i=k}^{n} U_{i:i} \text{ with } U_{i:i} = \mathbf{biupp}(\{1/\bar{y}_i, y_i/\bar{y}_i\}). \tag{16}$$

In what follows, we are to illustrate the deflation of zero singular values for a nonnegative bidiagonal product $A = \prod_{i=1}^{K} B_i \in \mathbb{R}^{n_0 \times n_K}$ as in (1) having S nontrivial element pairs in the factors $B_i \in \mathbb{R}^{n_{i-1}\times n_i}$ $(1 \le i \le K)$. Set the minimun dimension $r = \min_{0\le i\le K}\{n_i\}$. In Section 4.1, we provide a detailed explanation of the deflation method for the case $r = \min\{n_0, n_K\}$, and we then extend it to a general case in Section 4.2. Finally, the overall SVD algorithm is established in Section 4.3, and error analysis is conducted to illustrate the high relative accuracy of our proposed method.

4.1 The deflation for the case $r = \min\{n_0, n_K\}$

This section is dealt with deflating zero singular values for the case $r = \min\{n_0, n_K\}$.

Basing on the minimum dimension, we compute the representation $\mathcal{BD}(A) = (\{\bar{g}_{ij}, g_{ij}\})$ by using the operations (7), (9), (10), (11) and (12), which costs at most $O(Sr)$ subtraction-free operations.

Focusing on $\mathcal{BD}(A) = (\{\bar{g}_{ij}, g_{ij}\}) \in \mathbb{R}^{n_0 \times n_K}$, we next illustrate how to carry out orthogonal transformations to deflate zero singular values of A, which is explained by the aid of an example with $n_0 = 5$ and $n_K = 4$.

As a first step, we are to deflate zero singular values of A revealed by the zero elements $\bar{g}_{i1} = 0$ in the first column of $\mathcal{BD}(A)$. Assume that

$$\bar{g}_{i_1,1} = \bar{g}_{i_2,1} = \ldots = \bar{g}_{i_h,1} = 0, \quad i_1 < i_2 < \cdots < i_h.$$

Set $\alpha = (i_1 - 1, i_2 - 1, \ldots, i_h - 1)$. Then one can detect $A[\alpha|\,:] = 0$. Thus, there have permutation matrices P_{i_j} $(1 \le j \le h)$ as in (15) such that

$$P_{i_1}^T P_{i_2}^T \ldots P_{i_h}^T A = \begin{bmatrix} A(\alpha|\,:] \\ 0 \end{bmatrix}.$$

So, we orthogonally deflate possible h zero singular values by deleting the h zero rows, and the substantial operation is to compute $\mathcal{BD}(A(\alpha|\,:])$ by Algorithm 1, which costs at most $O(hn_0n_K)$ subtraction-free operations. Moreover, in this way we can orthogonally deflate the associated zero singular values to arrive at the case where $\bar{g}_{i1} = 1$ for all i. For the simplicity of our description about the next operation, in our example $A \in \mathbb{R}^{5\times 4}$, assume that

$$A =: \begin{bmatrix} \{\bar{g}_{11},g_{11}\} & \{\bar{g}_{12},g_{12}\} & \{\bar{g}_{13},g_{13}\} & \{\bar{g}_{14},g_{14}\} \\ \{1,g_{21}\} & \{\bar{g}_{22},g_{22}\} & \{\bar{g}_{23},g_{23}\} & \{\bar{g}_{24},g_{24}\} \\ \{1,g_{31}\} & \{\bar{g}_{32},g_{32}\} & \{\bar{g}_{33},g_{33}\} & \{\bar{g}_{34},g_{34}\} \\ \{1,g_{41}\} & \{\bar{g}_{42},g_{42}\} & \{\bar{g}_{43},g_{43}\} & \{\bar{g}_{44},g_{44}\} \\ \{1,g_{51}\} & \{\bar{g}_{52},g_{52}\} & \{\bar{g}_{53},g_{53}\} & \{\bar{g}_{54},g_{54}\} \end{bmatrix}.$$

Further, we are to perform an orthogonal transformation to zero the elements g_{i1}. Set $X = \mathbf{bilow}(\{1, -g_{i+1,1}\}_{i=1}^{n_0-1}) \in \mathbb{R}^{n_0 \times n_0}$. Denote $A' = XA$. Then

$$A' =: \begin{bmatrix} \{\bar{g}_{11},g_{11}\} & \{\bar{g}_{12},g_{12}\} & \{\bar{g}_{13},g_{13}\} & \{\bar{g}_{14},g_{14}\} \\ \{1,0\} & \{\bar{g}_{22},g_{22}\} & \{\bar{g}_{23},g_{23}\} & \{\bar{g}_{24},g_{24}\} \\ \{1,0\} & \{\bar{g}_{32},g_{32}\} & \{\bar{g}_{33},g_{33}\} & \{\bar{g}_{34},g_{34}\} \\ \{1,0\} & \{\bar{g}_{42},g_{42}\} & \{\bar{g}_{43},g_{43}\} & \{\bar{g}_{44},g_{44}\} \\ \{1,0\} & \{\bar{g}_{52},g_{52}\} & \{\bar{g}_{53},g_{53}\} & \{\bar{g}_{54},g_{54}\} \end{bmatrix},$$

which is obtained simply by setting the elements $g_{i1} = 0$ $(2 \le i \le n_0)$ in $\mathcal{BD}(A)$. Notice that the transformation $A' = XA$ is not orthogonal. So, by (16), we have an orthogonal matrix $G \in \mathbb{R}^{n\times n}$ such that

$$G^T X^{-1} = Y = \prod_{i=1}^{n_0} U_{i:i}\left(1/\bar{y}_i, y_i/\bar{y}_i\right), \text{ where } \bar{y}_i > 0,\ y_i \ge 0,\ \forall\ 1 \le i \le n_0.$$

Thus, let $A^{(1)} = G^T A$, then

$$A^{(1)} = (G^T X^{-1})(XA) = YA' =: \begin{bmatrix} \{\bar{g}_{11}^{(1)},g_{11}^{(1)}\} & \{\bar{g}_{12}^{(1)},g_{12}^{(1)}\} & \{\bar{g}_{13}^{(1)},g_{13}^{(1)}\} & \{\bar{g}_{14}^{(1)},g_{14}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{22}^{(1)},g_{22}^{(1)}\} & \{\bar{g}_{23}^{(1)},g_{23}^{(1)}\} & \{\bar{g}_{24}^{(1)},g_{24}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{32}^{(1)},g_{32}^{(1)}\} & \{\bar{g}_{33}^{(1)},g_{33}^{(1)}\} & \{\bar{g}_{34}^{(1)},g_{34}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{42}^{(1)},g_{42}^{(1)}\} & \{\bar{g}_{43}^{(1)},g_{43}^{(1)}\} & \{\bar{g}_{44}^{(1)},g_{44}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{52}^{(1)},g_{52}^{(1)}\} & \{\bar{g}_{53}^{(1)},g_{53}^{(1)}\} & \{\bar{g}_{54}^{(1)},g_{54}^{(1)}\} \end{bmatrix},$$

which is obtained by applying the operations (10), (11) and (12) to YA', costing at most $O(n_0 n_K)$ subtraction-free operations. By now, possible zero singular values of $A^{(1)}$, and so A, can be further revealed by the elements in $\mathcal{BD}(A^{(1)})$.

As a second step, we go on to orthogonally deflate zero singular values of $A^{(1)}$ revealed by the elements $\bar{g}_{1j}^{(1)} = 0$ in the first row of $\mathcal{BD}(A^{(1)})$. Assume that

$$\bar{g}_{1,j_1}^{(1)} = \bar{g}_{1,j_2}^{(1)} = \ldots = \bar{g}_{1,j_l}^{(1)} = 0, \quad j_1 < j_2 < \cdots < j_l.$$

Set $\beta = (j_1 - 1, j_2 - 1, \ldots, j_l - 1)$. Then one can detect $A^{(1)}[: |\beta] = 0$. Thus, there have permutation matrices Q_{j_k} $(1 \le k \le l)$ as in (15) such that

$$A^{(1)} Q_{j_l} \ldots Q_{j_2} Q_{j_1} = \left[\, A^{(1)}[: |\beta) \;\; 0 \,\right].$$

So, we orthogonally deflate possible l zero singular value by deleting the l zero columns, and the substantial operation is to compute $\mathcal{BD}(A^{(1)}[: |\beta))$ by Algorithm 1, which costs at most $O(l n_0 n_K)$ subtraction-free operations. In this way, we can deflate the associated zero singular values to arrive at the case that $\bar{g}_{1j}^{(1)} = 1$ for all j. For for the simplicity of our description about the next operation, in our example $A^{(1)} \in \mathbb{R}^{5\times 4}$, assume that

$$A^{(1)} =: \begin{bmatrix} \{\bar{g}_{11}^{(1)},g_{11}^{(1)}\} & \{1,g_{12}^{(1)}\} & \{1,g_{13}^{(1)}\} & \{1,g_{14}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{22}^{(1)},g_{22}^{(1)}\} & \{\bar{g}_{23}^{(1)},g_{23}^{(1)}\} & \{\bar{g}_{24}^{(1)},g_{24}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{32}^{(1)},g_{32}^{(1)}\} & \{\bar{g}_{33}^{(1)},g_{33}^{(1)}\} & \{\bar{g}_{34}^{(1)},g_{34}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{42}^{(1)},g_{42}^{(1)}\} & \{\bar{g}_{43}^{(1)},g_{43}^{(1)}\} & \{\bar{g}_{44}^{(1)},g_{44}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{52}^{(1)},g_{52}^{(1)}\} & \{\bar{g}_{53}^{(1)},g_{53}^{(1)}\} & \{\bar{g}_{54}^{(1)},g_{54}^{(1)}\} \end{bmatrix}.$$

