

A Periodic qd-Type Reduction for Computing Eigenvalues of Structured Matrix Products to High Relative Accuracy

Rong Huang¹

Received: 26 May 2017 / Revised: 28 September 2017 / Accepted: 12 October 2017 /
Published online: 23 October 2017
© Springer Science+Business Media, LLC 2017

Abstract In this paper, we consider the product eigenvalue problem for a wide class of structured matrices containing the well-known Vandermonde and Cauchy matrices. A periodic qd-type reduction method is developed for computing eigenvalues of products of these rectangular matrices so that no subtraction of like-signed numbers occurs. Consequently, all the eigenvalues of such a product are computed to high relative accuracy in a preferable complexity. Error analysis and numerical experiments are provided to confirm the high relative accuracy.

Keywords Structured matrices · Product eigenvalue problems · qd-Type algorithms · High relative accuracy

Mathematics Subject Classification 65F15 · 15A18

1 Introduction

The study of high-accuracy computations is an active research topic of great interest in recent years. For the eigenvalue problem of a single matrix, high-accuracy algorithms have been constructed for a few classes of matrices such as diagonally dominant matrices [1, 2], tridiagonal matrices [9], acyclic matrices [6], certain sign regular matrices [17, 19, 20], matrices with rank-revealing decompositions [11, 12] and structured matrices [5, 7, 8, 10, 22]. However, there are many situations in which one need to find eigenvalues of a product of

Research supported by the National Natural Science Foundation of China (Grant No. 11471279), the Natural Science Foundation for Distinguished Young Scholars of Hunan Province (Grant No. 2017JJ1025) and the Research Foundation of Education Bureau of Hunan Province (Grant No. 14B178).

✉ Rong Huang
rongh98@aliyun.com

¹ School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, Hunan, China

two or more matrices. Indeed, the product eigenvalue problem arises in many applications and has been extensively studied, see [29,30] and the references therein for an overview. Since accuracy is always lost when explicitly forming the product, a variety of algorithms such as LR-type algorithms [18,28], Jacobi-type algorithms [15] and QR-type algorithms [4,14,21] have been developed for a product of two or more matrices by implicitly working with the factors. Our interest of this paper is the high-accuracy computation for the product eigenvalue problem of large numbers of structured matrices.

Structured matrices such as Vandermonde and Cauchy matrices frequently appear in various areas of modern computing [23,26,27]. In this paper, we are concerned with a wide class of structured matrices containing Vandermonde and Cauchy matrices.

Definition 1 [16] Let $A \in \mathbb{R}^{n \times m}$. If

$$\begin{cases} \text{rank} A[s+1:i+1|1:j] \leq \text{rank} A[s:i|1:j], & \forall i \geq j, i-j+1 \leq s \leq i, \\ \text{rank} A[1:i|s+1:j+1] \leq \text{rank} A[1:i|s:j], & \forall j \geq i, j-i+1 \leq s \leq j, \end{cases}$$

then A is called a *consecutive-rank-descending* (CRD) matrix. Here, denote by $A[i:j|k:l]$ the submatrix of $A \in \mathbb{R}^{n \times m}$ having row and column indexes in the ranges i through j and k through l , respectively.

We will show that Vandermonde and Cauchy matrices are special CRD matrices in Section 2 later. Recall that single Vandermonde and Cauchy matrices are badly ill-conditioned [24]. So, it is an interesting challenge to accurately find eigenvalues of products of these badly ill-conditioned matrices. When dealing with the product eigenvalue problem, the currently used algorithms are QR-type algorithms. However, it is known that QR-type algorithms destroy matrix structures. Thus, we turn to an alternative approach, i.e., LR-type algorithms. One advantage of LR-type algorithms over QR-type algorithms is that they can preserve matrix structures, but the disadvantage is the possible instability in floating point arithmetic. In fact, the works by Parlett and Fernando [13,25] have brought a further for LR-type algorithms. Indeed, its variants such as qd-type algorithms have been developed to successfully achieve the desirable high relative accuracy for the eigenvalue problem of single matrices [3,13,25]. Therefore, our main contribution of this paper is to compute all the eigenvalues of products of rectangular CRD matrices with high relative accuracy by developing a periodic qd-type reduction.

The rest of the paper is organized as follows. In Sect. 2, we provide the unique representation for the class of CRD matrices, and we then verify that the well-known Vandermonde and Cauchy matrices are special CRD matrices. In Sect. 3, we provide a qd-type algorithm for updating the representation of the product of two specific CRD matrices under the assumption that no breakdown occurs. In Sect. 4, the periodic qd-type method is developed to efficiently reduce a product of rectangular CRD matrices into the tridiagonal form by implicitly working with its factors. In Sect. 5, we identify the subset of these products for which no subtraction of like-signed numbers occurs throughout the reduction process. Consequently, all the eigenvalues of such a product are computed to high relative accuracy in a preferable complexity. Error analysis is provided to illustrate the high relative accuracy. In Sect. 6, numerical experiments are presented to confirm the high relative accuracy.

2 The Unique Representation

In our work [16], the unique representation for (p, q) -diagonal consecutive-rank-descending matrices has been derived, and CRD matrices are just special $(0, 0)$ -diagonal consecutive-

Then in the absence of breakdown,

$$U_r^{-1}C = \bar{C}\bar{U}_{r+1}^{-1} =: \begin{bmatrix} 1 & c_2 & & & \\ & \ddots & \ddots & & \\ & & 1 & c_{r-1} & \\ & & & 1 & \bar{c}_r \\ & & & & \ddots & \ddots \\ & & & & & 1 & \bar{c}_m \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & & & \\ & \ddots & \ddots & & \\ & & 1 & 0 & \\ & & & 1 & \bar{u}_{r+1} \\ & & & & \ddots & \ddots \\ & & & & & 1 & \bar{u}_m \\ & & & & & & 1 \end{bmatrix}^{-1}, \tag{10}$$

where with the convention $u_{r-1} = 0$,

$$\begin{cases} z_j = c_j - u_j, \\ \bar{u}_j = 0, \bar{c}_j = z_j, & \text{if } u_{j-1} = 0, \\ t_j = \frac{z_j}{z_{j-1}}, \bar{u}_j = t_j u_{j-1}, \bar{c}_j = t_j c_{j-1}, & \text{if } u_{j-1} \neq 0, \end{cases} \quad \forall r \leq j \leq m. \tag{11}$$

Moreover, the transformation (10) is subtraction-free if and only if $\text{sign}(c_j u_j) = -1$ or 0 for all $r \leq j \leq m$. Consequently, in this case no breakdown occurs for the transformation (10).

Proof Comparing the entries in both sides of $C\bar{U}_{r+1} = U_r\bar{C}$, we have that

$$\begin{cases} c_r = u_r + \bar{c}_r, \\ c_{j-1}\bar{u}_j = u_{j-1}\bar{c}_j, \bar{u}_j + c_j = u_j + \bar{c}_j, \end{cases} \quad \forall r + 1 \leq j \leq m,$$

from which the formula (11) is derived under the assumption that there is no breakdown. Moreover, for all $r \leq j \leq m$, $z_j = c_j - u_j$ is subtraction-free if and only if $\text{sign}(c_j u_j) = -1$ or 0 . Consequently, in this case no breakdown occurs because $|z_{j-1}| \geq |u_{j-1}| > 0$ for the case $u_{j-1} \neq 0$ ($r \leq j \leq m$). \square

Our main result of this section is the following:

Theorem 2 Let $A \in \mathbb{R}^{n \times m}$ be CRD of full rank, and let $U_r \in \mathbb{R}^{n \times n}$ ($2 \leq r \leq n$) be as in (6). Then in the absence of breakdown, $A' = U_r^{-1}A$ is CRD of full rank.

Proof Let $A = B_1 B_2 \dots B_{n-1} D C_{m-1} \dots C_2 C_1$ be as in (1), and let $P = (p_{ij}) \in \mathbb{R}^{n \times m}$ be its parameter matrix satisfying the fact (2). In the absence of breakdown, we recursively get by using the transformation (7) that

$$\begin{aligned} A' &= \underline{U_r^{-1} B_1} \dots B_{n-1} D C_{m-1} \dots C_1 = B'_1 \underline{U_r^{-1} B_2} \dots B_{n-1} D C_{m-1} \dots C_1 \\ &= \dots = B'_1 \dots B'_{n-1} \bar{U}_r^{-1} D C_{m-1} \dots C_1, \end{aligned} \tag{12}$$

where each B'_k ($1 \leq k \leq n-1$) has the same form as that of B_k , whose corresponding entries $p'_{i,i-n+k} = 0$ if and only if $p_{i,i-n+k} = 0$ for all $n-k < i \leq n$. So,

$$p'_{ij} = 0, \quad i > j \Rightarrow p'_{sj} = 0, \quad \forall s \geq i. \tag{13}$$

Further, by the transformation (9),

$$A' = B'_1 \dots B'_{n-1} \bar{U}_r^{-1} D C_{m-1} \dots C_1 = B'_1 \dots B'_{n-1} \underline{D' \bar{U}_r^{-1} C_{m-1}} \dots C_1, \tag{14}$$

where $\bar{U}_r \in \mathbb{R}^{m \times m}$ is upper bidiagonal with unit diagonal, $D' = \text{diag}(p'_{kk}) \in \mathbb{R}^{n \times m}$ is diagonal with

$$p'_{kk} \neq 0, \quad \forall 1 \leq k \leq \min\{n, m\}. \tag{15}$$

Finally, in the absence of breakdown, we recursively get by using the transformation (10) that

$$\begin{aligned}
 A' &= B'_1 \dots B'_{n-1} D' \tilde{U}_r^{-1} C_{m-1} \dots C_1 = B'_1 \dots B'_{n-1} D' C'_{m-1} \tilde{U}_{r+1}^{-1} C_{m-2} \dots C_{r-1} C_{r-2} \dots C_1 \\
 &= \dots = B'_1 \dots B'_{n-1} D' C'_{m-1} \dots C'_{r-1} C_{r-2} \dots C_1
 \end{aligned}
 \tag{16}$$

where for all $1 \leq s \leq m - r + 1$, $\tilde{U}_{r+s-1}^{-1} C_{m-s} = C'_{m-s} \tilde{U}_{r+s}^{-1}$ (here, $\tilde{U}_{m+1}^{-1} = I_m$), C'_{m-s} is bidiagonal different form C_{m-s} only in the entries p'_{ij} for $r - 1 \leq i \leq \min\{m - 1, n\}$ and $j = i + s$, which are computed from the entries p_{ij} of C_{m-s} by the formula (11) as follows:

$$\begin{cases}
 z_j = p_{ij} - u_j, \\
 \bar{u}_j = 0, \quad p'_{ij} = z_j, \quad \text{if } u_{j-1} = 0, \\
 t_j = \frac{z_j}{z_{j-1}}, \quad p'_{ij} = t_j p_{i-1, j-1}, \quad \bar{u}_j = t_j u_{j-1}, \quad \text{if } u_{j-1} \neq 0,
 \end{cases}
 \tag{17}$$

here, denote by u_j and \bar{u}_j the $(j - 1, j)$ th entries of \tilde{U}_{r+s-1} and \tilde{U}_{r+s} , respectively. For showing the fact (2), let $p'_{ij} = 0$ of C'_{m-s} for some $r - 1 \leq i \leq \min\{m - 1, n\}$ and $j = i + s$, then it need to prove $p'_{i, j+1} = 0$ with $j + 1 \leq m$. Notice that $1 \leq s \leq m - r + 1$. If $s = m - r + 1$, then $j + 1 > m$. So, assume that $1 \leq s < m - r + 1$ and $j < m$. According to (17), there are the following cases to be considered.

- For the case $u_{j-1} = 0$, we have that $\bar{u}_j = 0$ and $0 = p'_{ij} = z_j = p_{ij} - u_j$. If $u_j \neq 0$, then by the formula (17), $t_{j+1} = \frac{z_{j+1}}{z_j} = \frac{z_{j+1}}{p_{ij}}$ is a breakdown. So, we must have that $u_j = 0$, and then, $p_{ij} = 0$. Thus,

$$p_{ij} = 0 \Rightarrow p_{i, j+1} = 0 \text{ by the fact (2); } \quad u_j = 0 \Rightarrow \bar{u}_{j+1} = 0 \text{ by the formula (17).}$$

Hence, for the next transformation $\tilde{U}_{r+s}^{-1} C_{m-s-1} = C'_{m-s-1} \tilde{U}_{r+s+1}^{-1}$, since $\bar{u}_j = 0$, we have by the formula (11) that $p'_{i, j+1} = p_{i, j+1} - \bar{u}_{j+1} = 0$.

- For the case $u_{j-1} \neq 0$, we have $0 = p'_{ij} = t_j p_{i-1, j-1}$, which means that $t_j = 0$ or $p_{i-1, j-1} = 0$.

- If $t_j = 0$, then $z_j = p_{ij} - u_j = 0$. Assume that $u_j \neq 0$. Then by the formula (17), $t_{j+1} = \frac{z_{j+1}}{z_j}$ is a breakdown. So, we must have $u_j = 0$, and then, $p_{ij} = 0$. Thus,

$$p_{ij} = 0 \Rightarrow p_{i, j+1} = 0; \quad u_j = 0 \Rightarrow \bar{u}_{j+1} = 0.$$

Hence, for the next transformation $\tilde{U}_{r+s}^{-1} C_{m-s-1} = C'_{m-s-1} \tilde{U}_{r+s+1}^{-1}$, since $\bar{u}_j = t_j u_{j-1} = 0$, we have by the formula (11) that $p'_{i, j+1} = p_{i, j+1} - \bar{u}_{j+1} = 0$.

- If $p_{i-1, j-1} = 0$ and $t_j \neq 0$, then $p_{i-1, j} = 0$ by the fact (2). Hence, for the next transformation $\tilde{U}_{r+s}^{-1} C_{m-s-1} = C'_{m-s-1} \tilde{U}_{r+s+1}^{-1}$, since $\bar{u}_j = t_j u_{j-1} \neq 0$, we have by the formula (11) that

$$p'_{i, j+1} = \bar{t}_{j+1} p_{i-1, j} = \frac{p_{i, j+1} - \bar{u}_{j+1}}{p_{i-1, j} - \bar{u}_j} p_{i-1, j} = 0.$$

Hence, we conclude that

$$p'_{ij} = 0, \quad j > i \Rightarrow p'_{is} = 0, \quad \forall s \geq j.
 \tag{18}$$

Thus, we get that A' is of the form (1) satisfying the fact (2) because of the facts (13), (15) and (18). Therefore, by Theorem 1, A' is CRD of full rank. \square

By applying a transpose transformation, we immediately get the following result.

Corollary 3 Let $A \in \mathbb{R}^{n \times m}$ be CRD of full rank, and let $L_r \in \mathbb{R}^{m \times m}$ ($2 \leq r \leq m$) be nonsingular whose transpose is as in (6). Then in the absence of breakdown, $A' = AL_r^{-1}$ is CRD of full rank.

