A Cross-Product Free Jacobi–Davidson Type Method for Computing a Partial Generalized Singular Value Decomposition of a Large Matrix Pair

Jinzhi Huang · Zhongxiao Jia

Received: 7 April 2022 / Revised: 27 October 2022 / Accepted: 30 October 2022 / Published online: 23 November 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
A cross-product free (CPF) Jacobi–Davidson type method is proposed to compute a partial generalized singular value decomposition (GSVD) of a large regular matrix pair \(\{A, B\}\), called CPF-JDGSVD. It implicitly solves the mathematically equivalent generalized eigenvalue problem of the cross-product matrix pair \(\{A^T A, B^T B\}\) using the Rayleigh–Ritz projection method but does not form the cross-product matrices explicitly, and thus avoids the possible accuracy loss of the computed generalized singular values and generalized singular vectors. The method is an inner-outer iteration method, where the expansion of the right searching subspace forms the inner iterations that approximately solve the correction equations involved and the outer iterations extract approximate GSVD components with respect to the subspaces. A convergence result is established for the outer iterations, compact bounds are derived for the condition numbers of the correction equations, and the least solution accuracy requirements on the inner iterations are found, which can maximize the overall efficiency of CPF-JDGSVD as much as possible. Based on them, practical stopping criteria are designed for the inner iterations. A thick-restart CPF-JDGSVD algorithm with deflation and purgation is developed to compute several GSVD components of \(\{A, B\}\) associated with the generalized singular values closest to a given target \(\tau\). Numerical experiments illustrate the efficiency of the algorithm.

Keywords Generalized singular value decomposition · Generalized singular value · Generalized singular vector · Extraction approach · Subspace expansion · Correction equation · Inner iteration · Outer iteration · Deflation · Purgation

J. Huang and Z. Jia have contributed equally to this work. The work of Jinzhi Huang and Zhongxiao Jia was supported by the Youth Program of the Natural Science Foundation of Jiangsu Province (No. BK20220482) and the National Science Foundation of China (No. 12171273), respectively.

Zhongxiao Jia jiazx@tsinghua.edu.cn
Jinzhi Huang jzhuang21@suda.edu.cn

1 School of Mathematical Sciences, Soochow University, 215006 Suzhou, China
2 Department of Mathematical Sciences, Tsinghua University, 100084 Beijing, China
Mathematics Subject Classification 65F15 · 15A18 · 15A12 · 65F10

1 Introduction

The generalized singular value decomposition (GSVD) of a matrix pair is first introduced by Van Loan [30] and then developed by Paige and Saunders [21]. It has become an important analysis means and computational tool [6], and has been used extensively in, e.g., solutions of discrete linear ill-posed problems [9], weighted or generalized least squares problems [3], information retrieval [11], linear discriminant analysis [22], and many others [2, 4, 6, 20, 29].

Let \( A \in \mathbb{R}^{m \times n} \) and \( B \in \mathbb{R}^{p \times n} \) with \( m \geq n \) be large matrices, and throughout the paper assume that the stacked matrix \( \begin{bmatrix} A \\ B \end{bmatrix} \) has full column rank, i.e., \( \mathcal{N}(A) \cap \mathcal{N}(B) = \{0\} \) with \( \mathcal{N}(A) \) and \( \mathcal{N}(B) \) the null spaces of \( A \) and \( B \), respectively. Then such matrix pair \( \{A, B\} \) is called \textit{regular}. Denote \( q_1 = \dim(\mathcal{N}(A)) \) and \( q_2 = \dim(\mathcal{N}(B)) \), \( q = n - q_1 - q_2 \), and \( l_1 = \dim(\mathcal{N}(A^T)) \) and \( l_2 = \dim(\mathcal{N}(B^T)) \). Then the GSVD of the regular matrix pair \( \{A, B\} \) is

\[
\begin{align*}
U^T AX &= \Sigma_A = \text{diag}\{C_0, 0_{l_1, q_1}, I_{q_2}\}, \\
V^T BX &= \Sigma_B = \text{diag}\{S_0, I_{q_1}, 0_{l_2, q_2}\},
\end{align*}
\]