Further, we are to orthogonally zero the elements $g_{1j}^{(1)}$ in the first row of the representation. Set $\mathcal{Y} = \mathbf{biupp}(\{1, -g_{1,j+1}^{(1)}\}_{j=2}^{n_K-1})$, and denote $A'^{(1)} = A^{(1)}\mathcal{Y}$. Then

$$A'^{(1)} =: \begin{bmatrix} \{\bar{g}_{11}^{(1)},g_{11}^{(1)}\} & \{1,g_{12}^{(1)}\} & \{1,0\} & \{1,0\} \\ \{1,0\} & \{\bar{g}_{22}^{(1)},g_{22}^{(1)}\} & \{\bar{g}_{23}^{(1)},g_{23}^{(1)}\} & \{\bar{g}_{24}^{(1)},g_{24}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{32}^{(1)},g_{32}^{(1)}\} & \{\bar{g}_{33}^{(1)},g_{33}^{(1)}\} & \{\bar{g}_{34}^{(1)},g_{34}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{42}^{(1)},g_{42}^{(1)}\} & \{\bar{g}_{43}^{(1)},g_{43}^{(1)}\} & \{\bar{g}_{44}^{(1)},g_{44}^{(1)}\} \\ \{1,0\} & \{\bar{g}_{52}^{(1)},g_{52}^{(1)}\} & \{\bar{g}_{53}^{(1)},g_{53}^{(1)}\} & \{\bar{g}_{54}^{(1)},g_{54}^{(1)}\} \end{bmatrix},$$

which is obtained simply by setting the elements $g_{1j}^{(1)} = 0$ $(3 \le j \le n_K)$ in $\mathcal{BD}(A^{(1)})$. Further, by (16), we have an orthogonal matrix V such that

$$\mathcal{Y}^{-1} V = \mathcal{X} = \prod_{n_K}^{i=2} L_{i:i}\left(1/\bar{x}_i, x_i/\bar{x}_i\right), \text{ where } \bar{x}_i > 0,\ x_i \ge 0,\ \forall\ 2 \le i \le n_K.$$

Thus, let $A^{(2)} = A^{(1)}V$, then

$$A^{(2)} = (A^{(1)}\mathcal{Y})(\mathcal{Y}^{-1}V) = A'^{(1)}\mathcal{X} =: \begin{bmatrix} \{\bar{g}_{11}^{(2)},g_{11}^{(2)}\} & \{1,g_{12}^{(2)}\} & \{1,0\} & \{1,0\} \\ \{1,0\} & \{\bar{g}_{22}^{(2)},g_{22}^{(2)}\} & \{\bar{g}_{23}^{(2)},g_{23}^{(2)}\} & \{\bar{g}_{24}^{(2)},g_{24}^{(2)}\} \\ \{1,0\} & \{\bar{g}_{32}^{(2)},g_{32}^{(2)}\} & \{\bar{g}_{33}^{(2)},g_{33}^{(2)}\} & \{\bar{g}_{34}^{(2)},g_{34}^{(2)}\} \\ \{1,0\} & \{\bar{g}_{42}^{(2)},g_{42}^{(2)}\} & \{\bar{g}_{43}^{(2)},g_{43}^{(2)}\} & \{\bar{g}_{44}^{(2)},g_{44}^{(2)}\} \\ \{1,0\} & \{\bar{g}_{52}^{(2)},g_{52}^{(2)}\} & \{\bar{g}_{53}^{(2)},g_{53}^{(2)}\} & \{\bar{g}_{54}^{(2)},g_{54}^{(2)}\} \end{bmatrix},$$

which is obtained by applying the operations (10), (11) and (12) to $A'^{(1)}\mathcal{X}$, costing at most $O(n_0 n_K)$ subtraction-free operations. Here, $g_{12}^{(2)}$ is not eliminated, since attempting to eliminate it will disturb the previously reduced element pairs $\{1, 0\}$.

Therefore, we have orthogonally deflated some zero singular values of A to get the resulting matrix

$$A^{(2)} = G^T AV = \begin{bmatrix} 1 & 0 \\ 0 & A'^{(2)} \end{bmatrix} U_{1:1}(g_{11}^{(2)}, g_{11}^{(2)} g_{12}^{(2)}),$$

where $g_{11}^{(2)} \neq 0$, and

$$\mathcal{BD}(A'^{(2)}) = \begin{bmatrix} \{\bar{g}_{22}^{(2)},g_{22}^{(2)}\} & \{\bar{g}_{23}^{(2)},g_{23}^{(2)}\} & \{\bar{g}_{24}^{(2)},g_{24}^{(2)}\} \\ \{\bar{g}_{32}^{(2)},g_{32}^{(2)}\} & \{\bar{g}_{33}^{(2)},g_{33}^{(2)}\} & \{\bar{g}_{34}^{(2)},g_{34}^{(2)}\} \\ \{\bar{g}_{42}^{(2)},g_{42}^{(2)}\} & \{\bar{g}_{43}^{(2)},g_{43}^{(2)}\} & \{\bar{g}_{44}^{(2)},g_{44}^{(2)}\} \\ \{\bar{g}_{52}^{(2)},g_{52}^{(2)}\} & \{\bar{g}_{53}^{(2)},g_{53}^{(2)}\} & \{\bar{g}_{54}^{(2)},g_{54}^{(2)}\} \end{bmatrix}.$$

More importantly, the remaining zero singular values of A are revealed by $\mathcal{BD}(A'^{(2)})$. Hence, proceeding with such deflations on the representation and going on, finally all the zero singular values of A are orthogonally deflated, arriving at a full-rank bidiagonal matrix, e.g., our example $A \in \mathbb{R}^{5\times 4}$ is orthogonally deflated into the full-rank bidiagonal matrix

$$B = \prod_{4}^{i=1} U_{i:i}(g_{ii}^{(4)}, g_{ii}^{(4)} g_{i,i+1}^{(4)}) =: \begin{bmatrix} \{\bar{g}_{11}^{(4)},g_{11}^{(4)}\} & \{1,g_{12}^{(4)}\} & \{1,0\} & \{1,0\} \\ \{1,0\} & \{\bar{g}_{22}^{(4)},g_{22}^{(2)}\} & \{1,g_{23}^{(2)}\} & \{1,0\} \\ \{1,0\} & \{1,0\} & \{\bar{g}_{33}^{(4)},g_{33}^{(4)}\} & \{1,g_{34}^{(4)}\} \\ \{1,0\} & \{1,0\} & \{1,0\} & \{\bar{g}_{44}^{(4)},g_{44}^{(4)}\} \\ \{1,0\} & \{1,0\} & \{1,0\} & \{1,0\} \end{bmatrix}.$$

By now the deflation has been completed.

The whole deflation process costs at most $O(\max\{n_0, n_K\} \cdot n_0 n_K)$ subtraction-free operations. Plus the cost of computing $\mathcal{BD}(A)$, the total cost is at most

$$O(rS + \max\{n_0^2 n_K, n_K^2 n_0\}).$$

4.2 The periodic deflation for a general case

Given a general product $A \in \mathbb{R}^{n_0 \times n_K}$ as in (1) having S nontrivial element pairs, the direct method for deflating zero singular values of A is that we first compute $\mathcal{BD}(A)$ and then apply the the deflation method presented in Section 4.1 to $\mathcal{BD}(A)$. The total cost of the direct method is at most

$$O(\min\{n_0, n_K\} \cdot S + \max\{n_0^2 n_K, n_K^2 n_0\}), \tag{17}$$

which is pessimistic if n_0 and n_K are very large. In this section, basing on the minimum dimension $r = \min_{0\le i\le K}\{n_i\}$, we develop a periodic deflation method such that the deflation is effectively carried out costing at most

$$O(rS + \max\{n_0^2 r, n_K^2 r\}),$$

which, compared with (17), is very preferable when $r \ll \min\{n_0, n_K\}$.

Recalling the factor B_i, for $1 \le i \le K$, of the product A as in (1) has the dimension $n_{i-1} \times n_i$, we set the minimum dimension $r = n_T$, and split A into $A = A_2 A_1$, where

$$A_2 = \prod_{i=1}^{T} B_i \in \mathbb{R}^{n_0\times r},\ \ A_1 = \prod_{i=T+1}^{K} B_i \in \mathbb{R}^{r\times n_K}.$$

with the convention $\prod_i^j = I_n$ if $j < i$. Let S_1 and S_2 be the numbers of nontrivial element pairs in A_1 and A_2, respectively. Then the total number is $S = S_1 + S_2$. Instead of $\mathcal{BD}(A)$, we compute $\mathcal{BD}(A_1)$ and $\mathcal{BD}(A_2)$, which cost at most $O(S_1 r)$ and $O(S_2 r)$ subtraction-free operations, respectively. So, the total cost is $O((S_1+S_2)\cdot r) = O(Sr)$.

In what follows, we illustrate how to orthogonally deflate zero singular values of $A = A_2A_1$ focusing on $\mathcal{BD}(A_1)$ and $\mathcal{BD}(A_2)$.

Step 1. First, we delete zero rows, indexed by $\alpha \in Q_{t_1,r}$, of $A_1 =: (\{\bar{a}_{ij}, a_{ij}\})$ revealed by the zero elements $\bar{a}_{i1} = 0$, and remove the corresponding columns of A_2, arriving at

$$A_2A_1 = A_2[:|\alpha)\cdot A_1(\alpha|:],$$

where $\mathcal{BD}(A_1(\alpha|:])$ and $\mathcal{BD}(A_2[:|\alpha))$ are computed by Algorithm 1 costing at most $O(t_1 r n_K)$ and $O(t_1 r n_0)$ subtraction-free operations, respectively. In this way we arrive at the submatrices $A_1' \in \mathbb{R}^{r'\times n_K}$ and $A_2' \in \mathbb{R}^{n_0\times r'}$ such that $A = A_2'A_1'$, where $\mathcal{BD}(A_1') = (\{\bar{a}_{ij}^{(1)}, a_{ij}^{(1)}\})$ satisfies that all $\bar{a}_{i1}^{(1)} = 1$. We go on to zero the elements $a_{i1}^{(1)}$. Set $L_1 = \textbf{bilow}(\{1, -a_{i+1,1}^{(1)}\}_{i=1}^{r'-1})$. Then

$$A = A_2'A_1' = (A_2'L_1^{-1})(L_1A_1') = A_2''A_1^{(1)}, \tag{18}$$

where

$$A_1^{(1)} = L_1A_1' =: \begin{bmatrix} \{\bar{a}_{11}^{(1)}, a_{11}^{(1)}\} & \{\bar{a}_{12}^{(1)}, a_{12}^{(1)}\} & \cdots & \{\bar{a}_{1n_K}^{(1)}, a_{1n_K}^{(1)}\} \\ \{1, 0_1\} & \{\bar{a}_{22}^{(1)}, a_{22}^{(1)}\} & \cdots & \{\bar{a}_{2n_K}^{(1)}, a_{2n_K}^{(1)}\} \\ \vdots & \vdots & \vdots & \vdots \\ \{1, 0_1\} & \{\bar{a}_{r'2}^{(1)}, a_{r'2}^{(1)}\} & \cdots & \{\bar{a}_{r'n_K}^{(1)}, a_{r'n_K}^{(1)}\} \end{bmatrix} \in \mathbb{R}^{r'\times n_K}$$

is obtained simply by setting $a_{i1}^{(1)} = 0_1$ $(2 \le i \le r')$ in $\mathcal{BD}(A_1')$, and the representation of $A_2'' = A_2'L_1^{-1} = A_2'\cdot\prod_{r'-1}^{i=1} L_{i:i}(1, a_{i+1,1}^{(1)}) \in \mathbb{R}^{n_0\times r'}$ is computed by using (10), (11) and (12), costing at most $O(r'n_0)$ subtraction-free operations.