Therefore, we have Algorithm 1 for updating the representation of the product of two specific CRD matrices.

Algorithm 1 The algorithm computes the parameter matrix $P = (p'_{ij})$ of $A' = U_r^{-1}A$ ($2 \leq r \leq n$), where $A \in \mathbb{R}^{n \times m}$ is CRD of full rank, and $U_r \in \mathbb{R}^{n \times n}$ is as in (6).

- 1: First, compute the elements p'_{ij} ($i > j$) of (12) by the formula (8).
- 2: Further, compute the elements p'_{ij} ($i = j$) of (14) by the formula of (9).
- 3: Finally, compute the elements p'_{ij} ($i < j$) of (16) by the formula (11).

Observe that the formulas of (8), (9) and (11) cost at most 6, 3 and 4 arithmetic operations, respectively. So, the cost of Algorithm 1 is at most

$$6(n - r + 2)(n - 1) + 3 \cdot (\min\{m, n\} - r + 2) + 4 \cdot (\min\{m, n\} - r + 2)(m - r + 1) \\ \leq \sum_{j=r-1}^n 12j + (\min\{m, n\} - r + 2)(4m - 4r + 7) \leq \sum_{j=r-1}^n (4m + 12j).$$

4 The Periodic qd-Type Reduction Method

In this section, we provide a periodic qd-type reduction for a product of rectangular CRD matrices of full rank

$$A = A_1 A_2 \dots A_K, \quad \text{where } A_i \in \mathbb{R}^{n_i \times n_{i+1}}, \quad 1 \leq i \leq K, \quad n_1 = n_{K+1}, \quad (19)$$

under the assumption that no breakdown occurs. First, we illustrate the reduction for $K = 3$ where each factor is a 4×4 CRD matrix of full rank, i.e.,

$$A = A_1 A_2 A_3 = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \begin{bmatrix} y & y & y & y \\ y & y & y & y \\ y & y & y & y \\ y & y & y & y \end{bmatrix} \begin{bmatrix} z & z & z & z \\ z & z & z & z \\ z & z & z & z \\ z & z & z & z \end{bmatrix}.$$

Denote by x_{ij} , y_{ij} and z_{ij} the parameters of A_1 , A_2 and A_3 , respectively. For the first stage, we perform the following operations.

- First, according to the form (1) of A_1 , i.e., $A_1 = B_1 B_2 B_3 D C_3 C_2 C_1$, it can be rewritten as

$$A_1 = \left(B_1 B_2 B_3 D \begin{bmatrix} 1 & x_{12} & & \\ & 1 & x_{23} & \\ & & 1 & x_{34} \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & x_{13} & \\ & & 1 & x_{24} \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & x_{14} \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & -x_{12} & & \\ & 1 & -x_{13} & \\ & & 1 & -x_{14} \\ & & & 1 \end{bmatrix} \right) \\ \times \begin{bmatrix} 1 & -x_{12} & & \\ & 1 & -x_{13} & \\ & & 1 & -x_{14} \\ & & & 1 \end{bmatrix}^{-1}$$

$$\begin{aligned}
 &= \left(B_1 B_2 B_3 D \begin{bmatrix} 1 & & & \\ & 1 & x_{23} & \\ & & 1 & x_{34} \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & x_{24} \\ & & & 1 \end{bmatrix} \right) \begin{bmatrix} 1 & -x_{12} & & \\ & 1 & -x_{13} & \\ & & 1 & -x_{14} \\ & & & 1 \end{bmatrix}^{-1} \\
 &= \begin{bmatrix} x' & 0 & 0 & 0 \\ x' & x' & x' & x' \\ x' & x' & x' & x' \\ x' & x' & x' & x' \end{bmatrix} \begin{bmatrix} 1 & -x_{12} & & \\ & 1 & -x_{13} & \\ & & 1 & -x_{14} \\ & & & 1 \end{bmatrix}^{-1} =: A'_1 S_1^{-1},
 \end{aligned}$$

this means that A'_1 is obtained form A_1 only by setting its parameters $x_{12} = x_{13} = x_{14} = 0$. So,

$$A = A_1 A_2 A_3 = A'_1 (S_1^{-1} A_2) A_3,$$

where by Theorem 2, $A_2 =: S_1^{-1} A_2$ is CRD of full rank whose parameters are still denoted as y_{ij} .

- Further, using a similar argument, the updated CRD matrix A_2 can be rewritten as

$$A_2 = \begin{bmatrix} y' & 0 & 0 & 0 \\ y' & y' & y' & y' \\ y' & y' & y' & y' \\ y' & y' & y' & y' \end{bmatrix} \begin{bmatrix} 1 & -y_{12} & & \\ & 1 & -y_{13} & \\ & & 1 & -y_{14} \\ & & & 1 \end{bmatrix}^{-1} =: A'_2 S_2^{-1},$$

where A'_2 is obtained from A_2 only by setting its parameters $y_{12} = y_{13} = y_{14} = 0$. So,

$$A = A'_1 A_2 A_3 = A'_1 A'_2 (S_2^{-1} A_3),$$

where by Theorem 2, $A_3 =: S_2^{-1} A_3$ is CRD of full rank whose parameters are still denoted as z_{ij} .

- Finally, the updated CRD matrix A_3 can be rewritten as

$$A_3 = \begin{bmatrix} z' & z' & 0 & 0 \\ z' & z' & z' & z' \\ z' & z' & z' & z' \\ z' & z' & z' & z' \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -z_{13} & \\ & & 1 & -z_{14} \\ & & & 1 \end{bmatrix}^{-1} =: A'_3 S_3^{-1},$$

where A'_3 is obtained from A_3 only by setting its parameters $z_{13} = z_{14} = 0$. So,

$$S_3^{-1} A S_3 = S_3^{-1} (A'_1 A'_2 A'_3 S_3^{-1}) S_3 = (S_3^{-1} A'_1) A'_2 A'_3,$$

where by Theorem 2, $A'_1 =: S_3^{-1} A'_1$ is CRD of full rank whose parameters are still denoted as x_{ij} with $x_{1j} = 0$ ($j = 2, 3, 4$) not being disturbed. So far, the product $A = A_1 A_2 A_3$ has been reduced by a similarity transformation into the form:

$$A' = A'_1 A'_2 A'_3 = \begin{bmatrix} x' & 0 & 0 & 0 \\ x' & x' & x' & x' \\ x' & x' & x' & x' \\ x' & x' & x' & x' \end{bmatrix} \begin{bmatrix} y' & 0 & 0 & 0 \\ y' & y' & y' & y' \\ y' & y' & y' & y' \\ y' & y' & y' & y' \end{bmatrix} \begin{bmatrix} z' & z' & 0 & 0 \\ z' & z' & z' & z' \\ z' & z' & z' & z' \\ z' & z' & z' & z' \end{bmatrix}.$$

For the second stage, we perform the following operations.

- First, taking into account the form (1), A'_3 can be rewritten as

$$A'_3 = \begin{bmatrix} 1 & & & \\ -z_{21} & 1 & & \\ & -z_{31} & 1 & \\ & & -z_{41} & 1 \end{bmatrix}^{-1} \begin{bmatrix} z'' & z'' & 0 & 0 \\ 0 & z'' & z'' & z'' \\ 0 & z'' & z'' & z'' \\ 0 & z'' & z'' & z'' \end{bmatrix} =: W_3^{-1} A''_3,$$

where A''_3 is obtained from A'_3 only by setting its parameters $z_{21} = z_{31} = z_{41} = 0$. So,

$$A' = A'_1 A'_2 A'_3 = A'_1 (A'_2 W_3^{-1}) A''_3,$$

where by Corollary 3, $A'_2 =: A'_2 W_3^{-1}$ is CRD of full rank whose parameters are still denoted as y_{ij} with $y_{1j} = 0$ ($j = 2, 3, 4$) not being disturbed.

– Further, the updated CRD matrix A'_2 can be rewritten as

$$A'_2 = \begin{bmatrix} 1 & & & \\ -y_{21} & 1 & & \\ & -y_{31} & 1 & \\ & & -y_{41} & 1 \end{bmatrix}^{-1} \begin{bmatrix} y'' & 0 & 0 & 0 \\ 0 & y'' & y'' & y'' \\ 0 & y'' & y'' & y'' \\ 0 & y'' & y'' & y'' \end{bmatrix} =: W_2^{-1} A''_2,$$

where A''_2 is obtained from A'_2 only by setting its parameters $y_{21} = y_{31} = y_{41} = 0$. So,

$$A' = A'_1 A'_2 A'_3 = (A'_1 W_2^{-1}) A''_2 A''_3,$$

where by Corollary 3, $A'_1 =: A'_1 W_2^{-1}$ is CRD of full rank whose parameters are still denoted as x_{ij} with $x_{1j} = 0$ ($j = 2, 3, 4$) not being disturbed.

– Finally, the updated CRD matrix A'_1 can be rewritten as

$$A'_1 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ -x_{31} & & 1 & \\ & & -x_{41} & 1 \end{bmatrix}^{-1} \begin{bmatrix} x'' & 0 & 0 & 0 \\ x'' & x'' & x'' & x'' \\ 0 & x'' & x'' & x'' \\ 0 & x'' & x'' & x'' \end{bmatrix} =: W_1^{-1} A''_1,$$

where A''_1 is obtained from A'_1 only by setting its parameters $x_{31} = x_{41} = 0$. So,

$$W_1 A' W_1^{-1} = W_1 (W_1^{-1} A''_1 A''_2 A''_3) W_1^{-1} = A''_1 A''_2 (A''_3 W_1^{-1}),$$

where by Corollary 3, $A''_3 =: A''_3 W_1^{-1}$ is CRD of full rank whose parameters are still denoted as z_{ij} with $z_{i1} = 0$ ($i = 2, 3, 4$) and $z_{1j} = 0$ ($j = 3, 4$) not being disturbed. Thus, the product $A = A_1 A_2 A_3$ has been reduced into the form:

$$A''_1 A''_2 A''_3 = \begin{bmatrix} x'' & 0 & 0 & 0 \\ x'' & x'' & x'' & x'' \\ 0 & x'' & x'' & x'' \\ 0 & x'' & x'' & x'' \end{bmatrix} \begin{bmatrix} y'' & 0 & 0 & 0 \\ 0 & y'' & y'' & y'' \\ 0 & y'' & y'' & y'' \\ 0 & y'' & y'' & y'' \end{bmatrix} \begin{bmatrix} z'' & z'' & 0 & 0 \\ 0 & z'' & z'' & z'' \\ 0 & z'' & z'' & z'' \\ 0 & z'' & z'' & z'' \end{bmatrix}.$$

When proceeding with the analogous operations on rows/columns 2 to 4, we conclude that $A = A_1 A_2 A_3$ is reduced by similarity transformations into the tridiagonal form:

$$\begin{bmatrix} \tilde{x} & 0 & 0 & 0 \\ \tilde{x} & \tilde{x} & 0 & 0 \\ 0 & \tilde{x} & \tilde{x} & 0 \\ 0 & 0 & \tilde{x} & \tilde{x} \end{bmatrix} \begin{bmatrix} \tilde{y} & 0 & 0 & 0 \\ 0 & \tilde{y} & 0 & 0 \\ 0 & 0 & \tilde{y} & 0 \\ 0 & 0 & 0 & \tilde{y} \end{bmatrix} \begin{bmatrix} \tilde{z} & \tilde{z} & 0 & 0 \\ 0 & \tilde{z} & \tilde{z} & 0 \\ 0 & 0 & \tilde{z} & \tilde{z} \\ 0 & 0 & 0 & \tilde{z} \end{bmatrix}.$$

For the general product (19), let $r = \min_{1 \leq t \leq K} \{n_t\}$, then by performing the above reductions on rows/columns 1 to r , we conclude that the product $A = A_1 A_2 \dots A_K$ is reduced by similarity transformations into the form

$$T = A'_1 A'_2 \dots A'_{K-1} A'_K$$

$$= \begin{bmatrix} x & & & & \\ x & x & & & \\ & \ddots & \ddots & \ddots & \\ & & x & x & \dots & x \\ & & & \vdots & \ddots & \vdots \\ & & & & x & \dots & x \end{bmatrix} \begin{bmatrix} y & & & & \\ & \ddots & \ddots & \ddots & \\ & & y & \dots & y \\ & & & \vdots & \vdots \\ & & & & y & \dots & y \end{bmatrix} \dots \begin{bmatrix} y & & & & \\ & \ddots & \ddots & \ddots & \\ & & y & \dots & y \\ & & & \vdots & \vdots \\ & & & & y & \dots & y \end{bmatrix} \begin{bmatrix} z & z & & & \\ & \ddots & \ddots & \ddots & \\ & & z & \dots & z \\ & & & \vdots & \vdots \\ & & & & z & \dots & z \end{bmatrix},$$

where if $n_1 = n_{K+1} = r$, then

$$A'_1 = \begin{bmatrix} x & & & 0 \dots 0 \\ x & x & & 0 \dots 0 \\ & \ddots & \ddots & \vdots \\ & & x & x & 0 \dots 0 \end{bmatrix} \in \mathbb{R}^{n_1 \times n_2}, \quad A'_K = \begin{bmatrix} z & z & & \\ & \ddots & \ddots & \\ & & z & z \\ 0 & \dots & 0 & 0 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n_K \times n_{K+1}},$$

and if $n_t = r$ for some $2 \leq t \leq K$, then

$$A'_{t-1} = \begin{bmatrix} y & & \\ & \ddots & \\ 0 & \dots & y & 0 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n_{t-1} \times n_t} \quad (t \neq 2), \quad A'_t = \begin{bmatrix} y & & 0 \dots 0 \\ \ddots & \ddots & \vdots \\ & y & 0 \dots 0 \end{bmatrix} \in \mathbb{R}^{n_t \times n_{t+1}} \quad (t \neq K).$$

So, the product $A = A_1 A_2 \dots A_K$ has been reduced by similarity transformations into

$$T = \begin{bmatrix} T' & 0 \\ 0 & 0 \end{bmatrix}, \tag{20}$$

where if $n_1 = r$, then

$$\begin{aligned} T' &= A'_1[1:r] \cdot (A'_2[1:r] \dots A'_{K-1}[1:r]) \cdot A'_K[1:r] \\ &= \begin{bmatrix} x & & & \\ x & x & & \\ & \ddots & \ddots & \\ & & x & x \end{bmatrix} \begin{bmatrix} y & & \\ & \ddots & \\ & & y \end{bmatrix} \dots \begin{bmatrix} y & & \\ & \ddots & \\ & & y \end{bmatrix} \begin{bmatrix} z & z & & \\ & \ddots & \ddots & \\ & & z & z \\ & & & z & z \end{bmatrix} \in \mathbb{R}^{r \times r}, \end{aligned}$$

and if $n_1 > r$, then

$$\begin{aligned} T' &= A'_1[1:r+1|1:r] \cdot (A'_2[1:r] \dots A'_{K-1}[1:r]) \cdot A'_K[1:r|1:r+1] \\ &= \begin{bmatrix} x & & & \\ x & x & & \\ & \ddots & \ddots & \\ & & x & x \end{bmatrix} \begin{bmatrix} y & & \\ & \ddots & \\ & & y \end{bmatrix} \dots \begin{bmatrix} y & & \\ & \ddots & \\ & & y \end{bmatrix} \begin{bmatrix} z & z & & \\ & \ddots & \ddots & \\ & & z & z \\ & & & z & z \end{bmatrix} \in \mathbb{R}^{(r+1) \times (r+1)}. \end{aligned}$$

According to the analysis above, we finally have Algorithm 2 (here, denote by $U_i(x_i, \dots, x_n) \in \mathbb{R}^{n \times n}$ the unit bidiagonal matrix with the $(t-1, t)$ th entry x_t for $i \leq t \leq n$) to reduce the general product (19) into the tridiagonal form (20), which costs the arithmetic operations of at most

$$\sum_{i=1}^r \left\{ \sum_{t=1}^{K-1} \sum_{j=i}^{n_{t+1}} (4n_{t+2} + 12j) + \sum_{j=i+1}^{n_{K+1}} (4n_2 + 12j) + \sum_{t=2}^K \sum_{j=i}^{n_t} (4n_{t-1} + 12j) + \sum_{j=i+1}^{n_1} (4n_K + 12j) \right\} \approx O(r(n_1 n_2 + n_2 n_3 + \dots + n_K n_{K+1})).$$

So, the complexity of our periodic qd-type reduction is very preferable when r is small.