where \( X = [X_1, X_2, X_3] \) is nonsingular, \( U = [U_1, U_2, U_3] \) and \( V = [V_1, V_2, V_3] \) are orthogonal, and \( C = \text{diag}\{\sigma_1, \ldots, \sigma_q\} \) and \( S = \text{diag}\{\beta_1, \ldots, \beta_q\} \) are diagonal matrices that satisfy

\[
0 < \alpha_i, \beta_i < 1 \quad \text{and} \quad \alpha_i^2 + \beta_i^2 = 1, \quad i = 1, \ldots, q;
\]

see [21]. Here, in order to distinguish the block submatrices of \( X, U, V \), we have used the subscripts to denote their column and row numbers, and have denoted by \( I_k \) and \( 0_{k,l} \) the identity matrix of order \( k \) and zero matrix of order \( k \times l \), respectively. The subscripts of identity and zero matrices will be omitted in the sequel when their orders are clear from the context. It follows from (1.1) that \( X^T (A^T A + B^T B) X = I_q \), i.e., the columns of \( X \) are \( (A^T A + B^T B) \)-orthonormal.

The GSVD components \( \{0_{l_1, q_1}, I_{q_1}, U_{l_1}, V_{l_1}, X_{q_1}\} \) and \( \{I_{q_2}, 0_{l_2, q_2}, U_{q_2}, V_{q_2}, X_{q_2}\} \) are associated with the zero and infinite generalized singular values of \( \{A, B\} \), called the \textit{trivial} ones, and the columns of \( U_{l_1}, V_{l_2} \) and \( X_{q_1}, X_{q_2} \) form orthonormal and \( (A^T A + B^T B) \)-orthonormal bases of \( \mathcal{N}(A^T) \) and \( \mathcal{N}(B^T) \), respectively. Denote by \( u_i, v_i \) and \( x_i \) the \( i \)-th columns of \( U_q, V_q \) and \( X_q \), respectively, \( i = 1, \ldots, q \). Then the quintuple \( (\alpha_i, \beta_i, u_i, v_i, x_i) \) is called a \textit{nontrivial GSVD component} of \( \{A, B\} \) with the generalized singular value \( \sigma_i = \frac{\alpha_i}{\beta_i} \), the left generalized singular vectors \( u_i, v_i \) and the right generalized singular vector \( x_i \). We also refer to a pair \( (\alpha_i, \beta_i) \) as a generalized singular value of \( \{A, B\} \).

For a given target \( \tau > 0 \), assume that the nontrivial generalized singular values \( \sigma_i, i = 1, 2, \ldots, q \) of \( \{A, B\} \) are labeled as

\[
|\sigma_1 - \tau| \leq |\sigma_2 - \tau| \leq \cdots \leq |\sigma_{\ell} - \tau| < |\sigma_{\ell+1} - \tau| \leq \cdots \leq |\sigma_q - \tau|.
\]  

(1.2)

We are interested in computing the \( \ell \) GSVD components \( (\alpha_i, \beta_i, u_i, v_i, x_i) \) corresponding to the generalized singular values closest to \( \tau \). If \( \tau \) is close to one of the ends of the nontrivial generalized singular spectrum, they are called the\textit{extreme}, i.e., largest or smallest, ones. If \( \tau \) is inside the nontrivial generalized singular spectrum of \( \{A, B\} \), then \( (\alpha_i, \beta_i, u_i, v_i, x_i) \), \( i = 1, \ldots, \ell \), are called \textit{interior GSVD components} of \( \{A, B\} \). Without loss of generality, we always assume that \( \tau \) is not equal to any generalized singular value of \( \{A, B\} \).
Zha [32] proposes a joint bidiagonalization (JBD) method for computing extreme GSVD components of the large matrix pair \( \{A, B\} \). At each step of the underlying JBD process, one needs to solve an \((m+p) \times n\) least squares problem with the coefficient matrix \( \begin{bmatrix} A & I \\ B & I \end{bmatrix} \), which may be costly using an iterative solver [17, 19]. Jia and Li [17] consider the JBD method in finite precision arithmetic and propose an efficient JBD process with partial reorthogonalization, which maintains necessary numerical (semi-)orthogonality among the basis vectors generated by the process. Hochstenbach [10] presents a Jacobi–Davidson (JD) type GSVD (JDGSVD) method to compute \( \ell \) GSVD components of \( \{A, B\} \) with the full column rank \( B \), where, at each expansion step, a correction equation of dimension \( m + n \) is solved iteratively. The upper \( m \)-dimensional part and the lower \( n \)-dimensional part of the approximate solution are used to expand one of the left searching subspaces and the right searching subspace. Refahi Sheikhani and Kordrostami [23] propose four variants of the JDGSVD method, where some slightly modified \((m + n)\)-by-(\(m + n\)) inner correction equations are solved iteratively using one of the four Krylov type methods: the full orthogonalization method (FOM), the generalized minimal residual method (GMRES), the weighted-FOM and weighted-GMRES methods. The JDGSVD method and these four variants formulate the GSVD of \( \{A, B\} \) as the mathematically equivalent generalized eigendecomposition of the augmented matrix pair \( \begin{bmatrix} A^T & I \\ B^T & B \end{bmatrix} \) for the full column rank \( B \) (resp. \( \begin{bmatrix} B^T & B \\ A^T & A \end{bmatrix} \) for the full column rank \( A \)).