Step 2. Consider that deleting zero rows of the resulting matrix $A_2'' =: (\{\bar{b}_{ij}, b_{ij}\})$ revealed by the zero elements $\bar{b}_{i1} = 0$, whose number is assumed to be $t_2 = n_0 - n_0'$, is meant deflating possible zero singular values of $A = A_2'' A_1^{(1)}$. Thus, by the deflation in Section 4.1, we construct an orthogonal matrix G_1 to deflate the zero singular values, which costs at most $O(t_2 n_0 r' + n_0' r')$ subtraction-free operations, arriving at

$$A^{(1)} = (G_1^T A_2'') A_1^{(1)} = A_2^{(1)} A_1^{(1)}, \tag{19}$$

where

$$A_2^{(1)} = G_1^T A_2'' =: \begin{bmatrix} \{\bar{b}_{11}^{(1)}, b_{11}^{(1)}\} & \{\bar{b}_{12}^{(1)}, b_{12}^{(1)}\} & \cdots & \{\bar{b}_{1r'}^{(1)}, b_{1r'}^{(1)}\} \\ \{1, 0_1\} & \{\bar{b}_{22}^{(1)}, b_{22}^{(1)}\} & \cdots & \{\bar{b}_{2r'}^{(1)}, b_{2r'}^{(1)}\} \\ \vdots & \vdots & \vdots & \vdots \\ \{1, 0_1\} & \{\bar{b}_{n_0',2}^{(1)}, b_{n_0',2}^{(1)}\} & \cdots & \{\bar{b}_{n_0',r'}^{(1)}, b_{n_0',r'}^{(1)}\} \end{bmatrix} \in \mathbb{R}^{n_0' \times r'}.$$

Step 3. Further, we delete zero columns, indexed by $\beta \in Q_{t_3,r'}$, of $A_2^{(1)} =: (\{\bar{b}_{ij}^{(1)}, b_{ij}^{(1)}\})$ revealed by the zero elements $\bar{b}_{1j}^{(1)} = 0$, and remove the corresponding rows of $A_1^{(1)}$, arriving at

$$A_2^{(1)} A_1^{(1)} = A_2^{(1)}[: |\beta) \cdot A_1^{(1)}(\beta| :],$$

where $\mathcal{BD}(A_2^{(1)}[: |\beta))$ and $\mathcal{BD}(A_1^{(1)}(\beta| :])$ are computed by Algorithm 1 costing at most $O(t_3 r' n_0')$ and $O(t_3 r' n_K)$ subtraction-free operations, respectively. In this way we arrive at the submatrices $A_2'^{(1)} \in \mathbb{R}^{n_0' \times r''}$ and $A_1'^{(1)} \in \mathbb{R}^{r'' \times n_K}$ such that $A^{(1)} = A_2'^{(1)} A_1'^{(1)}$, where $\mathcal{BD}(A_2'^{(1)}) = (\{\bar{b}_{ij}^{(2)}, b_{ij}^{(2)}\})$ satisfies that all $\bar{b}_{1j}^{(2)} = 1$. We go on to zero the elements $b_{1j}^{(2)}$. Set $U_1 = \mathbf{biupp}(\{1, -b_{1,j+1}^{(2)}\}_{j=1}^{r''-1})$. Then

$$A^{(1)} = (A_2'^{(1)} U_1)(U_1^{-1} A_1'^{(1)}) = A_2^{(2)} A_1''^{(1)},$$

where

$$A_2^{(2)} = A_2'^{(1)} U_1 =: \begin{bmatrix} \{\bar{b}_{11}^{(2)}, b_{11}^{(2)}\} & \{1, 0_2\} & \cdots & \{1, 0_2\} \\ \{1, 0_1\} & \{\bar{b}_{22}^{(2)}, b_{22}^{(2)}\} & \cdots & \{\bar{b}_{2r''}^{(2)}, b_{2r''}^{(2)}\} \\ \vdots & \vdots & \vdots & \vdots \\ \{1, 0_1\} & \{\bar{b}_{n_0',2}^{(2)}, b_{n_0',2}^{(2)}\} & \cdots & \{\bar{b}_{n_0',r''}^{(2)}, b_{n_0',r''}^{(2)}\} \end{bmatrix} \in \mathbb{R}^{n_0' \times r''}$$

is obtained simply by setting the elements $b_{1j}^{(2)} = 0_2$ $(2 \le j \le r'')$ in $\mathcal{BD}(A_2'^{(1)})$, and

$$A_1''^{(1)} = U_1^{-1} A_1'^{(1)} =: \begin{bmatrix} \{\bar{a}_{11}'', a_{11}''\} & \{\bar{a}_{12}'', a_{12}''\} & \cdots & \{\bar{a}_{1n_K}'', a_{1n_K}''\} \\ \{1, 0_1\} & \{\bar{a}_{22}'', a_{22}''\} & \cdots & \{\bar{a}_{2n_K}'', a_{2n_K}''\} \\ \vdots & \vdots & \vdots & \vdots \\ \{1, 0_1\} & \{\bar{a}_{r'',2}'', a_{r'',2}''\} & \cdots & \{\bar{a}_{r'',n_K}'', a_{r'',n_K}''\} \end{bmatrix} \in \mathbb{R}^{r'' \times n_K}$$

is computed by (10), (11) and (12), costing at most $O(r''n_K)$ subtraction-free operations. In particular, the previously reduced element pairs $\{1, 0_1\}$ of $A_1'^{(1)}$ and $A_2'^{(1)}$ are not disturbed.

Step 4. Notice that deleting zero columns of the resulting matrix $A_1''^{(1)} =: (\{\bar{a}''_{ij}, a''_{ij}\})$ revealed by the zero elements $\bar{a}''_{1j} = 0$, whose number is assumed to be $t_4 = n_K - n'_K$, is meant deflating possible zero singular values of $A^{(1)} = A_2^{(2)} A_1''^{(1)}$. Thus, by the deflation in Section 4.1, we construct an orthogonal matrix V_1 to deflate the zero singular values, which costs at most $O(t_4 r'' n_K + r'' n'_K)$ subtraction-free operations, arriving at

$$A^{(2)} = A_2^{(2)} (A_1''^{(1)} V_1) = A_2^{(2)} A_1^{(2)}, \tag{20}$$

where

$$A_1^{(2)} = A_1''^{(1)} V_1 =: \begin{bmatrix} \{\bar{a}_{11}^{(2)}, a_{11}^{(2)}\} & \{1, a_{12}^{(2)}\} & \{1, 0_2\} & \cdots & \{1, 0_2\} \\ \{1, 0_1\} & \{\bar{a}_{22}^{(2)}, a_{22}^{(2)}\} & \{\bar{a}_{23}^{(2)}, a_{23}^{(2)}\} & \cdots & \{\bar{a}_{2n'_K}^{(2)}, a_{2n'_K}^{(2)}\} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \{1, 0_1\} & \{\bar{a}_{r'',2}^{(2)}, a_{r'',2}^{(2)}\} & \{\bar{a}_{r'',3}^{(2)}, a_{r'',3}^{(2)}\} & \cdots & \{\bar{a}_{r'',n'_K}^{(2)}, a_{r'',n'_K}^{(2)}\} \end{bmatrix}.$$

In particular, the previously reduced element pairs $\{1, 0_1\}$ of $A_1''^{(1)}$ are not disturbed.

Therefore, we have orthogonally deflated the zero singular values of $A = A_2 A_1$ to arrive at the resulting matrix

$$A^{(2)} = G_1^T A V_1 = A_2^{(2)} A_1^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & A_2'^{(2)} \cdot A_1'^{(2)} \end{bmatrix} \cdot U_{1:1}(b_{11}^{(2)} a_{11}^{(2)}, b_{11}^{(2)} a_{11}^{(2)} a_{12}^{(2)}),$$

where $b_{11}^{(2)} a_{11}^{(2)} \neq 0$, and

$$\mathcal{BD}(A_2'^{(2)}) = \begin{bmatrix} \{\bar{b}_{22}^{(2)}, b_{22}^{(2)}\} & \cdots & \{\bar{b}_{2r''}^{(2)}, b_{2r''}^{(2)}\} \\ \vdots & \vdots & \vdots \\ \{\bar{b}_{n'_0,2}^{(2)}, b_{n'_0,2}^{(2)}\} & \cdots & \{\bar{b}_{n'_0,r''}^{(2)}, b_{n'_0,r''}^{(2)}\} \end{bmatrix},$$

and

$$\mathcal{BD}(A_1'^{(2)}) = \begin{bmatrix} \{\bar{a}_{22}^{(2)}, a_{22}^{(2)}\} & \cdots & \{\bar{a}_{2n'_K}^{(2)}, a_{2n'_K}^{(2)}\} \\ \vdots & \vdots & \vdots \\ \{\bar{a}_{r'',2}^{(2)}, a_{r'',2}^{(2)}\} & \cdots & \{\bar{a}_{r'',n'_K}^{(2)}, a_{r'',n'_K}^{(2)}\} \end{bmatrix}.$$

More importantly, the remaining zero singular values of A are revealed by $\mathcal{BD}(A_1'^{(2)})$ and $\mathcal{BD}(A_2'^{(2)})$. Therefore, proceeding with such deflations on the representations and going on, we subsequently get the orthogonal matrices G' and V' such that the zero singular values of $A' = A_2'^{(2)} A_1'^{(2)}$ are deflated to get that

$$B' = G'^T A' V' = \begin{bmatrix} \bar{B}' & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(n_0'-1)\times(n_K'-1)},$$

where $\bar{B}'$ is bidiagonal of full rank.