Observe that the matrix T' of (20) is tridiagonal of the factored form. Consequently, all the eigenvalues of the product A are obtained by applying various tridiagonal eigensolvers such as LR-type algorithms to T' . More interestingly, it turns out that our method can achieve the high relative accuracy, as shown in the next section.

Algorithm 2 The algorithm reduces the product (19) into the tridiagonal form (20) by virtue of the parameter matrices $P_t = (p_{ij}^{(t)}) \in \mathbb{R}^{n_t \times n_{t+1}}$ of the factors $A_t \in \mathbb{R}^{n_t \times n_{t+1}}$ ($1 \leq t \leq K$ and $n_1 = n_{K+1}$).

```

1: for  $i = 1 : \min_{1 \leq t \leq K} \{n_t\}$  do
2:   for  $t = 1 : K$  do
3:     if  $t < K$  then
4:        $A_{t+1} := [U_{i+1}(-p_{i,i+1}^{(t)}, \dots, -p_{i,n_{t+1}}^{(t)})]^{-1} A_{t+1}$  by Algorithm 1
5:        $p_{i,i+1}^{(t)} = \dots = p_{i,n_{t+1}}^{(t)} = 0$ 
6:     else
7:        $A_1 := [U_{i+2}(-p_{i,i+2}^{(K)}, \dots, -p_{i,n_{K+1}}^{(K)})]^{-1} A_1$  by Algorithm 1
8:        $p_{i,i+2}^{(K)} = \dots = p_{i,n_{K+1}}^{(K)} = 0$ 
9:     end if
10:   end for
11:   for  $t = K : -1 : 1$  do
12:     if  $t > 1$  then
13:        $A_{t-1}^T := [U_{i+1}(-p_{i+1,i}^{(t)}, \dots, -p_{n_t,i}^{(t)})]^{-1} A_{t-1}^T$  by Algorithm 1
14:        $p_{i+1,i}^{(t)} = \dots = p_{n_t,i}^{(t)} = 0$ 
15:     else
16:        $A_K^T := [U_{i+2}(-p_{i+2,i}^{(1)}, \dots, -p_{n_1,i}^{(1)})]^{-1} A_K^T$  by Algorithm 1
17:        $p_{i+2,i}^{(1)} = \dots = p_{n_1,i}^{(1)} = 0$ 
18:     end if
19:   end for
20: end for

```

5 Accurate Eigenvalue Computations

In this section, we first identify the subset of the general product (19) for which no subtraction of like-signed numbers occurs throughout the periodic qd-type reduction. We then compute all the eigenvalues of such a product to high relative accuracy. Error analysis is presented to illustrate the high relative accuracy.

5.1 The Subtraction-Free Algorithm 1

Lemma 4 Let $A = B_1 B_2 \dots B_{n-1} \in \mathbb{R}^{n \times n}$ be CRD whose parameter matrix $P = (p_{ij}) \in \mathbb{R}^{n \times n}$ and each factor $B_t \in \mathbb{R}^{n \times n}$ ($1 \leq t \leq n-1$) is as in (1), and let $U \in \mathbb{R}^{n \times n}$ be bidiagonal whose diagonal entries $d_j > 0$ ($1 \leq j \leq n$) and $(j-1, j)$ th entries u_j ($2 \leq j \leq n$). Then the qd-type transformation

$$U^{-1}A = B'_1 B'_2 \dots B'_{n-1} U'^{-1}$$

is subtraction-free if and only if

$$\text{sign}(u_j P[j|1 : j-1]) = -1 \text{ or } 0, \forall 2 \leq j \leq n. \tag{21}$$

Consequently, in this case no breakdown occurs such that each B'_t ($1 \leq t \leq n-1$) has the same form as that of B_t all of whose corresponding entries p'_{ij} satisfy that

$$\text{sign}(p'_{ij}) = \text{sign}(p_{ij}), \forall i > j, \tag{22}$$

and U' is bidiagonal whose diagonal entries $d'_j > 0$ ($1 \leq j \leq n$) and $(j-1, j)$ th entries $u'_j = u_j$ ($2 \leq j \leq n$).

Proof We use induction on the total number of the factors in A to prove the result. By Lemma 1, the qd-type transformation $U^{-1}B_1 = B'_1\bar{U}^{-1}$ is subtraction-free if and only if

$$\text{sign}(d_n p_{n1} u_n) = \text{sign}(p_{n1} u_n) = -1 \text{ or } 0; \tag{23}$$

and consequently, no breakdown occurs such that $\bar{U} \in \mathbb{R}^{n \times n}$ is bidiagonal whose diagonal entries \bar{d}_j and $(j - 1, j)$ th entries \bar{u}_j are the following:

$$\begin{cases} \bar{d}_j = d_j > 0, & 1 \leq j \leq n - 2, \\ \bar{d}_n = d_n - u_n p_{n1} > 0, \quad \bar{d}_{n-1} = \frac{d_{n-1}}{d_n} d_n > 0, \\ \bar{u}_j = u_j, & \forall 2 \leq j \leq n, \end{cases}$$

and B'_1 has the same form as that of B_1 with the corresponding entry $p'_{n1} = \frac{d_{n-1}}{d_n} p_{n1}$ such that

$$\text{sign}(p'_{n1}) = \text{sign}(p_{n1}). \tag{24}$$

Notice that $B_2 \dots B_{n-1}$ is CRD whose parameters p_{ij} ($0 < i - j \leq n - 2$) satisfy the fact (2). Hence, by applying our induction assumption, the qd-type transformation

$$\bar{U}^{-1}B_2 \dots B_{n-1} = B'_2 \dots B'_{n-1}U'^{-1}$$

is subtraction-free if and only if

$$\text{sign}(u_j P[j|1 : j - 1]) = -1 \text{ or } 0, \quad \forall 2 \leq j \leq n - 1; \quad \text{sign}(u_n P[n|2 : n - 1]) = -1 \text{ or } 0; \tag{25}$$

and consequently, no breakdown occurs such that each B'_t ($2 \leq t \leq n - 1$) has the same form as that of B_t all of whose corresponding entries p'_{ij} satisfy that

$$\text{sign}(p'_{ij}) = \text{sign}(p_{ij}), \quad \forall i > j \text{ and } (i, j) \neq (n, 1), \tag{26}$$

and $U' \in \mathbb{R}^{n \times n}$ is bidiagonal whose diagonal entries $d'_j > 0$ ($1 \leq j \leq n$) and $(j - 1, j)$ th entries $u'_j = u_j$ ($2 \leq j \leq n$). Therefore, the fact (25) together with the fact (23) implies that the qd-type transformation $A' = U^{-1}A$ is subtraction-free if and only if the fact (21) holds. Consequently, the facts (24) and (26) imply that the fact (22) is true. \square

Lemma 5 *Let $A = C_{m-1} \dots C_2 C_1 \in \mathbb{R}^{m \times m}$ be CRD whose parameter matrix $P \in \mathbb{R}^{m \times m}$ and each factor $C_t \in \mathbb{R}^{m \times m}$ ($1 \leq t \leq m - 1$) is as in (1), and let $U \in \mathbb{R}^{m \times m}$ be bidiagonal with unit diagonal whose $(j - 1, j)$ th entries $u_j \neq 0$ for all $1 \leq j \leq h$ and $u_j = 0$ otherwise. Then the qd-type transformation*

$$U^{-1}A = C'_{m-1} \dots C'_2 C'_1$$

is subtraction-free if and only if

$$\begin{cases} \text{sign}(u_j P[l - 1, \dots, j - 1|j]) = -1 \text{ or } 0, & \forall l \leq j \leq h; \\ \text{sign}(P[l - 1, \dots, h|j]) = s_j, & \forall h + 1 \leq j \leq m. \end{cases} \tag{27}$$

Consequently, no breakdown occurs such that $A' = U^{-1}A$ is CRD whose parameter matrix P' satisfies that

$$\begin{cases} \text{sign}(u_j P'[l - 1, \dots, j - 1|j]) = -1, & \forall l \leq j \leq h; \\ \text{sign}(P'[l - 1, \dots, h|j]) = s_j, & \forall h + 1 \leq j \leq m; \\ P'[i|j] = P[i|j], & \text{otherwise;} \end{cases} \tag{28}$$

Proof We use induction on the total number of the factors in A to prove the result. By Lemma 3, the qd-type transformation $U^{-1}C_{m-1} = C'_{m-1}\bar{U}^{-1}$ is subtraction-free if and only if

$$\text{sign}(u_j p_{j-1,j}) = -1 \text{ or } 0, \quad \forall l \leq j \leq h; \tag{29}$$

and consequently, no breakdown occurs such that $\bar{U} \in \mathbb{R}^{m \times m}$ is bidiagonal with unit diagonal whose $(j - 1, j)$ th entries \bar{u}_j ($2 \leq j \leq m$) are the following:

$$\bar{u}_j = \frac{p_{j-1,j} - u_j}{p_{j-2,j-1} - u_{j-1}} u_{j-1}, \quad l + 1 \leq j \leq \min\{h + 1, m\}; \text{ otherwise } \bar{u}_j = 0,$$

from which it follows that

$$\begin{cases} \text{sign}(\bar{u}_j) = -\text{sign}(p_{j-1,j} - u_j) = \text{sign}(u_j), \quad l + 1 \leq j \leq h; \\ \text{sign}(\bar{u}_{h+1}) = -\text{sign}(p_{h,h+1} - u_{h+1}) = -\text{sign}(p_{h,h+1}), \quad \text{if } h + 1 \leq m; \end{cases} \tag{30}$$

and C'_{m-1} has the same form as that of C_{m-1} with the corresponding bidiagonal entries

$$\begin{cases} p'_{l-1,l} = p_{l-1,l} - u_l \neq 0, \\ p'_{j-1,j} = \frac{p_{j-1,j} - u_j}{p_{j-2,j-1} - u_{j-1}} p_{j-2,j-1}, \quad \forall l + 1 \leq j \leq \min\{h + 1, m\}, \\ p'_{j-1,j} = p_{j-1,j}, \quad \text{otherwise,} \end{cases}$$

from which it follows that

$$\begin{cases} \text{sign}(u_l p'_{l-1,l}) = \text{sign}(u_l(p_{l-1,l} - u_l)) = -1, \\ \text{sign}(u_j p'_{j-1,j}) = -1 \text{ or } 0, \quad \forall l + 1 \leq j \leq h, \\ \text{sign}(p'_{h,h+1}) = \text{sign}(p_{h,h+1}) \text{ or } 0, \quad \text{if } h + 1 \leq m. \end{cases} \tag{31}$$

Notice that $C_{m-2} \dots C_1$ is CRD whose parameters p_{ij} ($j - i \geq 2$) satisfy the fact (2). There are the following cases that we need to consider.

- For the case $p_{h,h+1} \neq 0$, we have by the fact (30) that

$$\bar{u}_j \neq 0, \quad l + 1 \leq j \leq h + 1; \text{ otherwise } \bar{u}_j = 0.$$

Thus, by our induction assumption, the qd-type transformation

$$A'' = \bar{U}^{-1}C_{m-2} \dots C_1 = C'_{m-2} \dots C'_1$$

is subtraction-free if and only if

$$\begin{cases} \text{sign}(\bar{u}_j P[(l + 1) - 2, \dots, j - 2|j]) = -1 \text{ or } 0, \quad \forall l + 1 \leq j \leq h + 1; \\ \text{sign}(P[(l + 1) - 2, \dots, (h + 2) - 2|j]) = s_j, \quad \forall h + 2 \leq j \leq m; \end{cases} \tag{32}$$

and consequently, no breakdown occurs such that the parameter matrix $P' = (p'_{ij})$ of A'' satisfies that

$$\begin{cases} \text{sign}(\bar{u}_j P'[(l + 1) - 2, \dots, j - 2|j]) = -1, \quad \forall l + 1 \leq j \leq h + 1; \\ \text{sign}(P'[(l + 1) - 2, \dots, (h + 2) - 2|j]) = s_j, \quad \forall h + 2 \leq j \leq m; \\ p'_{ij} = p_{ij} \quad (j - i \neq 1), \text{ otherwise.} \end{cases} \tag{33}$$

Therefore, the fact (32) together with the facts (29), (30) implies that the qd-type transformation $A' = U^{-1}A$ is subtraction-free if and only if the fact (27) is true. Consequently, the fact (33) with (30), (31) implies that the fact (28) is true.

– For the case $p_{h,h+1} = 0$, we have by the fact (30) that

$$\bar{u}_j \neq 0, \quad l + 1 \leq j \leq h; \quad \text{otherwise } \bar{u}_j = 0.$$

Thus, by our induction assumption, the qd-type transformation

$$A'' = \bar{U}^{-1}C_{m-2} \dots C_1 = C'_{m-2} \dots C'_1$$

is subtraction-free if and only if

$$\begin{cases} \text{sign}(\bar{u}_j P[(l + 1) - 2, \dots, j - 2|j]) = -1 \text{ or } 0, \quad \forall l + 1 \leq j \leq h; \\ \text{sign}(P[(l + 1) - 2, \dots, (h + 1) - 2|j]) = s_j, \quad \forall h + 1 \leq j \leq m; \end{cases} \quad (34)$$

and consequently, no breakdown occurs such that the parameter matrix $P' = (p'_{ij})$ of A'' satisfies that

$$\begin{cases} \text{sign}(\bar{u}_j P'[(l + 1) - 2, \dots, j - 2|j]) = -1, \quad \forall l + 1 \leq j \leq h; \\ \text{sign}(P'[(l + 1) - 2, \dots, (h + 1) - 2|j]) = s_j, \quad \forall h + 1 \leq j \leq m; \\ p'_{ij} = p_{ij} \quad (j - i \neq 1), \quad \text{otherwise.} \end{cases} \quad (35)$$

Notice that by the fact (2), $p_{h,h+1} = 0$ implies that $p_{hj} = 0$ for all $j \geq h + 1$. Therefore, the fact (34) together with the facts (29), (30) implies that the qd-type transformation $A' = U^{-1}A$ is subtraction-free if and only if the fact (27) is true. Consequently, the fact (35) with (30), (31) implies that the fact (28) is true.