Regarding JD type methods for large matrix eigenproblems and SVD and GSVD problems, solution accuracy of the correction equations involved critically affects their overall efficiency. How least accurately should the correction equations be solved has been a central problem that limits developments and practical applications of these methods. In recent years, it has been proven in [12, 15, 16] that it generally suffices to approximately solve all the correction equations with low or modest accuracy so that the outer iterations of the resulting inexact methods behave as if all the correction equations had been solved exactly. That is, an inexact method and its corresponding exact counterpart require very comparable outer iterations to achieve a prescribed stopping tolerance. Here the low or modest accuracy of inner iterations means that the relative residual norm of approximate solution of a correction equation is around \(10^{-3}\); see [12, 15, 16]. Such basic and important result can be easily adapted to any JD type GSVD method, e.g., JDGSVD and its variants, the method in [14] and the method to be proposed in this paper, so that their overall efficiency can be improved substantially.

Zwaan and Hochstenbach [34] present a generalized Davidson (GDGSVD) method and a multidirectional (MDGSVD) method, which are designed to compute an extreme partial GSVD of \( \{A, B\} \). At the extraction phase, the methods avoid forming \( A^T A \) and \( B^T B \), called the cross-product matrices of \( A \) and \( B \) [26], and apply the standard extraction approach to \( \{A, B\} \) directly for computing approximate GSVD components with respect to the given left and right searching subspaces, in which the two left subspaces are formed by premultiplying the right subspace with \( A \) and \( B \), respectively. At each iteration of the GDGSVD method, the right searching subspace is spanned by the residuals of the generalized Davidson method [1, Sec. 11.2.4 and Sec. 11.3.6] applied to the generalized eigenvalue problem of \( \{A^T A, B^T B\} \). In the MDGSVD method, a truncation technique is designed to discard an inferior search direction so as to improve the searching subspaces. Exploiting the Kronecker canonical form of a regular matrix pair [27], Zwaan [33] shows that the GSVD problem of \( \{A, B\} \) can be formulated as a \((2m + p + n) \times (2m + p + n)\) generalized eigenvalue problem without using any cross-product matrix or other matrix-matrix products. Currently, such formulation is of major theoretical value because the nontrivial eigenvalues and eigenvectors of the structured gen-
eralized eigenvalue problem are always complex with the generalized eigenvalues being the conjugate quaternions \((\sqrt{\sigma_j}, -\sqrt{\sigma_j}, i\sqrt{\sigma_j}, -i\sqrt{\sigma_j})\) with \(i\) the imaginary unit. Obviously, the size of the generalized eigenvalue problem is much bigger than that of the GSVD of \([A, B]\); the conditioning of eigenvalues and eigenvectors of this problem is unclear. Furthermore, there is no structure-preserving algorithm for the generalized eigenvalue problem in [33], and it is extremely difficult to seek for a numerically stable and efficient one.