Now, denote $G = G_1 \cdot \mathrm{diag}(1, G')$, $V = V_1 \cdot \mathrm{diag}(1, V')$ and $\bar{B} = \mathrm{diag}(1, \bar{B}') \cdot U_{1:1}(b_{11}^{(2)} a_{11}^{(2)}, b_{11}^{(2)} a_{11}^{(2)} a_{12}^{(2)})$. Then we conclude that all the zero singular values of $A = A_2 A_1$ are orthogonally deflated to get that

$$B = G^T A V = \begin{bmatrix} \bar{B} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n_0 \times n_K}, \tag{21}$$

here, $G \in \mathbb{R}^{n_0 \times n_0}$ and $V \in \mathbb{R}^{n_K \times n_K}$ are orthogonal, and $\bar{B}$ is bidiagonal of full rank.

Therefore, we summarize the periodic deflation as Algorithm 2. The whole deflation process consists of the steps of deleting (eliminating) one zero row (column) for $\mathcal{BD}(A_1) \in \mathbb{R}^{r \times n_K}$ and $\mathcal{BD}(A_2) \in \mathbb{R}^{n_0 \times r}$, each of which costs at most $O(rn_K)$ and $O(rn_0)$ subtraction-free operations, respectively. So, the cost of the deflation is at most

$$O(\max\{n_0, n_K, r\} \cdot \max\{rn_0, rn_K\}).$$

Plus the costs of computing $\mathcal{BD}(A_1)$ and $\mathcal{BD}(A_2)$, the total cost is at most

$$O(rS + \max\{n_0^2 r, n_K^2 r\}),$$

which is preferable when r is small enough. In addition, when accumulating the orthogonal matrices G and V, the costs are $O(n_0^3)$ and $O(n_K^3)$, respectively.

4.3 The overall SVD algorithm

Once all the zero singular values of A are orthogonally deflated to get $B = \mathrm{diag}(\bar{B}, 0)$ as in (21), the nonzero ones are computed by applying the LAPACK routine `DLASQ1` to the full-rank bidiagonal matrix $\bar{B}$ of order $\bar{r}$, costing $O(\bar{r}^2)$ operations [11, 15, 32]. Therefore, we summarize Algorithm 3 to compute the SVD of the product A as in (1), which costs at most $O(rS + \max\{n_0^2 r, n_K^2 r\})$ arithmetic operations plus the $O(n_K^3)$ and $O(n_0^3)$ costs of accumulating the left and right singular vector matrices, respectively.

In what follows, we are to show that all the zero singular values are deflated exactly, and all the nonzero ones are computed to high relative accuracy.

Remind that every single subtraction-free operation causes at most μ (2μ) relative error in any minor (singular value) of a nonnegative bidiagonal product [23], where μ is the unit roundoff. In the sequel, the computed quantity wears a hat.

Theorem 1. *Let $A = B_1 B_2 \ldots B_K \in \mathbb{R}^{n_0 \times n_K}$ be as in (1) having S nontrivial element pairs in the bidiagonal factors $B_i \in \mathbb{R}^{n_{i-1} \times n_i}$ ($1 \le i \le K$), and let $r = \min_{0 \le k \le K}\{n_k\}$. Denote by σ_i and $\hat{\sigma}_i$ the exact and computed nonzero descending-ordered singular values of A by Algorithm 3, respectively. Set $C = rS + \max\{n_0^2 r, n_K^2 r\}$. Then all the*

Algorithm 2 Given the product $A = B_1 B_2 \dots B_K \in \mathbb{R}^{n_0 \times n_K}$ as in (1), the algorithm orthogonally deflate zero singular values of A into a bidiagonal matrix.

Input: the nontrivial element pairs in the factors $B_i \in \mathbb{R}^{n_{i-1} \times n_i}$ $(1 \le i \le K)$.
Output: the bidiagonal matrix $B = G^T A V$ as in (21)

Set $n = n_0$, $m = n_K$, $G = I_n$ and $V = I_m$.
Set $t = 1$ and $r = \min_{0 \le l \le K}\{n_l\} = n_T$.
Split A into $A = A_2 A_1$ with $A_1 = B_{T+1} \dots B_K$ and $A_2 = B_1 \dots B_T$.
First, compute $\mathcal{BD}(A_1) = (\{\bar{g}_{ij}^{(1)}, g_{ij}^{(1)}\})$ and $\mathcal{BD}(A_2) = (\{\bar{g}_{ij}^{(2)}, g_{ij}^{(2)}\})$.
while $t \le r$ **do**
  if $0 < T < K$ **then**
    for $i = r : -1 : t+1$ **do**
      if $\bar{g}_{it}^{(1)} = 0$ **then**
        Compute $A_1 := A_1(i-1| :]$ and $A_2 := A_2[: |i-1)$ by Alg. 1.
        $r := r - 1$
      end if
    end for
    Denote $\mathcal{BD}(A_1) = (\{\bar{g}_{ij}^{(1)}, g_{ij}^{(1)}\})$, and let $L_t^{(1)} = \textbf{bilow}(\{1, -g_{i+1,t}^{(1)}\}_{i=t}^{r-1})$.
    Compute $A_1 := L_t^{(1)} A_1$ simply by setting $g_{t+1,t}^{(1)} = \dots = g_{rt}^{(1)} = 0$.
    Compute $A_2 := A_2 (L_t^{(1)})^{-1}$ by (10), (11) and (12).
  end if
  Set $s = 2$ if $T \neq 0$; otherwise, $s = 1$.
  for $i = n_0 : -1 : t+1$ **do**
    if $\bar{g}_{it}^{(s)} = 0$ **then**
      Compute $A_s := A_s(i-1| :]$ by Alg. 1.
      $G := G \cdot \mathrm{diag}(P_{i-1}, I_{n-n_0})$, where $P_{i-1} \in \mathbb{R}^{n_0 \times n_0}$ is as in (15).
      $n_0 := n_0 - 1$.
    end if
  end for
  Denote $\mathcal{BD}(A_s) = (\{\bar{g}_{ij}^{(s)}, g_{ij}^{(s)}\})$, and let $L_t^{(s)} = \textbf{bilow}(\{1, -g_{i+1,t}^{(s)}\}_{i=t}^{n_0-1})$.
  Compute $A_s := L_t^{(s)} A_s$ simply by setting $g_{t+1,t}^{(s)} = \dots = g_{n_0,t}^{(s)} = 0$.
  Compute $(G_t^{(s)})^T (L_t^{(s)})^{-1} = (U_t^{(s)})^{-1}$ by (16).
  Compute $A_s := (U_t^{(s)})^{-1} A_s$ by (10), (11) and (12).
  $G := G \cdot \mathrm{diag}(G_t^{(s)}, I_{n-n_0})$.
  if $0 < T < K$ **then**
    for $j = r : -1 : t+1$ **do**
      if $\bar{g}_{tj}^{(2)} = 0$ **then**
        Compute $A_2 := A_2[: |j-1)$ and $A_1 := A_1(j-1| :]$ by Alg. 1.
        $r := r - 1$
      end if
    end for
    Denote $\mathcal{BD}(A_2) = (\{\bar{g}_{ij}^{(2)}, g_{ij}^{(2)}\})$, and let $U_t^{(2)} = \textbf{biupp}(\{1, -g_{t,j+1}^{(2)}\}_{j=t}^{r-1})$.
    Compute $A_2 := A_2 U_t^{(2)}$ simply by setting $g_{t,t+1}^{(2)} = \dots = g_{tr}^{(2)} = 0$.
    Compute $A_1 := (U_t^{(2)})^{-1} A_1$ by (10), (11) and (12).
  end if
  Set $z = 1$ if $T \neq K$, otherwise, $z = 2$.
  for $j = n_K : -1 : t+1$ **do**
    if $\bar{g}_{tj}^{(z)} = 0$ **then**
      Compute $A_z := A_z[: |j-1)$ by Alg. 1.
      $V := V \cdot \mathrm{diag}(Q_{j-1}, I_{m-n_K})$, where $Q_{j-1} \in \mathbb{R}^{n_K \times n_K}$ is as in (15).
      $n_K := n_K - 1$.
    end if
  end for
  Denote $\mathcal{BD}(A_z) = (\{\bar{g}_{ij}^{(z)}, g_{ij}^{(z)}\})$, and let $U_t^{(z)} = \textbf{biupp}(\{1, -g_{t,i+1}^{(z)}\}_{i=t+1}^{n_K-1})$.
  Compute $A_z := A_z U_t^{(z)}$ simply by setting $g_{t,t+2}^{(z)} = \dots = g_{t,n_K}^{(z)} = 0$.
  Compute $(U_t^{(z)})^{-1} V_t^{(z)} = (L_t^{(z)})^{-1}$ by (16).
  Compute $A_z := A_z (L_t^{(z)})^{-1}$ by (10), (11) and (12).
  $V := V \cdot \mathrm{diag}(V_t^{(z)}, I_{m-n_K})$.
  Set $t := t + 1$.
end while
The bidiagonal matrix $B = G^T A V = \mathrm{diag}(A_2 A_1, 0) \in \mathbb{R}^{n_0 \times n_K}$.

Algorithm 3 Given the product $A = B_1 B_2 \dots B_K \in \mathbb{R}^{n_0 \times n_K}$ as in (1), the algorithm computes the SVD of A.

Input: the nontrivial element pairs in the factors $B_i \in \mathbb{R}^{n_{i-1} \times n_i}$ $(1 \le i \le K)$.
Output: the SVD $A = U \cdot \text{diag}(\Sigma, 0) \cdot V^T$

1: Deflate zero singular values of A into $\mathcal{U}_1^T A \mathcal{V}_1 = \text{diag}(\bar{B}, 0)$ by Algorithm 2.
2: Compute the SVD $\mathcal{U}_2^T \bar{B} \mathcal{V}_2 = \Sigma$ by the LAPACK routine `DLASQ1`.
3: Then the SVD $A = U \cdot \text{diag}(\Sigma, 0) \cdot V^T$, where $U = \mathcal{U}_1 \cdot \text{diag}(\mathcal{U}_2, I_{n_0 - \bar{r}}) \in \mathbb{R}^{n_0 \times n_0}$ and $V = \mathcal{V}_1 \cdot \text{diag}(\mathcal{V}_2, I_{n_K - \bar{r}}) \in \mathbb{R}^{n_K \times n_K}$ are orthogonal.

zero singular values are deflated exactly, and all the nonzero ones are computed as

$$\hat{\sigma}_i = (1 + \eta_i)\sigma_i, \ |\eta_i| \le \frac{O(2C\mu)}{1 - O(2C\mu)}, \quad \forall\ 1 \le i \le \text{rank}(A).$$

Proof. Spilt the product A into $A = A_2 A_1$ with $A_1 \in \mathbb{R}^{r \times n_K}$ and $A_2 \in \mathbb{R}^{n_0 \times r}$. According to the procedure of Algorithm 3, the following statements hold.