The result is proved. □

By Lemmas 4 and 5, we identify Algorithm 1 to be subtraction-free as follows.

Theorem 3 *Let $A \in \mathbb{R}^{n \times m}$ be CRD of full rank with the parameter matrix $P \in \mathbb{R}^{n \times m}$, and let $U \in \mathbb{R}^{n \times n}$ be bidiagonal whose diagonal entries $d_j > 0$ ($1 \leq j \leq n$) and $(j - 1, j)$ th entries u_j ($2 \leq j \leq n$) are the following:*

$$\text{sign}(u_j) = g_j \neq 0, \quad l_r \leq j \leq h_r, \quad r = 1, 2, \dots, t; \quad \text{otherwise } u_j = 0,$$

here, $l_{r+1} - h_r \geq 2$ for all $1 \leq r \leq t - 1$. Denote $\text{sign}(P[j|j]) = w_j$ for all $1 \leq j \leq \min\{n, m\}$. Then the qd-type transformation $A' = U^{-1}A$ by Algorithm 1 is subtraction-free if and only if

$$\begin{cases} \text{sign}(P[j|1 : j - 1]) = -g_j \text{ or } 0, \quad \forall l_r \leq j \leq h_r; \\ \text{sign}(P[l_r - 1, \dots, j - 1|j]) = -g_j w_{j-1} w_j \text{ or } 0, \quad \forall l_r \leq j \leq h'_r; \quad r = 1, 2, \dots, t, \\ \text{sign}(P[l_r - 1, \dots, h'_r|j]) = s_j, \quad \forall h'_r + 1 \leq j \leq m; \end{cases}$$

here $h'_r = \min\{h_r, n, m\}$. Consequently, in this case no breakdown occurs such that A' is CRD of full rank whose parameter matrix $P' \in \mathbb{R}^{n \times m}$ satisfies that

$$\begin{cases} \text{sign}(P'[l_r - 1, \dots, j - 1|j]) = -g_j w_{j-1} w_j, \quad \forall l_r \leq j \leq h'_r; \\ \text{sign}(P'[l_r - 1, \dots, h'_r|j]) = s_j, \quad \forall h'_r + 1 \leq j \leq m; \\ \text{sign}(P'[i|j]) = \text{sign}(P[i|j]), \quad \text{otherwise;} \end{cases} \quad r = 1, 2, \dots, t.$$

Proof Denote by $U_{l_r} \in \mathbb{R}^{n \times n}$ ($1 \leq r \leq t$) the bidiagonal matrix whose diagonal entries \tilde{d}_j ($1 \leq j \leq n$) and $(j - 1, j)$ th entries \tilde{u}_j ($2 \leq j \leq n$) are the following:

$$\begin{cases} \tilde{d}_j = d_j, \quad l_r - 1 \leq j \leq h_r; \quad \text{otherwise } \tilde{d}_j = 1; \\ \tilde{u}_j = u_j, \quad l_r \leq j \leq h_r; \quad \text{otherwise } \tilde{u}_j = 0. \end{cases}$$

Consider that $l_{r+1} - h_r \geq 2$ for all $1 \leq r \leq t - 1$. Then $U = U_{l_1} \dots U_{l_2} U_{l_1}$. Hence, the qd-type transformation $U^{-1}A = U_{l_1}^{-1}U_{l_2}^{-1} \dots U_{l_t}^{-1}A$ is subtraction-free if and only if the sequence qd-type transformations $A_{r-1} = U_{l_r}^{-1}A_r$ (denote $A_t = A$) for $r = t, \dots, 2, 1$ are subtraction-free. Therefore, we conclude by Lemmas 4, 2 and 5 that the result is true. \square

Remark 1 In Theorem 3, all the diagonal entries of U have been assumed to be positive. Clearly, a general case can be easily dealt with by choosing a signature matrix S such that all the diagonal entries of SU are positive.

5.2 The Subtraction-Free LR Algorithm

Given the LU factorization $T_0 = L_0U_0$, the basic LR algorithm is performed as follows:

$$\begin{cases} T_k = L_kU_k, \\ T_{k+1} = L_k^{-1}T_kL_k = U_kL_k, \end{cases} \quad k = 0, 1, \dots \tag{36}$$

Lemma 6 Let $T \in \mathbb{R}^{n \times n}$ be nonsingular as follows

$$T = LU = \begin{bmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & l_n \end{bmatrix} \begin{bmatrix} d_1 & u_2 & & \\ & \ddots & \ddots & \\ & & d_{n-1} & u_n \\ & & & d_n \end{bmatrix}. \tag{37}$$

Then in the absence of breakdown, the LR transformation

$$UL = \bar{L}\bar{U} =: \begin{bmatrix} 1 & & & \\ \bar{l}_2 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & \bar{l}_n \end{bmatrix} \begin{bmatrix} \bar{d}_1 & u_2 & & \\ & \ddots & \ddots & \\ & & \bar{d}_{n-1} & u_n \\ & & & \bar{d}_n \end{bmatrix}, \tag{38}$$

where with the convention $l_{n+1} = u_{n+1} = 0$,

$$\begin{cases} z_1 = d_1, \bar{d}_j = z_j + l_{j+1}u_{j+1}, \\ \bar{l}_{j+1} = \frac{d_{j+1}}{d_j}l_{j+1}, z_{j+1} = \frac{d_{j+1}}{d_j}z_j, \end{cases} \quad \forall 1 \leq j \leq n. \tag{39}$$

Moreover, the transformation (38) is subtraction-free if and only if $\text{sign}(d_j l_{j+1} u_{j+1}) = 1$ or 0 for all $1 \leq j \leq n - 1$. Consequently, in this case no breakdown occurs such that for all $1 \leq j \leq n$, $\text{sign}(\bar{d}_j) = \text{sign}(d_j)$, $\text{sign}(\bar{d}_j \bar{l}_{j+1} u_{j+1}) = \text{sign}(d_j d_{j+1})$ or 0, and $\bar{l}_{j+1} = 0$ if and only if $l_{j+1} = 0$.

Proof Comparing the entries in both sides of (38), we have with the convention $l_{n+1} = u_{n+1} = 0$ that

$$\begin{cases} \bar{d}_1 = d_1 + l_2 u_2, \\ \bar{d}_j \bar{l}_{j+1} = d_{j+1} l_{j+1}, \\ \bar{d}_{j+1} + \bar{l}_{j+1} u_{j+1} = d_{j+1} + l_{j+2} u_{j+2}, \end{cases} \quad j = 1, 2, \dots, n - 1,$$

from which it follows that

$$\begin{cases} z_1 = \bar{d}_1 - l_2 u_2 = d_1, \bar{d}_1 = z_1 + l_2 u_2, \\ \bar{l}_{j+1} = \frac{d_{j+1}}{d_j} l_{j+1}, \\ z_{j+1} = \bar{d}_{j+1} - l_{j+2} u_{j+2} = d_{j+1} - \bar{l}_{j+1} u_{j+1} = \frac{d_{j+1}}{d_j} z_j, \\ \bar{d}_{j+1} = z_{j+1} + l_{j+2} u_{j+2}, \end{cases} \quad j = 1, 2, \dots, n - 1.$$

So, the formula (39) is derived. Moreover, by the formula (39), since $z_1 = d_1 \neq 0$ with $\text{sign}(z_1) = \text{sign}(d_1)$, we have that $\bar{d}_1 = z_1 + l_2u_2$ is subtraction-free if and only if $\text{sign}(z_1l_2u_2) = \text{sign}(d_1l_2u_2) = 1$ or 0. Consequently, $\bar{d}_1 \neq 0$ with $\text{sign}(\bar{d}_1) = \text{sign}(z_1) = \text{sign}(d_1)$ such that $z_2 = \frac{d_2}{d_1}z_1 \neq 0$ with $\text{sign}(z_2) = \text{sign}(d_2)$. So, assume that $z_j \neq 0$ with $\text{sign}(z_j) = \text{sign}(d_j)$ for some $1 \leq j \leq n - 1$. Then $\bar{d}_j = z_j + l_{j+1}u_{j+1}$ is subtraction-free if and only if $\text{sign}(z_jl_{j+1}u_{j+1}) = \text{sign}(d_jl_{j+1}u_{j+1}) = 1$ or 0, and consequently, $\bar{d}_j \neq 0$ with $\text{sign}(\bar{d}_j) = \text{sign}(z_j) = \text{sign}(d_j)$ such that $z_{j+1} = \frac{d_{j+1}}{d_j}z_j \neq 0$ with $\text{sign}(z_{j+1}) = \text{sign}(d_{j+1})$. Notice that $\bar{d}_n = z_n$. Thus, we conclude that the LR transformation (38) is subtraction-free if and only if $\text{sign}(d_jl_{j+1}u_{j+1}) = 1$ or 0 for all $1 \leq j \leq n - 1$. Consequently, no breakdown occurs such that for all $1 \leq j \leq n$, $\text{sign}(\bar{d}_j) = \text{sign}(d_j)$, and $\text{sign}(\bar{d}_jl_{j+1}u_{j+1}) = \text{sign}(d_jd_{j+1})$ or 0 because $\bar{d}_jl_{j+1} = d_jl_{j+1}$ and $\text{sign}(l_{j+1}u_{j+1}) = \text{sign}(d_j)$ or 0. Clearly, $l_{j+1} = 0$ if and only if $l_{j+1} = 0$. \square

Now, we identify the basic LR algorithm to be subtraction-free as follows.

Theorem 4 *Let $T \in \mathbb{R}^{n \times n}$ be as in (37). Then the basic LR algorithm of T is subtraction-free if and only if $\text{sign}(l_{j+1}u_{j+1}) = \text{sign}(d_j) = \text{sign}(d_{j+1})$ whenever $l_{j+1}u_{j+1} \neq 0$ for all $1 \leq j \leq n - 1$. Consequently, in this case no breakdown occurs.*

Proof By Lemma 6, the first LR iteration of (36) is subtraction-free if and only if $\text{sign}(d_jl_{j+1}u_{j+1}) = 1$ whenever $l_{j+1}u_{j+1} \neq 0$ for all j ; and then, the second LR iteration of (36) is subtraction-free if and only if $\text{sign}(d_jd_{j+1}) = 1$ whenever $l_{j+1}u_{j+1} \neq 0$ by considering that $l_{j+1} = 0$ if and only if $l_{j+1} = 0$; and afterwards, the r th ($r = 3, 4, \dots$) iteration of (36) must be subtraction-free by considering $\text{sign}(\bar{d}_j) = \text{sign}(d_j)$ for all j . Notice that the facts $\text{sign}(d_jl_{j+1}u_{j+1}) = 1$ and $\text{sign}(d_jd_{j+1}) = 1$ imply that $\text{sign}(l_{j+1}u_{j+1}) = \text{sign}(d_j) = \text{sign}(d_{j+1})$. \square

5.3 Computing Eigenvalues with High Relative Accuracy

Now we are ready to show that our eigensolver of combining Algorithm 2 with the LR algorithm can achieve the high relative accuracy for computing all the eigenvalues of the product $A = A_1A_2 \dots A_K \in \mathbb{R}^{n \times n}$ of (19). For simplicity, assume that each factor A_t ($1 \leq t \leq K$) is square of full rank, which can be generalized to the rectangular case by taking account into Theorems 3 and 4.

Before proceeding, we remark that if there exists $2 \leq r \leq n$ such that $P_t[1 : r - 1|r] = 0$ for all $1 \leq t \leq K$, then by the fact (2), $P_t[1 : r - 1|r : n] = 0$ such that

$$A_t = \begin{bmatrix} A_{11}^{(t)} & 0 \\ A_{21}^{(t)} & A_{22}^{(t)} \end{bmatrix} \in \mathbb{R}^{n \times n}, \text{ where } A_{11}^{(t)} \in \mathbb{R}^{r \times r}, \forall 1 \leq t \leq K,$$

consequently, the eigenvalue problem of the product A is split into two subproblems of the products $A_{11} = \prod_{t=1}^K A_{11}^{(t)}$ and $A_{22} = \prod_{t=1}^K A_{22}^{(t)}$. A similar splitting fact holds if there exists $2 \leq r \leq n$ such that $P_t[r|1 : r - 1] = 0$ for all $1 \leq t \leq K$. Hence, assume that for all $2 \leq r \leq n$, there exist $1 \leq t_1^{(r)}, t_2^{(r)} \leq K$ such that

$$P_{t_1^{(r)}}[1 : r - 1|r] \neq 0 \text{ and } P_{t_2^{(r)}}[r|1 : r - 1] \neq 0. \tag{40}$$

Now, we identify the subset of these products for which the periodic qd-type reduction method is subtraction-free. Recall the convention $\prod_l^h \cdot = 1$ if $h < l$.

Theorem 5 Let $A = A_1 \dots A_{K-1} A_K \in \mathbb{R}^{n \times n}$, where each factor $A_t \in \mathbb{R}^{n \times n}$ ($1 \leq t \leq K$) is a nonsingular CRD matrix with the parameter matrix $P_t \in \mathbb{R}^{n \times n}$ such that the fact (40) is satisfied. Then the periodic qd-type reduction method, i.e., Algorithm 2, is subtraction-free to reduce A into a tridiagonal matrix whose basic LR algorithm is subtraction-free if and only if for all $1 \leq t \leq K$,

$$\begin{cases} \text{sign}(P_t[j|1 : j - 1]) = f_j^{(t)}, \\ \text{sign}(P_t[1 : j - 1|j]) = s_j^{(t)}, \\ \text{sign}(P_t[j|j]) = w_j^{(t)}, \end{cases} \quad \forall 1 \leq j \leq n, \tag{41}$$

where

$$\begin{cases} \prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1, \\ f_j^{(t_1)} f_j^{(t_2)} = \prod_{l=t_1}^{t_2-1} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } f_j^{(t_1)} f_j^{(t_2)} \neq 0 \text{ for } t_1 < t_2, \\ s_j^{(t_1)} s_j^{(t_2)} = \prod_{l=t_1+1}^{t_2} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } s_j^{(t_1)} s_j^{(t_2)} \neq 0 \text{ for } t_1 < t_2, \\ s_j^{(t_1)} f_j^{(t_2)} = \prod_{l=\min\{t_1+1, t_2\}}^{\max\{t_1, t_2-1\}} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } s_j^{(t_1)} f_j^{(t_2)} \neq 0, \end{cases} \quad \forall 2 \leq j \leq n. \tag{42}$$

Proof We use induction on the total number h of the eliminated rows and columns of the factors to prove the result. If $h = 0$, then A itself is of the tridiagonal form with $A = LU$, where $L \in \mathbb{R}^{n \times n}$ is bidiagonal with unit diagonal whose $(j, j - 1)$ th entries $l_j = p_{j,j-1}^{(1)}$ ($2 \leq j \leq n$), and $U \in \mathbb{R}^{n \times n}$ is bidiagonal whose diagonal entries $d_j = \prod_{l=1}^K p_{jj}^{(l)}$ ($1 \leq j \leq n$) and $(j - 1, j)$ th entries $u_j = d_{j-1} p_{j-1,j}^{(K)}$ ($2 \leq j \leq n$). Because of the fact (40), $l_j \neq 0$ and $u_j \neq 0$ for all $2 \leq j \leq n$. Thus, by Theorem 4, the basic LR algorithm of A is subtraction-free if and only if

$$\text{sign}(l_j u_j) = \text{sign}(d_{j-1}) = \text{sign}(d_j), \quad \forall 2 \leq j \leq n,$$

i.e., for all $2 \leq j \leq n$,

$$\text{sign}(p_{j,j-1}^{(1)} p_{j-1,j}^{(K)}) = f_j^{(1)} s_j^{(K)} = 1, \quad \text{sign}(d_{j-1} d_j) = \prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1,$$

this means that the result is true for the case $h = 0$. Now assume that the result is true for the case that the total number of the eliminated rows and columns is less than h . Notice that for the periodic qd-type method, the eliminations on columns of $A_1 \dots A_{K-1} A_K$ are equivalent to the eliminations on rows of $(A_1 \dots A_{K-1} A_K)^T$. So, without loss of generality, assume the first elimination of the method is performed on the first row of the factor A_t for some $1 \leq t \leq K$. There are the following cases to be considered.

- The case $1 \leq t < K$. For the parameter matrix $P_t = (p_{ij}^{(t)}) \in \mathbb{R}^{n \times n}$ of A_t , if $p_{1j}^{(t)} = 0$ for all $2 \leq j \leq n$, then all the off-diagonal entries on the 1th row of A_t are zero, and so, there is nothing to be eliminated. Thus, by the fact (2), we assume that

$$p_{11}^{(t)} \neq 0, \dots, p_{1h}^{(t)} \neq 0 \ (h \geq 2); \quad p_{1j}^{(t)} = 0, \quad \forall j > h.$$

Let $U = U_2(-p_{12}^{(t)}, \dots, -p_{1h}^{(t)}) \in \mathbb{R}^{n \times n}$ with $\text{sign}(p_{1j}^{(t)}) = s_j^{(t)} \neq 0$ ($2 \leq j \leq h$). Then the elimination of these nonzero entries is performed by the periodic qd-type method as follows

$$A_1 \dots A_{t-1} \underline{A_t A_{t+1}} \dots A_K = A_1 \dots A_{t-1} \underline{A'_t A'_{t+1}} \dots A_K, \tag{43}$$

where A'_t is a nonsingular CRD matrix obtained from A_t only by setting its $(1, j)$ th $(2 \leq j \leq h)$ parameters to be zero, i.e., the parameter matrix $P'_t \in \mathbb{R}^{n \times n}$ of A'_t is the following:

$$P'_t[1|j] = 0, \forall 2 \leq j \leq n; \quad P'_t[i|j] = P_t[i|j], \text{ otherwise};$$

and the qd-type transformation $A'_{t+1} = U^{-1}A_{t+1}$ is computed by Algorithm 1. Thus, by Theorem 3, the elimination of (43) is subtraction-free if and only if the fact (41) with (42) is satisfied for the parameter matrix $P_{t+1} \in \mathbb{R}^{n \times n}$ as follows:

$$\begin{cases} \text{sign}(P_{t+1}[j|j]) = w_j^{(t+1)}, \quad \forall 1 \leq j \leq h, \\ \text{sign}(P_{t+1}[j|1 : j - 1]) = f_j^{(t+1)} = s_j^{(t)} \text{ or } 0, \quad \forall 2 \leq j \leq h, \\ \text{sign}(P_{t+1}[1 : j - 1|j]) = s_j^{(t+1)} = s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} \text{ or } 0, \quad \forall 2 \leq j \leq h, \\ \text{sign}(P_{t+1}[1 : h|j]) = s_j^{(t+1)}, \quad \forall h + 1 \leq j \leq n, \end{cases} \tag{44}$$

and consequently, A'_{t+1} is a nonsingular CRD matrix whose parameter matrix $P'_{t+1} \in \mathbb{R}^{n \times n}$ satisfies that

$$\begin{cases} \text{sign}(P'_{t+1}[1 : j - 1|j]) = s_j^{(t+1)} = s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} \neq 0, \quad \forall 2 \leq j \leq h, \\ \text{sign}(P'_{t+1}[1 : h|j]) = s_j^{(t+1)} = s_j^{(t+1)}, \quad \forall h + 1 \leq j \leq n, \\ \text{sign}(P'_{t+1}[i|j]) = \text{sign}(P_{t+1}[i|j]), \text{ otherwise.} \end{cases} \tag{45}$$

In particular, the fact (40) is satisfied for the product $A_1 \dots A'_t A'_{t+1} \dots A_K$.

- For the sufficiency, it need to show that the fact (41) with (42) holds for all the factors of $A_1 \dots A'_t A'_{t+1} \dots A_K$. Consider that the fact (41) with (42) has been satisfied for all the factors A_i ($1 \leq i \leq K$). Because of the connections between P_j and P'_j ($j = t, t + 1$), it remains to show that the fact (42) is satisfied for all $s_j^{(t+1)}$ ($2 \leq j \leq h$) of P'_{t+1} , which is proved as follows: for all $2 \leq j \leq h$,

$$\begin{cases} s_j^{(r)} s_j^{(t+1)} = s_j^{(r)} s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} = \prod_{l=r+1}^{t+1} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } s_j^{(r)} \neq 0 \text{ for } r \leq t, \\ s_j^{(t+1)} s_j^{(r)} = s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} s_j^{(r)} = \prod_{l=t+2}^r w_{j-1}^{(l)} w_j^{(l)}, \text{ if } s_j^{(r)} \neq 0 \text{ for } r > t + 1, \\ f_j^{(r)} s_j^{(t+1)} = f_j^{(r)} s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} = \prod_{l=r}^{t+1} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } f_j^{(r)} \neq 0 \text{ for } r \leq t + 1, \\ s_j^{(t+1)} f_j^{(r)} = s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} f_j^{(r)} = \prod_{l=t+2}^{r-1} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } f_j^{(r)} \neq 0 \text{ for } r > t + 1. \end{cases}$$

So, we conclude by the induction assumption that the sufficiency is true.

- For the necessity, by the induction assumption, the fact (41) with (42) is true for all the factors of $A_1 \dots A'_t A'_{t+1} \dots A_K$, and in particular,

$$\begin{cases} P'_t[1|j] = 0, \quad \forall 2 \leq j \leq n; \\ \text{sign}(P'_t[j|1 : j - 1]) = \text{sign}(P_t[j|1 : j - 1]) = f_j^{(t)}, \quad 2 \leq j \leq n; \\ \text{sign}(P'_t[1 : j - 1|j]) = \text{sign}(P_t[2 : j - 1|j]) = s_j^{(t)}, \quad 3 \leq j \leq h; \\ \text{sign}(P'_t[1 : j - 1|j]) = \text{sign}(P_t[1 : j - 1|j]) = s_j^{(t)}, \quad h + 1 \leq j \leq n; \\ \text{sign}(P'_t[j|j]) = \text{sign}(P_t[j|j]) = w_j^{(t)}, \quad 1 \leq j \leq n, \end{cases}$$

where if $s_j^{(t)} \neq 0$ for some $3 \leq j \leq h$, since $s_j^{(t+1)} = s_j^{(t)} w_{j-1}^{(t+1)} w_j^{(t+1)} \neq 0$ and $s_j^{(t)} s_j^{(t+1)} = w_{j-1}^{(t+1)} w_j^{(t+1)}$, we have that $s_j^{(t)} = s_j^{(t)}$, which together with $\text{sign}(p_{1j}^{(t)}) = s_j^{(t)} \neq 0$ ($2 \leq j \leq h$) implies that

$$\text{sign}(P_t[1 : j - 1|j]) = s_j^{(t)} \neq 0, \quad \forall 2 \leq j \leq h.$$

For showing that the fact (41) with (42) holds for all the factors A_i ($1 \leq i \leq K$), because of the connections between P_j and P'_j ($j = t, t + 1$), it remains to show that the fact (42) is satisfied for all $s_j^{(t)}$ ($2 \leq j \leq h$) of P_t , which is proved as follows: for all $2 \leq j \leq h$, by considering that $s_j^{(t+1)} \neq 0$, since

$$\begin{cases} s_j^{(r)} s_j^{(t+1)} = \prod_{l=r+1}^{t+1} w_{j-1}^{(l)} w_j^{(l)}, & \text{if } s_j^{(r)} \neq 0 \text{ for } r < t, \\ s_j^{(t+1)} s_j^{(r)} = \prod_{l=t+2}^r w_{j-1}^{(l)} w_j^{(l)}, & \text{if } s_j^{(r)} \neq 0 \text{ for } r \geq t + 1, \\ f_j^{(r)} s_j^{(t+1)} = \prod_{l=r}^{t+1} w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for } r \leq t + 1, \\ s_j^{(t+1)} f_j^{(r)} = \prod_{l=t+2}^{r-1} w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for } r > t + 1, \end{cases}$$

we have that

$$\begin{cases} s_j^{(r)} s_j^{(t)} = \prod_{l=r+1}^t w_{j-1}^{(l)} w_j^{(l)}, & \text{if } s_j^{(r)} \neq 0 \text{ for } r < t, \\ s_j^{(t)} s_j^{(r)} = \prod_{l=t+1}^r w_{j-1}^{(l)} w_j^{(l)}, & \text{if } s_j^{(r)} \neq 0 \text{ for } r > t + 1, \\ f_j^{(r)} s_j^{(t)} = \prod_{l=r}^t w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for } r \leq t + 1, \\ s_j^{(t)} f_j^{(r)} = \prod_{l=t+1}^{r-1} w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for } r > t + 1. \end{cases}$$

Therefore, we conclude that the necessity is true.

- The case $t = K$. For the parameter matrix $P_K = (p_{ij}^{(K)}) \in \mathbb{R}^{n \times n}$ of A_K , if $p_{1j}^{(K)} = 0$ for all $3 \leq j \leq n$, then all the $(1, j)$ -th ($3 \leq j \leq n$) entries of A_K are zero, and so, there is nothing to be eliminated. Thus, by the fact (2), we assume that

$$p_{11}^{(K)} \neq 0, \dots, p_{1h}^{(K)} \neq 0 \ (h \geq 3); \quad p_{1j}^{(K)} = 0, \quad \forall j > h.$$

Let $U = U_3(-p_{13}^{(K)}, \dots, -p_{1h}^{(K)})$ with $\text{sign}(p_{1j}^{(K)}) = s_j^{(K)} \neq 0$ ($3 \leq j \leq h$). Then the elimination of these nonzero entries is performed by the periodic qd-type method as follows

$$U^{-1}(A_1 \dots A_{K-1} A_K)U = A'_1 A_2 \dots A_{K-1} A'_K, \tag{46}$$

where A'_K is a nonsingular CRD matrix obtained from A_K only by setting its nonzero $(1, j)$ th ($3 \leq j \leq h$) parameters to be zero, i.e., the parameter matrix $P'_K \in \mathbb{R}^{n \times n}$ of A'_K is the following:

$$P'_K[1|j] = 0, \quad \forall 3 \leq j \leq n; \quad P'_K[i|j] = P_K[i|j], \text{ otherwise};$$

and the qd-type transformation $A'_1 = U^{-1}A_1$ is computed by Algorithm 1. Notice that all the $(1, j)$ -th ($2 \leq j \leq n$) entries of A_1 are zero, this means that $P_1[1|j] = 0$ for all $2 \leq j \leq n$. Thus, by Theorem 3, the elimination of (46) is subtraction-free if and only

if the fact (41) with (42) is satisfied for the parameter matrix $P_1 \in \mathbb{R}^{n \times n}$ as follows:

$$\begin{cases} \text{sign}(P_1[j|1 : j - 1]) = f_j^{(1)} = s_j^{(K)} \text{ or } 0, \forall 3 \leq j \leq h, \\ \text{sign}(P_1[1 : j - 1|j]) = s_j^{(1)} = s_j^{(K)} w_{j-1}^{(1)} w_j^{(1)} \text{ or } 0, \forall 3 \leq j \leq h, \\ \text{sign}(P_1[1 : h|j]) = s_j^{(1)}, \forall h + 1 \leq j \leq n, \end{cases} \quad (47)$$

and consequently, A'_1 is a nonsingular CRD matrix whose parameter matrix $P'_1 \in \mathbb{R}^{n \times n}$ satisfies that

$$\begin{cases} \text{sign}(P'_1[1 : j - 1|j]) = s_j^{(1)} = s_j^{(K)} w_{j-1}^{(1)} w_j^{(1)} \neq 0, \forall 3 \leq j \leq h, \\ \text{sign}(P'_1[1 : h|j]) = s_j^{(1)} = s_j^{(1)}, \forall h + 1 \leq j \leq n, \\ \text{sign}(P'_1[i|j]) = \text{sign}(P_1[i|j]), \text{ otherwise.} \end{cases} \quad (48)$$

In particular, the fact (40) is satisfied for the product $A'_1 A_2 \dots A_{K-1} A'_K$.

- For the sufficiency, it need to show that the fact (41) with (42) holds for all the factors of $A'_1 A_2 \dots A_{K-1} A'_K$. Consider that the fact (41) with (42) has been satisfied for all the factors A_i ($1 \leq i \leq K$). Because of the connections between P_j and P'_j ($j = 1, K$), it remains to show that the fact (42) is satisfied for all $s_j^{(1)}$ ($3 \leq j \leq h$) of P'_j , which is proved as follows: for all $3 \leq j \leq h$, by considering that $\prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1$, we have that

$$\begin{cases} s_j^{(1)} s_j^{(r)} = s_j^{(K)} w_{j-1}^{(1)} w_j^{(1)} s_j^{(r)} = w_{j-1}^{(1)} w_j^{(1)} \prod_{l=r+1}^K w_{j-1}^{(l)} w_j^{(l)} \\ \quad = \prod_{l=2}^r w_{j-1}^{(l)} w_j^{(l)}, \text{ if } s_j^{(r)} \neq 0 \text{ for } r > 1, \\ s_j^{(1)} f_j^{(r)} = s_j^{(K)} w_{j-1}^{(1)} w_j^{(1)} f_j^{(r)} = w_{j-1}^{(1)} w_j^{(1)} \prod_{l=r}^K w_{j-1}^{(l)} w_j^{(l)} \\ \quad = \prod_{l=2}^{r-1} w_{j-1}^{(l)} w_j^{(l)}, \text{ if } f_j^{(r)} \neq 0 \text{ for } r > 1, \\ s_j^{(1)} f_j^{(1)} = s_j^{(K)} w_{j-1}^{(1)} w_j^{(1)} f_j^{(1)} = w_{j-1}^{(1)} w_j^{(1)}. \end{cases}$$

Therefore, we conclude by the induction assumption that the sufficiency is true.