- First, we compute $\mathcal{BD}(A_1)$ and $\mathcal{BD}(A_2)$ costing at most $O(S_1 r)$ and $O(S_2 r)$ subtraction-free operations, which cause at most $O(S_1 r\mu)$ and $O(S_2 r\mu)$ relative errors for any minor of the computed quantities $\hat{A}_1 =: \mathcal{BD}(\hat{A}_1)$ and $\hat{A}_2 =: \mathcal{BD}(\hat{A}_2)$, respectively; i.e., for any $\alpha \in Q_{k,n_0}$, $\gamma \in Q_{k,r}$ and $\beta \in Q_{k,n_K}$,

$$\begin{cases} \det\hat{A}_2[\alpha|\gamma] = (1 + \epsilon_{\alpha,\gamma})\det A_2[\alpha|\gamma], \ |\epsilon_{\alpha,\gamma}| \le \frac{O(S_2 r\mu)}{1 - O(S_2 r\mu)}, \\ \det\hat{A}_1[\gamma|\beta] = (1 + \epsilon_{\gamma,\beta})\det A_1[\gamma|\beta], \ |\epsilon_{\gamma,\beta}| \le \frac{O(S_1 r\mu)}{1 - O(S_1 r\mu)}, \end{cases}$$

  and there is $O((S_1 + S_2) r\mu)$ relative error for any singular value of $\hat{A}_2 \hat{A}_1$, i.e.,

$$\sigma_i(\hat{A}_2 \hat{A}_1) = (1 + \delta_i)\sigma_i(A_2 A_1) = (1 + \delta_i)\sigma_i, \ |\delta_i| \le \frac{O(2(S_1 + S_2) r\mu)}{1 - O(2(S_1 + S_2) r\mu)}, \ \forall\ i. \quad (22)$$

  Notice that $\det A_2[\alpha|\gamma] \ge 0$ and $\det A_1[\gamma|\beta] \ge 0$. Thus,

$$\det(\hat{A}_2 \hat{A}_1)[\alpha|\beta] = (1 + \epsilon_{\alpha,\beta}) \cdot \det(A_2 A_1)[\alpha|\beta], \ |\epsilon_{\alpha,\beta}| \le \frac{O((S_1 + S_2) r\mu)}{1 - O((S_1 + S_2) r\mu)}.$$

  This means that $\det(\hat{A}_2 \hat{A}_1)[\alpha|\beta] = 0$ if and only if $\det(A_2 A_1)[\alpha|\beta] = 0$. So,

$$\text{rank}(\hat{A}_2 \hat{A}_1) = \text{rank}(A_2 A_1) = \text{rank}(A). \quad (23)$$

- We delete zero rows and the corresponding columns of $\hat{A}_1$ and $\hat{A}_2$ to get the $r' \times n_K$ and $n_0 \times r'$ submatrices $A_1' =: \mathcal{BD}(A_1')$ and $A_2' =: \mathcal{BD}(A_2')$ by Algorithm 1, costing at most $O((r - r') r n_K)$ and $O((r - r') r n_0)$ subtraction-free operations, respectively. Then for the computed quantities $\hat{A}_1' =: \mathcal{BD}(\hat{A}_1')$ and $\hat{A}_2' =: \mathcal{BD}(\hat{A}_2')$,

$$\text{rank}(\hat{A}_2' \hat{A}_1') = \text{rank}(A_2' A_1') = \text{rank}(\hat{A}_2 \hat{A}_1), \quad (24)$$

and

$$\sigma_i(\hat{A}_2'\hat{A}_1') = (1+\delta_i')\sigma_i(A_2'A_1') = (1+\delta_i')\sigma_i(\hat{A}_2\hat{A}_1), \tag{25}$$

where

$$|\delta_i'| \le \frac{O(2r(r-r')(n_0+n_K)\mu)}{1-O(2r(r-r')(n_0+n_K)\mu)}, \ \forall\ i.$$

Further, construct a bidiagonal matrix L_1 to zero the elements in the first column of $\mathcal{BD}(\hat{A}_1')$, arriving at $A_1^{(1)} = L_1\hat{A}_1'$ and $A_2'' = \hat{A}_2'L_1^{-1}$ as in (18), costing at most $O(r'n_0)$ subtraction-free operations. Then for the computed quantities $\hat{A}_1^{(1)} =: \mathcal{BD}(\hat{A}_1^{(1)})$ and $\hat{A}_2'' =: \mathcal{BD}(\hat{A}_2'')$,

$$\operatorname{rank}(\hat{A}_2''\hat{A}_1^{(1)}) = \operatorname{rank}(A_2''A_1^{(1)}) = \operatorname{rank}(\hat{A}_2'\hat{A}_1'), \tag{26}$$

and

$$\sigma_i(\hat{A}_2''\hat{A}_1^{(1)}) = (1+\delta_i'')\sigma_i(A_2''A_1^{(1)}) = (1+\delta_i'')\sigma_i(\hat{A}_2'\hat{A}_1'), \tag{27}$$

where

$$|\delta_i''| \le \frac{O(2r'n_0\mu)}{1-O(2r'n_0\mu)}, \ \forall\ i.$$

- We construct an orthogonal matrix G_1 to deflate zero singular values of $\hat{A}_2''\hat{A}_1^{(1)}$, arriving at $A_2^{(1)} = G_1^T\hat{A}_2''$ as in (19), costing at most $O((n_0-n_0')r'n_0 + r'n_0')$ subtraction-free operations. Then for the computed quantity $\hat{A}_2^{(1)} =: \mathcal{BD}(\hat{A}_2^{(1)})$,

$$\operatorname{rank}(\hat{A}_2^{(1)}\hat{A}_1^{(1)}) = \operatorname{rank}(A_2^{(1)}\hat{A}_1^{(1)}) = \operatorname{rank}(\hat{A}_2''\hat{A}_1^{(1)}), \tag{28}$$

and

$$\sigma_i(\hat{A}_2^{(1)}\hat{A}_1^{(1)}) = (1+\delta_i''')\sigma_i(A_2^{(1)}\hat{A}_1^{(1)}) = (1+\delta_i''')\sigma_i(\hat{A}_2''\hat{A}_1^{(1)}), \tag{29}$$

where

$$|\delta_i'''| \le \frac{O(2((n_0-n_0')r'n_0 + r'n_0')\mu)}{1-O(2((n_0-n_0')r'n_0 + r'n_0')\mu)}, \ \forall\ i.$$

- Now, we have obtained that by (24), (26) and (28),

$$\operatorname{rank}(\hat{A}_2^{(1)}\hat{A}_1^{(1)}) = \operatorname{rank}(\hat{A}_2\hat{A}_1), \tag{30}$$

and by (25), (27) and (29),

$$\sigma_i(\hat{A}_2^{(1)}\hat{A}_1^{(1)}) = (1+\delta_i''')(1+\delta_i'')(1+\delta_i')\sigma_i(\hat{A}_2\hat{A}_1) = (1+\delta_i^{(1)})\sigma_i(\hat{A}_2\hat{A}_1), \tag{31}$$

where, let

$$C_1 = r(r-r')(n_0+n_K) + r'n_0 + (n_0-n_0')r'n_0 + r'n_0',$$

then

$$|\delta_i^{(1)}| \le \frac{O(2C_1\mu)}{1-O(2C_1\mu)}, \ \forall\ i.$$

When continuing with such deflations on $\hat{A}_1^{(1)} \in \mathbb{R}^{r' \times n_K}$ and $\hat{A}_2^{(1)} \in \mathbb{R}^{n_0' \times r'}$ to get $A_1^{(2)} \in \mathbb{R}^{r'' \times n_K'}$ and $A_1^{(2)} \in \mathbb{R}^{n_0' \times r''}$ as in (20), which costs at most $O(C_2)$ subtraction-free operations with

$$C_2 = r'(r' - r'')(n_0' + n_K) + r'' n_K + (n_K - n_K') r'' n_K + r'' n_K',$$

using the same argument above, we have that for the computed quantities $\hat{A}_1^{(2)} =: \mathcal{BD}(\hat{A}_1^{(2)})$ and $\hat{A}_2^{(2)} =: \mathcal{BD}(\hat{A}_2^{(2)})$,

$$\operatorname{rank}(\hat{A}_2^{(2)} \hat{A}_1^{(2)}) = \operatorname{rank}(\hat{A}_2^{(1)} \hat{A}_1^{(1)}), \tag{32}$$

and

$$\sigma_i(\hat{A}_2^{(2)} \hat{A}_1^{(2)}) = (1 + \delta_i^{(2)}) \sigma_i(\hat{A}_2^{(1)} \hat{A}_1^{(1)}), \ |\delta_i^{(2)}| \leq \frac{O(2C_2\mu)}{1 - O(2C_2\mu)}, \ \forall\, i. \tag{33}$$

Moreover, when proceeding with the deflations on $\hat{A}_2^{(2)}$ and $\hat{A}_1^{(2)}$ to get the bidiagonal matrix $\bar{B}$ as in (21), costing at most $O(C - rS - C_1 - C_2)$ subtraction-free operations, we inductively get that for the computed quantity $\hat{\bar{B}}$,

$$\operatorname{rank}(\hat{\bar{B}}) = \operatorname{rank}(\hat{A}_2^{(2)} \hat{A}_1^{(2)}),$$

and

$$\sigma_i(\hat{\bar{B}}) = (1 + \zeta_i') \sigma_i(\hat{A}_2^{(2)} \hat{A}_1^{(2)}), \quad |\zeta_i'| \leq \frac{O(2(C - rS - C_1 - C_2)\mu)}{1 - O(2(C - rS - C_1 - C_2)\mu)}, \ \forall\, i.$$

Finally, when applying the LAPACK routine `DLASQ1` [32] to $\hat{\bar{B}}$ of order $\bar{r}$, the computed singular values $\hat{\sigma}_i$ satisfy that

$$\hat{\sigma}_i = (1 + \zeta_i'') \sigma(\hat{\bar{B}}), \quad |\zeta_i''| \leq \frac{O(\bar{r}^2\mu)}{1 - O(\bar{r}^2\mu)}, \ \forall\, i.$$

Therefore, according to the analysis above, we conclude that

$$\operatorname{rank}(\hat{\bar{B}}) = \operatorname{rank}(\hat{A}_2^{(2)} \hat{A}_1^{(2)}) = \operatorname{rank}(A) = \bar{r},$$

which means that all the zero singular values are exactly deflated. Moreover, the computed $\bar{r}$ nonzero singular values satisfy that for all $1 \leq i \leq \bar{r}$,

$$\hat{\sigma}_i = (1 + \zeta_i'')(1 + \zeta_i')(1 + \delta_i^{(2)})(1 + \delta_i^{(1)})(1 + \delta_i)\sigma_i = (1 + \eta_i)\sigma_i, \ |\eta_i| \leq \frac{O(2C\mu)}{1 - O(2C\mu)},$$

which means that all the nonzero singular values are computed to high relative accuracy. The result is proved. □

Finally, we conclude this section with accurately computing SVDs of arbitrary submatrices extracted from a product of structured matrices with repeated nodes, e.g., Cauchy, Vandermonde, Cauchy-Vandermonde, Bernstein-Vandermonde matrices and so on, irrespective of rank deficiency and ill-conditioning.