- For the necessity, by the induction assumption, the fact (41) with (42) is true for all the factors of $A'_1 A_2 \dots A_{K-1} A'_K$, and in particular,

$$\begin{cases} P'_K[1|j] = 0, \forall 3 \leq j \leq n; \quad P'_K[1|2] = P_K[1|2]; \\ \text{sign}(P'_K[j|1 : j - 1]) = \text{sign}(P_K[j|1 : j - 1]) = f_j^{(K)}, \quad 2 \leq j \leq n; \\ \text{sign}(P'_K[1 : j - 1|j]) = \text{sign}(P_K[2 : j - 1|j]) = s_j^{(K)}, \quad 3 \leq j \leq h; \\ \text{sign}(P'_K[1 : j - 1|j]) = \text{sign}(P_K[1 : j - 1|j]) = s_j^{(K)}, \quad h + 1 \leq j \leq n; \\ \text{sign}(P'_K[j|j]) = \text{sign}(P_K[j|j]) = w_j^{(K)}, \quad 1 \leq j \leq n, \end{cases}$$

where if $s_j^{(K)} \neq 0$ for some $3 \leq j \leq h$, since $s_j^{(1)} = s_j^{(K)} w_{j-1}^{(1)} w_j^{(1)} \neq 0$ and $s_j^{(1)} s_j^{(K)} = \prod_{l=2}^K w_{j-1}^{(l)} w_j^{(l)}$, we have that $s_j^{(K)} s_j^{(K)} = \prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1$, which together with $\text{sign}(p_{1j}^{(K)}) = s_j^{(K)} \neq 0$ ($3 \leq j \leq h$) implies that

$$\text{sign}(P_K[1 : j - 1|j]) = s_j^{(K)} \neq 0, \quad \forall 3 \leq j \leq h.$$

For showing that the fact (41) with (42) holds for all the factors A_i ($1 \leq i \leq K$), because of the connections between P_j and P'_j ($j = 1, K$), it remains to show that the fact (42) is satisfied for all $s_j^{(K)}$ ($3 \leq j \leq h$) of P_K , which is proved as follows: for all $3 \leq j \leq h$, by considering that $s_j^{(1)} \neq 0$ and $\prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1$, since

$$\begin{cases} s_j^{(1)} s_j^{(r)} = \prod_{l=2}^r w_{j-1}^{(l)} w_j^{(l)}, & \text{if } s_j^{(r)} \neq 0 \text{ for } r \geq 1, \\ s_j^{(1)} f_j^{(r)} = \prod_{l=2}^{r-1} w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for } r > 1, \\ s_j^{(1)} f_j^{(1)} = w_{j-1}^{(1)} w_j^{(1)}, & \text{if } f_j^{(1)} \neq 0, \end{cases}$$

we have that

$$\begin{cases} s_j^{(r)} s_j^{(K)} = \prod_{l=r+1}^K w_{j-1}^{(l)} w_j^{(l)}, & \text{if } s_j^{(r)} \neq 0 \text{ for } r \geq 1, \\ f_j^{(r)} s_j^{(K)} = \prod_{l=r}^K w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for } r \geq 1. \end{cases}$$

Therefore, we conclude that the necessity is true.

The result is proved. □

Remark 2 It must be pointed out that according to the proof of Theorem 5, the condition (40) can be removed for the sufficiency.

Corollary 4 Let $A = A_1 \dots A_{K-1} A_K \in \mathbb{R}^{n \times n}$, where each factor $A_t \in \mathbb{R}^{n \times n}$ ($1 \leq t \leq K$) is a nonsingular CRD matrix whose parameter matrix $P_t \in \mathbb{R}^{n \times n}$ satisfies that

$$P_t[1 : r - 1|r] \neq 0 \text{ and } P_t[r|1 : r - 1] \neq 0, \forall 1 \leq r \leq n.$$

Then Algorithm 2 is subtraction-free to reduce A into a tridiagonal matrix whose basic LR algorithm is subtraction-free if and only if for all $1 \leq t \leq K$,

$$\begin{cases} \text{sign}(P_t[j|1 : j - 1]) = f_j^{(t)}, \\ \text{sign}(P_t[1 : j - 1|j]) = s_j^{(t)}, \quad \forall 1 \leq j \leq n, \\ \text{sign}(P_t[j|j]) = w_j^{(t)}, \end{cases}$$

where

$$\begin{cases} \prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1, \\ f_j^{(t)} s_j^{(t)} = w_{j-1}^{(t)} w_j^{(t)}, \quad \forall 1 \leq t \leq K, \quad \forall 1 \leq j \leq n. \\ s_j^{(t)} = f_j^{(t+1)}, \quad \forall 1 \leq t \leq K - 1; s_j^{(K)} = f_j^{(1)}, \end{cases}$$

Corollary 5 Let $A = A_1 A_2^T \dots A_{2K-1} A_{2K}^T \in \mathbb{R}^{n \times n}$ be the product of Vandermonde matrices $A_t = [(x_i^{(t)})^{j-1}]_{i,j=1}^{n,n}$ ($1 \leq t \leq 2K$) with $0 \geq x_1^{(t)} > x_2^{(t)} > \dots > x_n^{(t)}$. Then Algorithm 2 is subtraction-free to reduce A into a tridiagonal matrix whose basic LR algorithm is subtraction-free.

Proof By the formula (4), the parameter matrix $P_t \in \mathbb{R}^{n \times n}$ of A_t ($1 \leq t \leq 2K$) satisfies that

$$\begin{cases} \text{sign}(P_t[j|1 : j - 1]) = 1, \\ \text{sign}(P_t[1 : j - 1|j]) = -1, \quad \forall 1 \leq j \leq n. \\ \text{sign}(P_t[j|j]) = (-1)^{j-1}, \end{cases}$$

Thus, we conclude by Corollary 4 that the result is true. □

Beside of the example by Corollary 5, more examples can be given for the periodic qd-type reduction method to be subtraction-free. For example, by the formula (4), the parameter matrix of a Vandermonde matrix $A = (x_i^{j-1}) \in \mathbb{R}^{n \times m}$ is nonnegative if $0 \leq x_1 < x_2 < \dots < x_n$; and by the formula (5), the parameter matrix of a Cauchy matrix $A = (\frac{1}{x_i + y_j}) \in \mathbb{R}^{n \times m}$ is positive (or all the diagonal and off-diagonal parameters are negative and positive, respectively) if $x_1 < x_2 < \dots < x_n, y_1 < y_2 < \dots < y_n, x_1 + y_1 > 0$ (or $x_n + y_n < 0$). So, any product consisting of these Vandermonde and Cauchy matrices is our desired example. In what follows, we illustrate how to specifically compute all the eigenvalues of such products to high relative accuracy.

For the product $A = A_1 \dots A_{K-1} A_K \in \mathbb{R}^{n \times n}$ with parameter matrices $P_t = (p_{ij}^{(t)})$ of the factors A_t ($1 \leq t \leq K$) satisfying the fact (41) with (42), the periodic qd-type method is subtraction-free to reduce A into a tridiagonal matrix $T \in \mathbb{R}^{n \times n}$ as follows:

$$T = \bar{A}_1 \dots \bar{A}_{K-1} \bar{A}_K = LDU, \tag{49}$$

where let $\bar{P}_t = (\bar{p}_{ij}^{(t)})$ be the parameter matrix of \bar{A}_t ($1 \leq t \leq K$), then $D = \text{diag}(d_{ii})$ with $d_{ii} = \prod_{l=1}^K \bar{p}_{ii}^{(l)}$ and $\text{sign}(\bar{p}_{ii}^{(l)}) = \text{sign}(p_{ii}^{(l)})$ for all i and l , L and U are lower and upper bidiagonal with unit diagonal whose bidiagonal entries $l_i = \bar{p}_{i,i-1}^{(1)}$ and $u_i = \bar{p}_{i-1,i}^{(K)}$, respectively; and $\text{sign}(l_i u_i) = 1$ or 0 for all i . Consequently, let $X = \text{diag}(x_1, \prod_{i=1}^2 x_i, \dots, \prod_{i=1}^n x_i) \in \mathbb{R}^{n \times n}$, where

$$x_i = \begin{cases} \max\{\text{sign}(l_i), \text{sign}(u_i)\}, & \text{if } l_i \neq 0 \text{ or } u_i \neq 0, \\ 1, & \text{otherwise,} \end{cases} \quad 1 \leq i \leq n,$$

then by the fact $\prod_{l=1}^K \text{sign}(\bar{p}_{i-1,i-1}^{(l)} \bar{p}_{ii}^{(l)}) = \prod_{l=1}^K \text{sign}(p_{i-1,i-1}^{(l)} p_{ii}^{(l)}) = 1$ for all i ,

$$X^{-1} T X = \begin{cases} |L||D||U|, & \text{if } \prod_{l=1}^K \text{sign}(p_{11}^{(l)}) = 1, \\ -|L||D||U|, & \text{if } \prod_{l=1}^K \text{sign}(p_{11}^{(l)}) = -1, \end{cases}$$

here, $|\cdot|$ is interpreted componentwise. Further, $|L||D||U|$ has the same eigenvalues as those of the symmetric tridiagonal matrix

$$T' = (D'U')^T (D'U') = B^T B, \quad D' = \text{diag}(\sqrt{|d_{ii}|}), \tag{50}$$

where U' is bidiagonal with unit diagonal whose $(i-1, i)$ th entries $u'_i = \sqrt{|\bar{p}_{i,i-1}^{(1)} \bar{p}_{i-1,i}^{(K)}|}$ ($2 \leq i \leq n$). So, all the eigenvalues of A are positive or negative squares of singular values of B according to $\prod_{l=1}^K \text{sign}(p_{11}^{(l)}) = 1$ or -1 . It is well known that all the singular values of a bidiagonal matrix is accurately computed by the dqds algorithm [13]. Therefore, we have Algorithm 3 to compute eigenvalues of the product satisfying (41) with (42).

5.4 Error Analysis

In this subsection, error analysis is provided to illustrate the high relative accuracy of Algorithm 3. We first show that for the product (19) satisfying the fact (41) with (42), all its eigenvalues are determined by its parameters to high relative accuracy.

Theorem 6 *Let $A = A_1 \dots A_{K-1} A_K \in \mathbb{R}^{n \times n}$ be the product of nonsingular CRD matrices $A_t \in \mathbb{R}^{n \times n}$ ($1 \leq t \leq K$) whose parameter matrices $P_t = (p_{ij}^{(t)}) \in \mathbb{R}^{n \times n}$ satisfy the fact (41)*

Algorithm 3 The algorithm computes eigenvalues of $A = A_1 \dots A_{K-1} A_K \in \mathbb{R}^{n \times n}$ with parameter matrices $P_t = (p_{ij}^{(t)})$ of the factors A_t ($1 \leq t \leq K$) satisfying (41) with (42).

- 1: First, A is reduced by Algorithm 2 into the tridiagonal matrix T of (49).
- 2: Compute singular values of the bidiagonal B of (50) by the dqds algorithm. Then all the eigenvalues of A' are obtained as the positive or negative squares of these singular values according to $\prod_{l=1}^K \text{sign}(p_{11}^{(l)}) = 1$ or -1 .

with (42), and let $\tilde{A} \in \mathbb{R}^{n \times n}$ be obtained from A by replacing one of these parameters $p_{ij}^{(t)}$ with $\tilde{p}_{ij}^{(t)} = p_{ij}^{(t)}(1 + \epsilon_{ij}^{(t)})$ ($1 \leq i, j \leq n$ and $1 \leq t \leq K$), where $|\epsilon_{ij}^{(t)}| \leq \epsilon$ and $2\epsilon < 1$. Then for the descending-ordered eigenvalues λ_i and $\tilde{\lambda}_i$ of A and \tilde{A} , respectively;

$$|\tilde{\lambda}_i - \lambda_i| \leq \frac{2\epsilon}{1 - 2\epsilon} |\lambda_i|, \quad \forall 1 \leq i \leq n.$$

Proof According to the fact (41), let $X = \text{diag}(x_1, \prod_{j=1}^2 x_j, \dots, \prod_{j=1}^n x_j) \in \mathbb{R}^{n \times n}$, where for all $1 \leq j \leq n$,

$$x_j = \begin{cases} f_j^{(r)} \prod_{l=1}^{r-1} w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(r)} \neq 0 \text{ for some } 1 \leq r \leq K; \\ s_j^{(r)} \prod_{l=1}^r w_{j-1}^{(l)} w_j^{(l)}, & \text{if } f_j^{(l)} = 0 \text{ for all } 1 \leq l \leq K, \text{ but } s_j^{(r)} \neq 0 \text{ for some } 1 \leq r \leq K; \\ 1, & \text{otherwise;} \end{cases}$$

and thus, let $Z_t = XY_t = \text{diag}(z_j^{(t)}) \in \mathbb{R}^{n \times n}$ where $Y_t = \text{diag}(\prod_{l=1}^t w_j^{(l)}) \in \mathbb{R}^{n \times n}$ for all $0 \leq t \leq K$ with the convention $Y_0 = I_n$. Then

$$Z_0 A Z_K = (Z_0 A_1 Z_1)(Z_1 A_2 Z_2) \dots (Z_{K-1} A_K Z_K) = \tilde{A}_1 \tilde{A}_2 \dots \tilde{A}_K,$$

where for each $1 \leq t \leq K$, since $A_t = B_1 \dots B_{n-1} D C_{n-1} \dots C_1$ is as in (1) satisfying the fact (41), we have

$$\begin{aligned} \tilde{A}_t &= Z_{t-1} A_t Z_t = |B_1| \dots |B_{n-1}| (Z_{t-1} D) C_{n-1} \dots C_1 Z_t \\ &= |B_1| \dots |B_{n-1}| |D| (Z_t C_{n-1} \dots C_1) Z_t \\ &= |B_1| \dots |B_{n-1}| |D| |C_{n-1}| \dots |C_1| (Z_t Z_t) \\ &= |B_1| \dots |B_{n-1}| |D| |C_{n-1}| \dots |C_1|, \end{aligned}$$

because the following statements hold by using the fact (42):

- first, for any $f_j^{(t)} \neq 0$ ($2 \leq j \leq n$),

$$f_j^{(t)} z_{j-1}^{(t-1)} z_j^{(t-1)} = f_j^{(t)} x_j \prod_{l=1}^{t-1} w_{j-1}^{(l)} w_j^{(l)} = f_j^{(t)} \left(f_j^{(r)} \prod_{l=1}^{r-1} w_{j-1}^{(l)} w_j^{(l)} \right) \prod_{l=1}^{t-1} w_{j-1}^{(l)} w_j^{(l)} = 1,$$

such that

$$Z_{t-1} B_i Z_{t-1} = |B_i|, \quad \forall 1 \leq i \leq n - 1;$$

- further, for any $w_j^{(t)}$ ($1 \leq j \leq n$),

$$w_j^{(t)} z_j^{(t-1)} = z_j^{(t)}, \quad \forall 1 \leq j \leq n$$

such that

$$Z_{t-1} D = |D| Z_t;$$

– finally, for any $s_j^{(t)} \neq 0$ ($2 \leq j \leq n$),

$$s_j^{(t)} z_{j-1}^{(t)} z_j^{(t)} = s_j^{(t)} x_j \prod_{l=1}^t w_{j-1}^{(l)} w_j^{(l)} = s_j^{(t)} \left(s_j^{(r)} \prod_{l=1}^r w_{j-1}^{(l)} w_j^{(l)} \right) \prod_{l=1}^t w_{j-1}^{(l)} w_j^{(l)} = 1,$$

such that

$$Z_t C_i Z_t = |C_i|, \forall 1 \leq i \leq n - 1.$$

The fact $\prod_{l=1}^K w_{j-1}^{(l)} w_j^{(l)} = 1$ implies that $\prod_{l=1}^K w_{j-1}^{(l)} = \prod_{l=1}^K w_j^{(l)}$ for all j . So, $Z_k = (\prod_{l=1}^K w_1^{(l)}) \cdot X$, and thus,

$$X^{-1} A X = \left(\prod_{l=1}^K w_1^{(l)} \right) \cdot \bar{A}_1 \bar{A}_2 \dots \bar{A}_K = \pm (\bar{A}_1 \bar{A}_2 \dots \bar{A}_K)$$

where each \bar{A}_t ($1 \leq t \leq K$) is a nonsingular CRD matrix whose parameter matrix $\bar{P}_t = (|p_{ij}^{(t)}|) \in \mathbb{R}^{n \times n}$. This means that $\bar{A}_1 \bar{A}_2 \dots \bar{A}_K$ is just a product of nonnegative bidiagonal matrices. Therefore, we conclude by [19, Theorem 7.2] that the result is true. \square

Now we are ready to show the high relative accuracy of Algorithm 3 as follows.

Theorem 7 *Let $A = A_1 \dots A_{K-1} A_K \in \mathbb{R}^{n \times n}$ be the product of nonsingular CRD matrices $A_t \in \mathbb{R}^{n \times n}$ ($1 \leq t \leq K$) satisfying the fact (41) with (42), and let λ_i and $\hat{\lambda}_i$ be its descending-ordered exact and computed eigenvalues by Algorithm 3, respectively. Then*

$$|\hat{\lambda}_i - \lambda_i| \leq \frac{O(2Kn^3)\mu}{1 - O(2Kn^3)\mu} |\lambda_i|, \quad i = 1, 2, \dots, n.$$

Proof For the first stage of Algorithm 3, let T and \hat{T} be the exact and computed tridiagonal forms obtained from A without any subtractions of like-signed numbers in $O(Kn^3)$

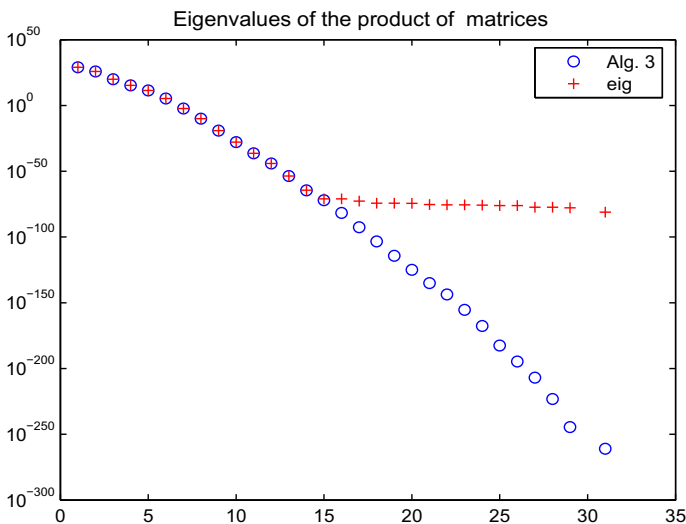


Fig. 1 The computed eigenvalues of the product in Example 1

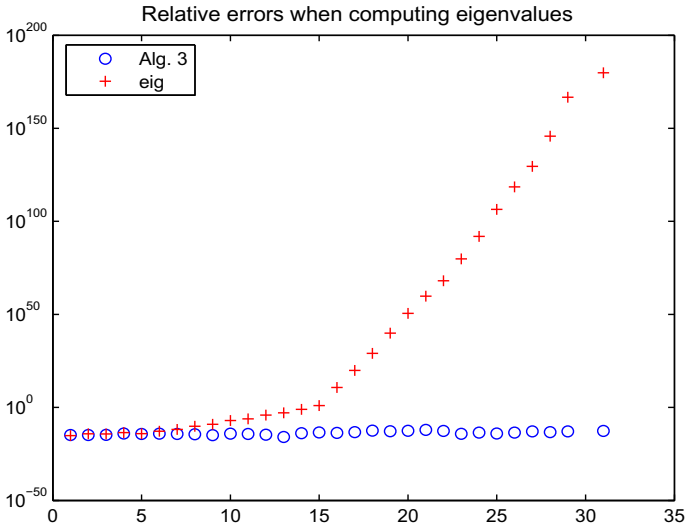


Fig. 2 The relative errors for the computed eigenvalues of the product in Example 1

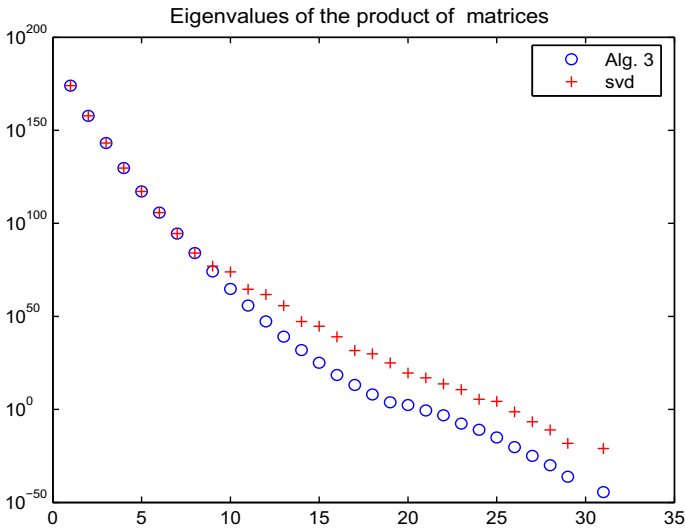


Fig. 3 The computed eigenvalues of the product in Example 2

arithmetic operations, respectively. Consider that every single subtraction-free arithmetic operation causes at most μ relative perturbation in at most one parameter of the factors of A . So, by Theorem 6, all the descending-ordered eigenvalues $\lambda_i(\hat{T})$ and $\lambda_i(T)$ satisfy that

$$|\lambda_i(\hat{T}) - \lambda_i(T)| \leq \frac{O(2Kn^3)\mu}{1 - O(2Kn^3)\mu} |\lambda_i(T)|, \quad \lambda_i(T) = \lambda_i(A), \quad i = 1, 2, \dots, n.$$

Further, for the second stage of Algorithm 3, let B and \hat{B} be the exact and computed bidiagonal matrices obtained from \hat{T} without any subtractions of like-signed numbers, respectively; then

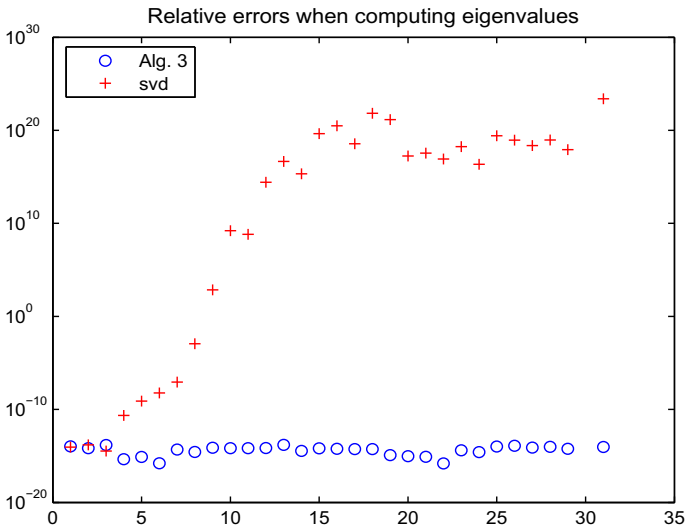


Fig. 4 The relative errors for the computed eigenvalues of the product in Example 2

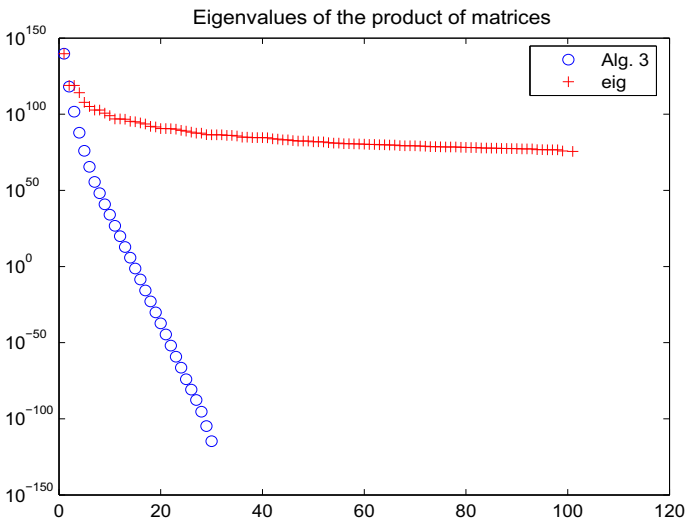


Fig. 5 The computed eigenvalues of the product in Example 3

by [8, Theorem 2], all the descending-ordered singular values $\sigma_i(\hat{B})$ and $\sigma_i(B)$ satisfy that

$$|\sigma_i(\hat{B}) - \sigma_i(B)| \leq \frac{O(n^3)\mu}{1 - O(n^3)\mu} \sigma_i(B), \quad \sigma_i^2(B) = |\lambda_i(\hat{T})|, \quad i = 1, 2, \dots, n.$$

In addition, the dqds algorithm computes each singular value of \hat{B} with a relative error not exceeding $O(n^2)\mu$ [13, 25]. Therefore, we conclude that the result is true. \square

Table 1 The relative errors for computing nonzero eigenvalues in Example 3

i	λ_i	$\frac{ \hat{\lambda}_i - \lambda_i }{ \lambda_i }$ by Algorithm 3	$\frac{ \hat{\lambda}_i - \lambda_i }{ \lambda_i }$ by eig
1	-4.8724e+139	8.4661e-015	4.3416e-016
2	-1.2781e+118	7.3602e-015	6.4850e+000
3	-4.8453e+101	3.0792e-015	1.7106e+017
4	-7.3031e+087	5.6854e-015	2.0035e+026
5	-8.6151e+075	6.1553e-015	8.0932e+031
6	-2.8816e+065	3.4083e-015	4.3360e+039
7	-3.7381e+055	3.9325e-015	1.4411e+047
8	-1.4449e+048	1.1230e-016	3.6239e+054
9	-6.9586e+040	4.4475e-015	6.6380e+059
10	-1.0817e+034	2.9844e-015	9.6114e+064
11	-5.4507e+026	4.1605e-015	1.5851e+070
12	-8.3721e+019	5.6752e-015	1.0319e+077
13	-6.4044e+012	1.3723e-015	7.1694e+083
14	-5.5881e+005	4.5832e-015	3.8750e+089
15	-4.3688e-002	3.4942e-015	2.6230e+096
16	-3.1274e-009	6.6123e-015	5.6140e+102
17	-2.0928e-016	4.3584e-015	1.8387e+109
18	-1.3072e-023	6.6319e-015	8.8989e+114
19	-7.7185e-031	6.4678e-015	5.2738e+121
20	-4.2582e-038	5.8844e-015	9.4070e+127
21	-2.3239e-045	4.5523e-015	1.7237e+135
22	-1.1738e-052	4.4242e-015	2.7974e+142
23	-5.5002e-060	2.2108e-015	2.7510e+149
24	-3.5171e-067	3.3721e-015	6.6886e+155
25	-8.5208e-075	4.7242e-015	1.0864e+163
26	-1.5318e-081	1.0697e-015	1.0389e+169
27	-1.9844e-088	1.2656e-015	1.9823e+175
28	-4.3120e-096	1.7358e-015	8.9878e+182
29	-1.6576e-105	3.5045e-016	2.3786e+191
30	-1.8062e-115	2.2464e-015	1.8034e+201

6 Numerical Experiments

In this section, we provide numerical experiments to confirm the high relative accuracy of our proposed method by measuring the relative error $|\hat{\lambda}_i - \lambda_i|/|\lambda_i|$ of the computed eigenvalues $\hat{\lambda}_i$, where λ_i is the exact descending-ordered eigenvalue by Mathematica with 200-decimal digit arithmetic. All the tests are conducted by MATLAB 7.0 in double precision arithmetic.

Example 1 Consider the product $A = B^6$ with $B = \left[\frac{n^2}{x_i + y_j} \right]_{i,j=1}^{n,n}$, where the descending-ordered vectors $x = (x_i), y = (y_j) \in \mathbb{R}^{1 \times n}$ are randomly chosen by the command rand. Set $n = 30$. The spectral condition number $k_2(B) = 7.3453e + 049$ by Mathematica. We compute eigenvalues of A by Algorithm 3 and Matlab command $(\text{eig}(B))^6$, respectively. The absolute values and relative errors of the computed eigenvalues are plotted in Figs. 1

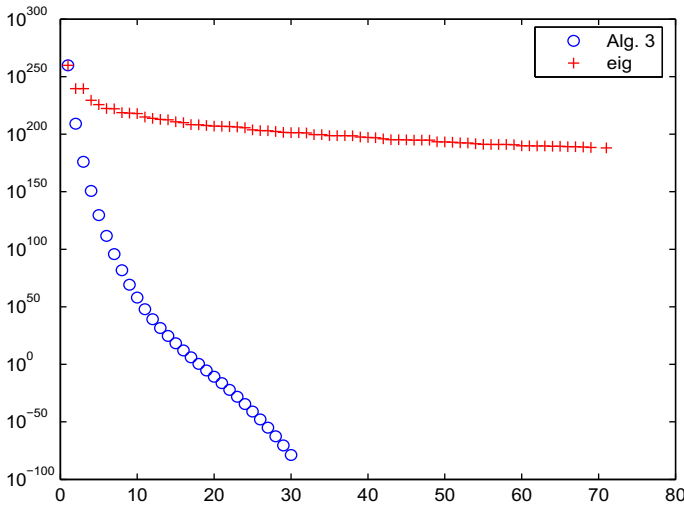


Fig. 6 The computed eigenvalues of the product in Example 4

and 2, respectively; where the maximum relative errors by Algorithm 3 and $(\text{eig}(B))^6$ are $8.6766e - 013$ and $6.5770e + 179$, respectively.