- First, depict these structured matrices as bidiagonal product forms, as shown in Section 2.1;
- Then, express arbitrary submatrices of products of these structured matrices into bidiagonal product forms, as shown in Section 2.2;
- Finally, compute the SVDs of submatrices by applying Algorithm 3 to the derived bidiagonal product forms.

Pleasantly, all the singular values including zero ones of arbitrary submatrices are computed to high relative accuracy, as demonstrated by numerical experiments later.

5 Numerical experiments

In this section, numerical experiments are performed to confirm the claimed high relative accuracy of our proposed method. We test our method on arbitrary submatrices extracted from the products of these structured matrices with repeated nodes:

- the Cauchy matrix $A = (1/(x_i + y_j)) \in \mathbb{R}^{ns_1 \times ms_2}$ with ascending-ordered nodes

$$\begin{cases} x_{(i-1)s_1+1} = \ldots = x_{is_1} = x_i^{(c)}, \ \forall\ 1 \le i \le n; \\ y_{(j-1)s_2+1} = \ldots = y_{js_2} = y_j^{(c)}, \ \forall\ 1 \le j \le m; \end{cases}$$

- the Vandermonde matrix $A = (x_i^{\lceil j/s_2 \rceil - 1}) \in \mathbb{R}^{ns_1 \times ms_2}$ with ascending-ordered nodes

$$x_{(i-1)s_1+1} = \ldots = x_{is_1} = x_i^{(v)}, \ \forall\ 1 \le i \le n;$$

- the Cauchy-Vandermonde matrix

$$A = (a_{ij}) \in \mathbb{R}^{ns_1 \times ms_2}, \text{ where } a_{ij} = \begin{cases} 1/(x_i + y_j), \ \forall\ 1 \le j \le ls_2, \\ x_i^{\lceil (j - ls_2)/s_2 \rceil - 1} \ \forall\ ls_2 + 1 \le j \le ms_2, \end{cases}$$

  with ascending-ordered nodes

$$\begin{cases} x_{(i-1)s_1+1} = \ldots = x_{is_1} = x_i^{(cv)}, \ \forall\ 1 \le i \le n; \\ y_{(j-1)s_2+1} = \ldots = y_{js_2} = y_j^{(cv)}, \ \forall\ 1 \le j \le l; \end{cases}$$

- the Bernstein-Vandermonde matrix

$$A = (a_{ij}) \in \mathbb{R}^{ns_1 \times ms_2}, \text{ where } a_{ij} = \binom{m-1}{\lceil j/s_2 \rceil - 1} (1 - x_i)^{m - \lceil j/s_2 \rceil} x_i^{\lceil j/s_2 \rceil - 1},$$

  with ascending-ordered nodes

$$x_{(i-1)s_1+1} = \ldots = x_{is_1} = x_i^{(bv)}, \ \forall\ 1 \le i \le n.$$

Using the method described in Section 2, we first obtain the *bidiagonal product* forms of the above structured matrices as well as their products, and then proceed with Algorithm 3 to compute the SVDs. The relative errors of the computed singular values $\hat{\sigma}_i$ are measured as

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

where the "exact" σ_i are produced by the function `SingularValueList` implemented in Mathematica with a (very expensive) 200-decimal digits arithmetic and rounded back to 16 decimal digits.

We compare the error results of our method with those of the Matlab command `svd` applied to the explicit product. We need to highlight the differences in implementations between our method and the command `svd`. Our approach works on the defining nodes of these structured matrices, while the command `svd` is applied to the product explicitly computed from these structured matrices. However, it should be noted that once the product is explicitly computed, there is no guarantee of relative accuracy for the smaller singular values even with exact computation, because even tiny errors in the computation of the matrix entries can lead to significant errors in the computed singular values, especially for the smaller ones. Our main goal in comparing the relative errors is simply to demonstrate that by carefully leveraging the structural properties, the SVDs of arbitrary submatrices of products of structured matrices can be computed with high relative accuracy, despite of ill conditioning and rank deficiency. All tests are conducted in MATLAB 7.0 using double precision arithmetic.

Example 1. *Consider the product $A = A_4A_3A_2A_1$, where $A_1 \in \mathbb{R}^{n_2s_2\times n_1s_1}$, $A_2 \in \mathbb{R}^{n_3s_3\times n_2s_2}$, $A_3 \in \mathbb{R}^{n_4s_4\times n_3s_3}$ and $A_4 \in \mathbb{R}^{n_5s_5\times n_4s_4}$ are Cauchy-Vandermonde, Bernstein-Vandermonde, Vandermonde and Cauchy matrices, respectively; whose nodes, denoted with superscript 'cv', 'bv', 'v' and 'c', respectively,*

$$\begin{cases} x_i^{(cv)} = i/n_2,\ 1 \le i \le n_2,\ y_j^{(cv)} = j/n_1,\ 1 \le j \le l; \\ x_i^{(bv)} = 1/(n_3 - i + 2)\ 1 \le i \le n_3; \\ x_i^{(v)} = 1/(n_4 - i + 1),\ 1 \le i \le n_4; \\ x_i^{(c)} = 1/(n_5 - i + 1),\ 1 \le i \le n_5;\ y_j^{(c)} = (j+1)/n_4,\ 1 \le j \le n_4. \end{cases}$$

Set $n_1 = 80$, $n_2 = 70$, $n_3 = 70$, $n_4 = 50$, $n_5 = 60$, $l = 10$ and $s_1 = 2$, $s_2 = 3$, $s_3 = 2$, $s_4 = 4$, $s_5 = 3$. We compute singular values of the submatrix $A' \in \mathbb{R}^{60\times 80}$ of A having the $(3(i-1)+1)$th rows and $(2(j-1)+2)$th columns for all $1 \le i \le 60$ and $1 \le j \le 80$. As expected, our method exactly identifies 10 zero singular values, and computes all the nonzero singular values to high relative accuracy. In contrast, the command `svd` accurately computes only the largest singular values, failing to capture the 10 zero singular values. The relative errors of the computed nonzero singular values are reported in Table 1.

Example 2. *Consider the product $A = A_1A_1^TA_1$, where $A_1 \in \mathbb{R}^{ns_1\times ms_2}$ is a Cauchy-Vandermonde matrix, whose nodes*

$$x_i^{(cv)} = i/2^{n-i+1},\ 1 \le i \le n;\ y_j^{(cv)} = j^2/2^{m-j+1},\ 1 \le j \le l.$$

Table 1 The relative errors of the computed singular values in Example 1.

i	$\sigma_i(A')$	Rel. error($\hat{\sigma}_i$) by Alg. 3	Rel. error($\hat{\sigma}_i$) by svd
1	1.574847649014136e+006	8.8706e-016	1.1827e-015
2	2.488891921155727e+003	3.8369e-015	1.7906e-014
3	1.530250850738087e-002	1.5871e-015	1.1686e-009
4	4.604560570301533e-007	2.7593e-015	1.1361e-005
5	1.230495724907674e-011	1.9694e-015	5.7054e+001
6	2.045994759189220e-016	4.5786e-015	2.5896e+006
7	2.674125770626443e-021	2.2507e-015	1.5249e+011
8	3.280221392098057e-026	3.3246e-015	1.1460e+016
9	2.792331556079233e-031	9.4095e-016	1.1193e+021
10	3.272383447034477e-036	8.1676e-016	8.6262e+025
11	2.500530956218926e-041	1.0194e-015	9.3835e+030
12	9.537769421320553e-046	1.3049e-015	4.1954e+034
13	4.707522299934931e-050	1.0086e-015	8.1688e+037
14	2.903452644383030e-055	1.2476e-016	4.9917e+041
15	1.673616756730713e-060	4.2933e-015	1.4743e+046
16	1.159846521080633e-065	4.5447e-015	7.4582e+050
17	7.568811773397085e-071	2.7629e-015	5.0092e+055
18	2.547381715070572e-075	3.4688e-015	6.4984e+059
19	3.875607399766417e-080	4.6387e-015	2.1535e+064
20	2.150255912305010e-085	2.9236e-015	2.0886e+069
21	9.419854429953473e-091	7.1745e-015	2.7971e+074
22	1.002382461091948e-095	2.6549e-015	1.0141e+079
23	1.988962611556771e-100	3.0624e-015	2.5014e+083
24	1.251313212016393e-105	1.8569e-015	3.2263e+088
25	7.041738853487294e-111	2.7273e-015	3.0746e+093
26	7.548781771221338e-116	1.4931e-016	1.4836e+098
27	7.236595626968585e-121	8.3177e-016	7.6894e+102
28	3.515150759856632e-126	2.4262e-015	1.2137e+108
29	2.224064308428203e-131	1.1252e-016	1.0440e+113
30	1.722231254363735e-136	2.2173e-015	7.3514e+117
31	9.649619609976743e-142	5.1326e-015	6.3227e+122
32	5.328970371601404e-147	1.8770e-015	1.0225e+128
33	3.012241081997066e-152	4.9261e-015	9.0698e+132
34	1.483092890644080e-157	4.3619e-016	1.1362e+138
35	6.715748200773162e-163	3.3071e-015	1.7681e+143
36	2.762617922541912e-168	5.1113e-016	2.2882e+148
37	9.830232713035425e-174	3.6531e-016	4.7691e+153
38	2.950824109659177e-179	3.4818e-016	9.3520e+158
39	7.215403029682406e-185	2.3538e-015	2.3604e+164
40	1.395795579034603e-190	5.3557e-016	8.9346e+169
41	2.087830631603745e-196	2.8455e-015	4.4184e+175
42	2.360667297832382e-202	9.2162e-015	1.5742e+181
43	1.970286285886377e-208	5.0459e-015	1.8417e+187
44	1.182321822224019e-214	5.4043e-015	1.7817e+193
45	4.946908994775014e-221	1.5894e-015	3.5330e+199
46	1.389686685088435e-227	7.4192e-015	6.0298e+205
47	2.492871628238566e-234	4.8184e-015	3.2611e+212
48	2.652362330479972e-241	6.7796e-015	1.2274e+219
49	1.473958086168526e-248	8.6182e-015	1.6109e+226
50	3.177462705198649e-256	6.8880e-015	5.0078e+233

Set $n = 50$, $m = 50$, $l = 15$ and $s_1 = 3$, $s_2 = 2$. We compute the singular values of the submatrix $A' \in \mathbb{R}^{50\times 60}$ of A having the $(3(i-1)+2)$th $(1 \le i \le 50)$ rows and the columns from 21 to 80. Once more, our method computes all the nonzero singular values to high relative accuracy, exactly returning 20 zero singular values. Conversely, the command `svd` *is unable to accurately compute the smaller singular values, including the 20 zero singular values, with the correct number of significant digits. The relative errors of the computed nonzero singular values are reported in Table 2.*

Example 3. *Consider the product $A = A_1 A_1^T A_1$, where $A_1 \in \mathbb{R}^{ns_1 \times ms_2}$ is a Vandermonde matrix, whose nodes*

$$x_i^{(v)} = (i+1)/(n^2 - 2i + 1),\ 1 \le i \le n.$$

Set $n = 50$, $m = 50$ and $s_1 = 2$, $s_2 = 3$. We compute singular values of the submatrix $A' \in \mathbb{R}^{70\times 50}$ of A having the rows from 11 to 80 and the $(3(i-1)+2)$th $(1 \le i \le 50)$ columns. A distinctive advantage of our method lies in its capability to exactly identify

Table 2 The relative errors of the computed singular values in Example 2.

i	$\sigma_i(A')$	Rel. error($\hat{\sigma}_i$) by Alg. 3	Rel. error($\hat{\sigma}_i$) by svd
1	3.468887277416921e+129	1.4198e-015	0
2	8.018730362382941e+096	2.0820e-014	5.9935e+004
3	4.573629685918318e+066	1.5216e-014	7.6397e+025
4	5.359733112254777e+040	0	2.5886e+038
5	3.850034487231992e+038	1.9625e-016	1.0364e+028
6	1.219745346737479e+035	0	2.4819e+026
7	1.042762368688207e+032	1.7276e-016	8.2678e+022
8	1.930454825605820e+029	1.8226e-016	1.3024e+020
9	1.125637514239660e+027	3.6630e-015	1.2180e+018
10	1.179589020167612e+018	2.8213e-015	5.9881e+026
11	1.323455242467037e+013	7.0837e-015	4.3037e+025
12	1.162171058127293e+010	2.4618e-015	4.1844e+028
13	2.648280221329972e+006	1.2308e-015	1.2481e+027
14	5.510081957368244e+000	1.9343e-015	8.3542e+031
15	6.337837039281268e-002	4.3793e-016	1.1133e+033
16	3.697851935885653e-003	1.5246e-015	2.9211e+031
17	3.061156104412551e-006	8.5778e-015	8.6077e+033
18	2.787053739109288e-009	6.9747e-015	6.0813e+035
19	7.170489313678953e-012	5.2948e-015	1.5246e+036
20	7.583241112107084e-017	3.2508e-016	3.8804e+037
21	3.272553868178148e-020	6.6207e-015	7.9559e+039
22	4.510413184796948e-025	3.0541e-015	2.4478e+043
23	5.059703390471064e-029	1.0635e-014	5.7334e+045
24	4.518370980853580e-035	1.1831e-015	6.0170e+049
25	1.018365532436303e-039	3.5242e-015	2.5675e+053
26	1.077544585350837e-046	8.4823e-015	3.6633e+059
27	8.872108292133314e-051	3.4784e-015	1.5567e+063
28	1.966010143181168e-059	2.0242e-015	1.9959e+070
29	2.394834026628231e-064	1.1269e-015	3.2862e+074
30	1.099622669639911e-075	7.8126e-016	4.1482e+084

the 15 zero singular values, in stark contrast to the command `svd` *that fails to report any zeros. The relative errors of the computed nonzero singular values are reported in Table 3, highlighting the high relative accuracy of our method.*

Example 4. *Consider the product $A = A_1 A_1^T A_1$, where $A_1 =: \mathcal{BD}(A_1) = (\bar{g}_{ij}, g_{ij}) \in \mathbb{R}^{90\times 50}$, here all $g_{ij} = r_i/j$ with r_i are randomly chosen by the Matlab command* `rand`*, and all $\bar{g}_{ij}$ are randomly chosen by the Matlab command* `randint`*. The rank of A is unknown. The explicit matrix A_1 is formed in a subtraction-free manner, ensuring a high level of relative accuracy. Since $\sigma_i(A) = (\sigma_i(A_1))^3$ for all i, the command* `svd` *computes singular values of A by $(\texttt{svd}(A_1))^3$. As observed, Alg. 3 computes all the 14 nonzero singular values to high relative accuracy, and all the 36 zero singular values are exactly returned. However, $(\texttt{svd}(\texttt{A}_1))^3$ only identifies 1 zero singular values and exhibits a loss of relative accuracy for some small non-zero singular values. The relative errors of the computed nonzero singular values are reported in Table 4.*

Appendix A The proofs of the passing-through operations (10) and (12)

The proof of the operation (10): Since both $U_{1:n}L_{1:n}$ and $\bar{L}_{1:n}\bar{U}_{1:n}$ are tridiagonal, comparing the entries on the diagonal, subdiagonal and superdiagonal gives

$$\begin{cases} \bar{y}_i'\bar{x}_i' + x_{i-1}'y_{i-1}' = \bar{y}_i\bar{x}_i + x_iy_i, \\ \bar{y}_i'x_i' = \bar{y}_{i+1}x_i, \;\; \bar{x}_i'y_i' = \bar{x}_{i+1}y_i, \end{cases} \qquad \forall\ i = 1, \dots, n,$$

Table 3 The relative errors of the computed singular values in Example 3.

i	$\sigma_i(A')$	Rel. error($\hat{\sigma}_i$) by Alg. 3	Rel. error($\hat{\sigma}_i$) by svd
1	2.510397022449398e+003	3.6229e-016	1.8115e-016
2	3.816985499740374e-004	1.4202e-016	2.8474e-012
3	4.143858773645414e-011	1.2476e-015	3.4959e-006
4	4.242521375454278e-018	1.0895e-015	8.1961e-001
5	4.239056395592293e-025	2.3831e-015	1.9518e+005
6	4.168659664179253e-032	2.3636e-015	3.8671e+010
7	4.043646122116740e-039	2.7433e-015	2.6793e+015
8	3.869430900719566e-046	2.6134e-015	5.7379e+020
9	3.649829639076117e-053	2.0325e-015	7.9162e+025
10	3.389249221406257e-060	5.2185e-015	1.5208e+031
11	3.093670547436131e-067	4.2596e-016	2.6090e+036
12	2.770991249357053e-074	6.9447e-015	7.7554e+041
13	2.430919688798274e-081	5.9702e-015	1.3220e+047
14	2.084507385475087e-088	2.2758e-015	3.1006e+052
15	1.743397226830317e-095	9.5404e-016	5.7334e+057
16	1.418889627711162e-102	5.7294e-015	8.9102e+062
17	1.120959850392690e-109	2.9520e-015	2.3207e+068
18	8.573784180593364e-117	3.7794e-015	6.4604e+073
19	6.330803127549630e-124	1.2336e-014	1.7316e+079
20	4.498919337915358e-131	6.5640e-015	5.5462e+084
21	3.066611613512485e-138	5.0588e-015	1.2460e+090
22	1.997581004819857e-145	4.6289e-015	3.5820e+095
23	1.238413047998368e-152	1.3522e-014	1.2220e+101
24	7.273625411986728e-160	1.7371e-015	3.6959e+106
25	4.026307600418828e-167	4.8632e-015	1.5156e+112
26	2.088084609674866e-174	3.2246e-015	3.5216e+117
27	1.007534076170427e-181	5.5766e-015	1.5141e+123
28	4.485972259301394e-189	5.3325e-016	9.1815e+128
29	1.824599254309977e-196	1.1331e-014	4.4799e+134
30	6.694074753934959e-204	2.5391e-015	2.6863e+140
31	2.178711168080216e-211	3.8750e-015	1.3847e+146
32	6.146438129885899e-219	6.3450e-015	7.7022e+151
33	1.450862706608087e-226	4.9099e-015	7.6456e+157
34	2.692622897706825e-234	1.5769e-014	9.1431e+163
35	3.389889240846505e-242	2.7014e-015	6.3336e+169

Table 4 The relative errors of the computed singular values in Example 4.

i	$\sigma_i(A')$	Rel. error($\hat{\sigma}_i$) by Alg. 3	Rel. error($\hat{\sigma}_i$) by svd
1	8.332062400159658e+171	1.0299e-015	3.0896e-015
2	2.564548603621145e+089	1.1023e-015	9.2338e+033
3	1.099405131386926e+038	3.4363e-016	1.9317e+080
4	2.878741420489996e+006	1.6176e-016	1.2702e+109
5	7.418482246460084e-005	1.0961e-015	3.1706e+115
6	2.915960871453962e-014	6.4928e-016	5.0791e+123
7	9.291825752068523e-016	4.4572e-015	1.1561e+121
8	3.041046489601403e-022	1.5462e-015	2.8172e+120
9	3.921877379377079e-035	4.0890e-015	4.2790e+126
10	9.631567808072229e-052	3.0809e-016	9.4277e+140
11	1.612410893623431e-055	6.8518e-015	2.0076e+139
12	3.747225733496450e-059	1.8880e-015	1.7617e+131
13	6.896108910152905e-078	4.1711e-016	8.2239e+149
14	6.455473653011518e-121	4.5289e-015	2.8651e+186

with the convention $x'_0 = y'_0 = 0$. To avoid subtraction, we introduce the intermediate quantities

$$z_i = \bar{y}_i \bar{x}_i - x'_{i-1} y'_{i-1}, \;\; 1 \le i \le n$$

and let

$$w_i := \bar{y}'_i \bar{x}'_i = z_i + x_i y_i.$$

Observe that $z_1 = \bar{y}_1 \bar{x}_1 \ge 0$. Assume that $z_i \ge 0$ and thus $w_i \ge 0$. Then,