Example 2 Consider the product $A = (BB^T)^3$ with $B = [(-x_i \cdot 10)^{j-1}]_{i,j=1}^{n,m}$, where the ascending-ordered vector $x = (x_i) \in \mathbb{R}^{1 \times n}$ is randomly chosen by the command `rand`. Set $n = 100$ and $m = 30$. Then $k_2(B) = 2.5298e + 036$ by Mathematica. We compute eigenvalues of A by Algorithm 3 and Matlab command $(\text{svd}(B))^6$, respectively. The absolute values and relative errors of the computed eigenvalues are plotted in Figs. 3 and 4, respectively; where the maximum relative errors by Algorithm 3 and $(\text{svd}(B))^6$ are $1.5229e - 014$ and $2.4737e + 023$, respectively.

Example 3 Consider the product $A = A_1 A_2 A_3 A_4 A_5 A_6 \in \mathbb{R}^{n_1 \times n_1}$, where

$$A_1 = \left[\left(\frac{1}{n_1 - i + 1} \right)^{j-1} \right]_{i,j=1}^{n_1, n_2}, \quad A_2 = \left[\left(\frac{-i^2}{n_2} \right)^{j-1} \right]_{i,j=1}^{n_2, n_3}, \quad A_3 = \left[(-n_4 - j)^{i-1} \right]_{i,j=1}^{n_3, n_4},$$

and

$$A_4 = \left[\frac{n_4}{i + j - 1} \right]_{i,j=1}^{n_4, n_5}, \quad A_5 = \left[\frac{n_6}{-i - j} \right]_{i,j=1}^{n_5, n_6}, \quad A_6 = \left[\frac{n_1}{i + j} \right]_{i,j=1}^{n_6, n_1}.$$

By the formulas (4) and (5), the fact (41) with (42) is satisfied. Set $n_1 = 100$, $n_2 = 30$, $n_3 = 40$, $n_4 = 50$, $n_5 = 60$, $n_6 = 70$. Then $k_2(A_1) = 1.7796e + 042$, $k_2(A_2) = 1.9307e + 060$, $k_2(A_3) = 4.0401e + 109$, $k_2(A_4) = 3.9278e + 069$, $k_2(A_5) = 1.5132e + 085$ and $k_2(A_6) = 1.4911e + 095$ by Mathematica. Notice that 70 eigenvalues of A are zero. We compute eigenvalues of A by Algorithm 3 and Matlab command $\text{eig}(A)$, respectively. The absolute values of all the computed eigenvalues are plotted in Fig. 5. As observed, these 70 zero eigenvalues are not correctly computed by $\text{eig}(A)$. The relative errors for computing 30 nonzero eigenvalues are reported in Table 1, which confirms that all the eigenvalues of A are computed by Algorithm 3 to high relative accuracy.

Table 2 The relative errors for computing nonzero eigenvalues in Example 4

i	λ_i	$\frac{ \hat{\lambda}_i - \lambda_i }{ \lambda_i }$ by Algorithm 3	$\frac{ \hat{\lambda}_i - \lambda_i }{ \lambda_i }$ by eig
1	6.6697e+259	6.9205e-014	0
2	1.1167e+209	8.0081e-015	3.1584e+030
3	1.0489e+176	1.4342e-014	3.3624e+063
4	4.2646e+150	6.8175e-016	6.2668e+078
5	4.4801e+129	1.9239e-015	1.1570e+096
6	4.3630e+111	8.5675e-016	4.6432e+110
7	6.7572e+095	1.1757e-014	1.7433e+126
8	5.9600e+081	5.2303e-015	8.8005e+136
9	1.5994e+069	1.0779e-014	1.4047e+149
10	8.9595e+057	7.4672e-015	9.3269e+159
11	8.7307e+047	2.4160e-015	8.7218e+166
12	1.4460e+039	4.1803e-016	4.5678e+174
13	3.0661e+031	3.2314e-015	2.5934e+181
14	4.1342e+024	3.2465e-015	7.3640e+187
15	1.8062e+018	7.0866e-016	3.0872e+192
16	1.3232e+012	2.5831e-015	7.1353e+197
17	1.4289e+006	5.0514e-015	1.3590e+202
18	2.2126e+000	8.0284e-016	7.7413e+207
19	5.0581e-006	2.1770e-015	5.9630e+212
20	1.6060e-011	4.6275e-015	6.2198e+217
21	4.3162e-017	3.8552e-015	1.5661e+223
22	6.9029e-023	1.5326e-015	6.2890e+228
23	6.3271e-029	2.1262e-015	2.5401e+234
24	3.1936e-035	2.6781e-015	1.3179e+240
25	8.5503e-042	0	6.1700e+244
26	1.1847e-048	7.6948e-016	9.8083e+250
27	8.2274e-056	0	1.2125e+258
28	2.6994e-063	7.7982e-015	7.6503e+264
29	3.7202e-071	1.5134e-015	9.7380e+271
30	1.6077e-079	2.0501e-015	1.1712e+280

Example 4 Consider the product $A = A_1 A_2 A_3 A_4 A_5 A_6 \in \mathbb{R}^{n_1 \times n_1}$, where

$$A_1 = \left[\left(\frac{-1}{n_1 - i + 1} \right)^{j-1} \right]_{i,j=1}^{n_1, n_2}, A_2 = [(-n_3 - j)^{i-1}]_{i,j=1}^{n_2, n_3}, A_3 = \left[\left(\frac{-1}{n_3 - i + 1} \right)^{j-1} \right]_{i,j=1}^{n_3, n_4},$$

and

$$A_4 = [(-n_5 - j)^{i-1}]_{i,j=1}^{n_4, n_5}, A_5 = \left[\left(\frac{-1}{n_5 - i + 1} \right)^{j-1} \right]_{i,j=1}^{n_5, n_6}, A_6 = [(-n_1 - j)^{i-1}]_{i,j=1}^{n_6, n_1}.$$

By the formula (4), the fact (41) with (42) is satisfied. Set $n_1 = 70, n_2 = 30, n_3 = 40, n_4 = 40, n_5 = 50, n_6 = 60$. Then $k_2(A_1) = 8.3931e + 042, k_2(A_2) = 1.9573e + 078, k_2(A_3) = 9.2919e + 068, k_2(A_4) = 4.0401e + 109, k_2(A_5) = 6.2048e + 089$ and $k_2(A_6) =$

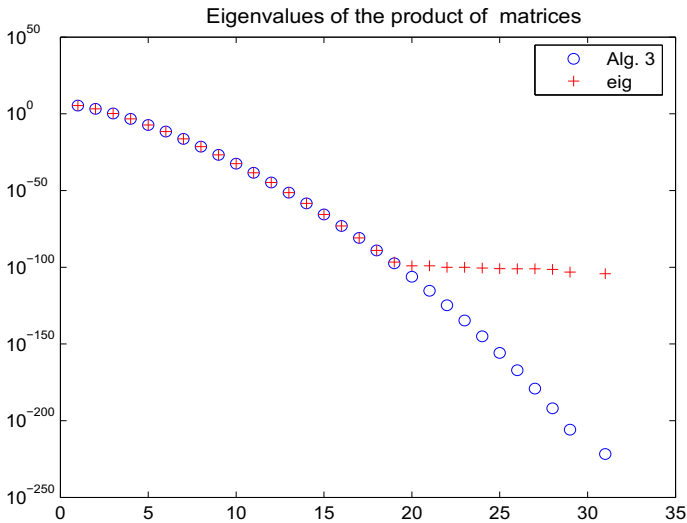


Fig. 7 The computed eigenvalues of the product in Example 5

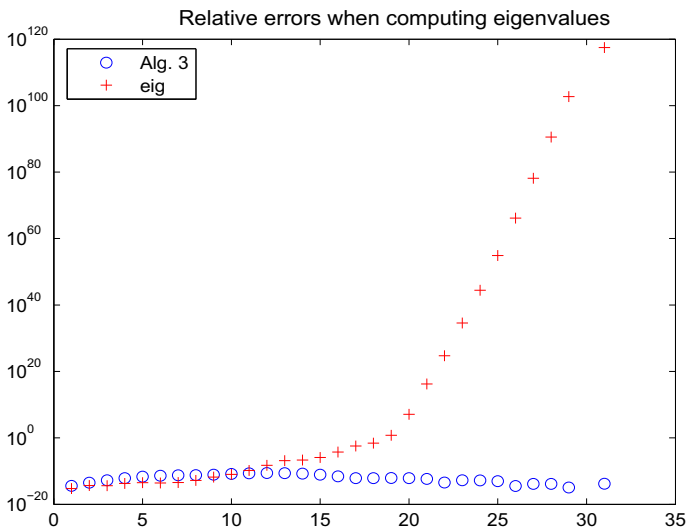


Fig. 8 The relative errors for the computed eigenvalues of the product in Example 5

$3.0557e + 175$ by Mathematica. Notice that 40 eigenvalues of A are zero. We compute eigenvalues of A by Algorithm 3 and Matlab command `eig(A)`, respectively. The absolute values of all the computed eigenvalues are plotted in Fig. 6. As observed, these 40 zero eigenvalues are not correctly computed by `eig(A)`. The relative errors for computing 30 nonzero eigenvalues are reported in Table 2, which confirms the high relative accuracy of Algorithm 3.

Finally, we show the accuracy of our method even though the fact (41) with (42) is not satisfied.

Example 5 Consider the product $A = B^6 \in \mathbb{R}^{n \times n}$ with $B = \left[\left(\frac{n}{n+i} \right)^{j-1} \right]_{i,j=1}^{n,n}$. Set $n = 30$. Then $k_2(B) = 7.9202e + 037$ by Mathematica. The fact (41) with (42) is not satisfied for A . Nevertheless, Algorithm 3 computes eigenvalues of A with more accuracy than those by the command $(\text{eig}(B))^6$, especially for eigenvalues with small magnitudes. All the absolute values and relative errors of the computed eigenvalues are plotted in Figs. 7 and 8, respectively. The maximum relative error by Algorithm 3 is $2.5882e - 011$, whereas the maximum relative error by $(\text{svd}(B))^6$ is $3.1404e + 117$.

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

References

1. Alfa, A.S., Xue, J., Ye, Q.: Accurate computation of the smallest eigenvalue of a diagonally dominant M-matrix. *Math. Comput.* **71**, 217–236 (2002)
2. Barlow, J., Demmel, J.: Computing accurate eigensystems of scaled diagonally dominant matrices. *SIAM J. Numer. Anal.* **27**, 762–791 (1990)
3. Bevilacqua, R., Bozzo, E., Corso, G.M.D.: qd-Type methods for quasiseparable matrices. *SIAM J. Matrix Anal. Appl.* **32**, 722–747 (2011)
4. Chu, D., Lathauwer, L.D., Moor, B.D.: A QR-type reduction for computing the SVD of a general matrix product/quotient. *Numer. Math.* **95**, 101–121 (2003)
5. Demmel, J.: Accurate singular value decompositions of structured matrices. *SIAM J. Matrix Anal. Appl.* **21**, 562–580 (1999)
6. Demmel, J., Gragg, W.: On computing accurate singular values and eigenvalues of acyclic matrices. *Linear Algebra Appl.* **185**, 203–218 (1993)
7. Demmel, J., Ioana, D., Holtz, O., Koev, P.: Accurate and efficient expression evaluation and linear algebra. *Acta Numer.* **17**, 87–145 (2008)
8. Demmel, J., Kahan, W.: Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Stat. Comput.* **11**, 873–912 (1990)
9. Dhillon, I.S., Parlett, B.N.: Orthogonal eigenvectors and relative gaps. *SIAM J. Matrix Anal. Appl.* **25**, 858–899 (2004)
10. Dopico, F.M., Koev, P.: Accurate symmetric rank revealing and eigendecompositions of symmetric structured matrices. *SIAM J. Matrix Anal. Appl.* **28**, 1126–1156 (2006)
11. Dopico, F.M., Koev, P., Molera, J.M.: Implicit standard Jacobi Givens high relative accuracy. *Numer. Math.* **113**, 519–553 (2009)
12. Dopico, F.M., Molera, J.M., Moro, J.: An orthogonal high relative accuracy algorithm for the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.* **25**, 301–351 (2003)
13. Fernando, K., Parlett, B.: Accurate singular values and differential qd algorithms. *Numer. Math.* **67**, 191–229 (1994)
14. Golub, G.H., Sölna, K., Dooren, P.V.: Computing the SVD of a general matrix product/quotient. *SIAM J. Matrix Anal. Appl.* **22**, 1–19 (2000)
15. Heath, M.T., Laub, A.J., Paige, C.C., Ward, R.C.: Computing the singular value decomposition of a product of two matrices. *SIAM J. Sci. Stat. Comput.* **7**, 1147–1159 (1986)
16. Huang, R.: Rank structure properties of rectangular matrices admitting bidiagonal-type factorizations. *Linear Algebra Appl.* **465**, 1–14 (2015)
17. Huang, R., Chu, D.L.: Relative perturbation analysis for eigenvalues and singular values of totally non-positive matrices. *SIAM J. Matrix Anal. Appl.* **36**, 476–495 (2015)
18. Kaufman, L.: The LZ-algorithm to solve the generalized eigenvalue problem. *SIAM J. Numer. Anal.* **11**, 997–1024 (1974)
19. Koev, P.: Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **27**, 1–23 (2005)
20. Koev, P., Dopico, F.M.: Accurate eigenvalues of certain sign regular matrices. *Linear Algebra Appl.* **424**, 435–447 (2007)
21. Moler, C.B., Stewart, G.W.: An algorithm for generalized matrix eigenvalue problems. *SIAM J. Numer. Anal.* **10**, 241–256 (1973)

22. Peláez, M.J., Moro, J.: Accurate factorization and eigenvalues algorithms for symmetric DSTU and TSC matrices. *SIAM J. Matrix Anal. Appl.* **28**, 1173–1198 (2006)
23. Pan, V.Y.: *Structured Matrices and Polynomials: Unified Superfast Algorithms*. Birkhäuser Boston Inc., Boston, MA (2001)
24. Pan, V.Y.: How bad are Vandermonde matrices. *SIAM J. Matrix Anal. Appl.* **27**, 676–694 (2016)
25. Parlett, B.: The new qd algorithms. *Acta Numer.* **17**, 459–491 (1995)
26. Phillips, G.M.: *Interpolation and Approximation by Polynomials*. Springer, New York (2003)
27. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. Cambridge University Press, Cambridge (2007)
28. Stewart, D.: 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)
29. Watkins, D.S.: Product eigenvalue problems. *SIAM Rev.* **47**, 3–40 (2005)
30. Watkins, D.S.: *The Matrix Eigenvalue Problem: GR and KrylovSubspace Methods*. SIAM, Philadelphia (2007)