- if $w_i > 0$, then we set

$$\bar{x}'_i = 1, \;\; \bar{y}'_i = w_i > 0, \;\; x'_i = \frac{\bar{y}_{i+1} x_i}{\bar{y}'_i} \ge 0, \;\; y'_i = \frac{y_i \bar{x}_{i+1}}{\bar{x}'_i} \ge 0,$$

thus,

$$z_{i+1} = \bar{y}_{i+1}\bar{x}_{i+1} - x_i'y_i' = \frac{\bar{y}_{i+1}\bar{x}_{i+1}(\bar{y}_i'\bar{x}_i' - x_iy_i)}{\bar{y}_i'\bar{x}_i'} = \frac{\bar{y}_{i+1}\bar{x}_{i+1}z_i}{\bar{x}_i'\bar{y}_i'} \geq 0;$$

- if $w_i = 0$, then $x_iy_i = 0$. If $y_i = 0$, then we set

$$\bar{x}_i' = w_i = 0,\ \bar{y}_i' = 1,\ x_i' = \bar{y}_{i+1}x_i \geq 0,\ y_i' = 0,$$

thus,

$$z_{i+1} = \bar{y}_{i+1}\bar{x}_{i+1} - x_i'y_i' = \bar{y}_{i+1}\bar{x}_{i+1} \geq 0;$$

and if $y_i \neq 0$, then $x_i = 0$, we set

$$\bar{x}_i' = 1,\ \bar{y}_i' = w_i = 0,\ x_i' = 0,\ y_i' = y_i\bar{x}_{i+1} \geq 0,$$

thus,

$$z_{i+1} = \bar{y}_{i+1}\bar{x}_{i+1} - x_i'y_i' = \bar{y}_{i+1}\bar{x}_{i+1} \geq 0.$$

Inductively we have shown that all the involved quantities are nonnegative, which costs at most $8n$ subtraction-free operations. Thus, (10) is proved. □

The proof of the operation (12): By comparing the $(i, i+1)$th, $(i+1, i+1)$th and $(i, i+2)$th entries on both sides of $U_{1:m}U_{1:m}' = \bar{U}_{1:m}'\bar{U}_{2:m}$, we have that $\bar{x}_1' = \bar{y}_1\bar{x}_1$, and

$$\begin{cases} y_i'\bar{x}_i' + \bar{y}_{i+1}'x_i' = x_i\bar{y}_i + \bar{x}_{i+1}y_i, \\ \bar{x}_{i+1}'\bar{y}_{i+1}' = \bar{x}_{i+1}\bar{y}_{i+1},\ y_{i+1}'x_i' = x_{i+1}y_i, \end{cases} \quad \forall\ i = 1, \ldots, m-1,$$

with the convention $y_1' = 0$. To avoid subtraction, we introduce the intermediate quantities

$$z_i = x_i\bar{y}_i - y_i'\bar{x}_i',\ 1 \leq i \leq m-1,$$

and let

$$w_i := \bar{y}_{i+1}'x_i' = z_i + \bar{x}_{i+1}y_i.$$

Observe that $z_1 = x_1\bar{y}_1 \geq 0$. Assume that $z_i \geq 0$ and thus $w_i \geq 0$. Then,

- if $w_i > 0$, then we set

$$x_i' = w_i > 0,\ \bar{y}_{i+1}' = 1,\ \bar{x}_{i+1}' = \frac{\bar{x}_{i+1}\bar{y}_{i+1}}{\bar{y}_{i+1}'} \geq 0,\ y_{i+1}' = \frac{x_{i+1}y_i}{x_i'} \geq 0,$$

thus,

$$z_{i+1} = x_{i+1}\bar{y}_{i+1} - y_{i+1}'\bar{x}_{i+1}' = \frac{x_{i+1}\bar{y}_{i+1}(\bar{y}_{i+1}'x_i' - \bar{x}_{i+1}y_i)}{\bar{y}_{i+1}'x_i'} = \frac{x_{i+1}\bar{y}_{i+1}z_i}{\bar{y}_{i+1}'x_i'} \geq 0;$$

- if $w_i = 0$, then $\bar{x}_{i+1}y_i = 0$. If $y_i = 0$, then we set

$$x_i' = w_i = 0,\ \bar{y}_{i+1}' = 1,\ \bar{x}_{i+1}' = \bar{x}_{i+1}\bar{y}_{i+1} \geq 0,\ y_{i+1}' = 0,$$

thus,

$$z_{i+1} = x_{i+1}\bar{y}_{i+1} - y'_{i+1}\bar{x}'_{i+1} = x_{i+1}\bar{y}_{i+1} \geq 0;$$

and if $y_i \neq 0$, then $\bar{x}_{i+1} = 0$, then we set

$$x'_i = 1,\ \bar{y}'_{i+1} = w_i = 0,\ \bar{x}'_{i+1} = 0,\ y'_{i+1} = x_{i+1}y_i \geq 0,$$

thus,

$$z_{i+1} = x_{i+1}\bar{y}_{i+1} - y'_{i+1}\bar{x}'_{i+1} = x_{i+1}\bar{y}_{i+1} \geq 0.$$

Inductively we have shown that all the involved quantities are nonnegative, which costs at most $8m$ subtraction-free operations. Thus, (12) is proved. □

Funding The work of Rong Huang is supported by the National Natural Science Foundation of China (Grant No. 12271153). The work of Jungong Xue is supported by the National Science Foundation of China (Grant No. 11771100) and the Laboratory of Mathematics for Nonlinear Science, Fudan University.

Data Availability The datasets generated during the current study are available from the corresponding authors on reasonable request.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

[1] L. Arnold, H. Crauel and J.-P. Eckmann, *Lyapunov exponents*, Springer-Verlag, Berlin, Ner york, 1991. Lecture Notes in Mathematics #1486.

[2] Z. Bai and J.W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl., **14**, 1464-1486 (1993).

[3] M.V. Barel, Y. Vanberghen and P.V. Dooren, *Using semiseparable matrices to compute the SVD of a general product/quotient*, J. Comput. Appl. Math., **234**, 3175-3180 (2010).

[4] A.W. Bojanczyk, L.M. Ewerbring, F.T. Luk and P.V. Dooren, *An accurate product SVD algorithm*, Signal Processing, **25**, 189-201 (1991).

[5] D. Chu, L.D. Lathauwer and B.D. Moor, *On the computation of the restricted singular value decomposition via the cosine-sine decomposition*, SIAM J. Matrix Anal. Appl., **22**, 580-601 (2000).

[6] D. Chu, L.D. Lathauwer and B.D. Moor, *A QR-type reduction for computing the SVD of a general matrix product/quotient*, Numer. Math., **95**, 101-121 (2003).

[7] J. Delgado, P. Koev, A. Marco, J-J. Martínez, J.M. Peña, P-O. Persson, S. Spasov, *Bidiagonal decompositions of Vandermonde-type matrices of arbitrary rank*, J. Comput. Appl. Math., **426**, 115064 (2023).

[8] J. Demmel, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl., **21**, 562-580 (1999).

[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**, 21-80 (1999).

[10] J. Demmel, D. Ioana, O. Holtz and P. Koev, *Accurate and efficient expression evaluation and linear algebra*, Acta Numer., **17**, 87-145 (2008).

[11] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci Stat. Comp., **11**, 873-912 (1990).

[12] F.M. Dopico, P. Koev and J.M. Molera, *Implicit standard Jacobi gives high relative accuracy*, Numer. Math., **113**, 519-553 (2009).

[13] F.M. Dopico, J.M. Molera and J. Moro, *An orthogonal high relative accuracy algorithm for the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., **25**, 301-351 (2003).

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

[15] K. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., **67**, 191-229 (1994).

[16] G.H. Golub, K. Sølna and P.V. Dooren, *Computing the SVD of a general matrix product/quotient*, SIAM J. Matrix Anal. Appl., **22**, 1-19 (2000).

[17] 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).

[18] R. Henningsson and M. Fontes, *SMSSVD: SubMatrix Selection Singular Value Decomposition*, Bioinformatics, **35**, 478-486 (2019).

[19] 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**, 165-179 (2003).

[20] R. Huang, *A periodic qd-type reduction for computing eigenvalues of structured matrix products to high relative accuracy*, J. Sci. Comput., **75**, 1129-1261 (2018).

[21] R. Huang, *A qd-type method for computing generalized singular values of BF*

matrix pairs with sign regularity to high relative accuracy, Math. Comp., **89**, 229-252 (2020).

[22] R. Huang, *Accurate computation of generalized eigenvalues of regular SR-BP pairs*, SIAM J. Matrix Anal. Appl., **42**, 1680-1702 (2021).

[23] P. Koev, *Accurate eigenvalues and SVDs of totally nonnegative matrices*, SIAM J. Matrix Anal. Appl., **27**, 1-23 (2005).

[24] P. Koev, *Accurate computations with totally nonnegative matrices*, SIAM J. Matrix Anal. Appl., **29**, 731-751 (2007).

[25] P. Koev, *Accurate eigenvalues and exact zero Jordan blocks of totally nonnegative matrices*, Numer. Math., **141**, 693-713 (2019).

[26] P. Koev, http://math.mit.edu/ plamen/software/TNTool.html

[27] A. Marco, J-J. Martínez, *Polynomial least squares fitting in the Bernstein basis*, Linear Algebra Appl., **433**, 1254-1264 (2010).

[28] A. Marco, J-J. Martínez, *Accurate computations with Said-Ball-Vandemonde matrices*, Linear Algebra Appl., 432, 2894-2908 (2010).

[29] A. Marco and J.J. Martínez, *A fast and accurate algorithm for solving Bernstein-Vandermonde linear systems*, Linear Algebra Appl., **422**, 616-628 (2007).

[30] J.J. Martínez and J.M. Peña, *Fast algorithms of Björck-Pereyra type for solving Cauchy-Vandermonde linear systems*, Appl. Numer. Math., **26**, 343-352 (1998).

[31] C.B. Moler and G.W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., **10**, 241-256 (1973).

[32] B. Parlett and O.A. Marques, *An implementation of the dqds algorithm (positive case)*, Linear Algebra Appl., **309**, 217-259 (2000).

[33] D. Stewart, *A new algorithm for the SVD of a long product of matrices and the stability of products*, Electron. Trans. Numer. Anal., **5**, 29-47 (1997).

[34] Q. Ye, *Computing singular values of diagonally dominant matrices to high relative accuracy*, Math. Comp., **77**, 2195-2230 (2008).