In order to compute GSVD components accurately and avoid possible accuracy loss caused by forming \(A^TA\) and \(B^TB\) or one of them [13], it is preferable to work on \(A\) and \(B\) directly for given left and right searching subspaces. Another merit is that this way does not enlarge the problem size. In this paper, we first propose a basic cross-product free (CPF) JD type method for computing one, i.e., \(\ell = 1\), GSVD component of \([A, B]\), which is referred to as CPF-JDGSVD in the sequel. As done in the GDGSVD and MDGSVD methods [34], instead of constructing left and right searching subspaces independently, given a right searching subspace, CPF-JDGSVD generates the two left searching subspaces by acting \(A\) and \(B\) on the right subspace, respectively, and constructs their orthonormal bases by computing two thin QR factorizations of the matrices that are formed by premultiplying the matrix consisting of the orthonormal basis vectors of the right subspace with \(A\) and \(B\), respectively. But unlike [10] and [34], at the extraction stage, our method projects the GSVD of \([A, B]\) onto the left and right searching subspaces directly, and obtains the Ritz approximations, among which the approximation to the desired GSVD component is picked up. If the approximation does not yet converge, we expand the subspaces. To do so, we derive certain \(n \times n\) correction equations, whose solutions are used to expand the right searching subspace.

We shall, for the first time, give a theoretical justification that the distances between the desired left generalized singular vectors and the left subspaces are as small as that between the desired right generalized singular vector and the right subspace, as long as the desired generalized singular value \(\sigma\) is of modest size. We shall give a detailed derivation of the correction equations involved in CPF-JDGSVD. The correction equations are supposed to be approximately solved iteratively, called inner iterations. CPF-JDGSVD is an inner-outer iterative method with extraction steps of approximate GSVD components called outer iterations. We establish a convergence result on the approximate generalized singular values in terms of the residual norms. Meanwhile, adapting the results in [12, 15, 16] to CPF-JDGSVD, we will present some results on the least solution accuracy of the inner iterations in CPF-JDGSVD and compact estimates for the condition numbers of the correction equations. Based on them, we propose practical stopping criteria for the inner iterations, so that, for a given stopping tolerance, CPF-JDGSVD requires almost the same number of outer iterations as the exact CPF-JDGSVD where the correction equations are solved accurately. These will make the CPF-JDGSVD robust and maximize its overall efficiency as much as possible.

Restart, deflation and purgation are common but nontrivial issues in large matrix computations. Their effective and efficient implementations are strongly problem and method dependent, and can be very subtle and different in solutions of eigenvalue problems, and SVD and GSVD computations. As will be clear, they are particularly complicated for CPF-JDGSVD. We will propose novel effective and efficient thick-restart, deflation and purgation modules for CPF-JDGSVD that enable us to compute several GSVD components efficiently.

The rest of this paper is organized as follows. In Sect. 2, we propose the CPF-JDGSVD method, and present some theoretical results on its rationale and convergence. In Sect. 3, we derive correction equations involved in CPF-JDGSVD, and establish compact bounds for their asymptotic condition numbers and solution accuracy requirements on the inner iterations. Based on them, we design practical stopping criteria for the inner iterations. In Sect. 4, we propose a thick-restart CPF-JDGSVD algorithm with deflation and purgation for
computing more than one GSVD components. Numerical experiments are reported in Sect. 5 to demonstrate the performance of CPF-JDGSVD. It is necessary to point out that in this paper we focus on developing a general-purpose CPF-JDGSVD algorithm for computing extreme or interior GSVD components of \{A, B\}, and do not aim at specific applications of GSVDs. Finally, we conclude the paper in Sect. 6.

Throughout the paper, denote by \( \| \cdot \| \) the 2-norm of a vector or matrix, and assume that \( \| A \| \) and \( \| B \| \) themselves are modest, which can be achieved by suitable scaling. Keep in mind our assumption that \( \begin{bmatrix} A \\ B \end{bmatrix} \) has full column rank. Since the stacked matrix \( \begin{bmatrix} A \\ B \end{bmatrix} \) is better and can be much better conditioned than both A and B, we will assume that \( \begin{bmatrix} A \\ B \end{bmatrix} \) is well conditioned, which is definitely true, provided that one of A and B is well conditioned.

2 The Basic CPF-JDGSVD Algorithm

We propose a basic CPF-JDGSVD method for computing \((\alpha, \beta, u, v, x) = (\alpha_1, \beta_1, u_1, v_1, x_1)\) of \{A, B\} corresponding to the generalized singular value \( \sigma = \sigma_1 \) closest to the target \( \tau \). The method includes three major ingredients: (i) the construction of left searching subspaces for a given right searching subspace, (ii) an extraction approach of approximate GSVD components, and (iii) an expansion approach of the right subspace. We will prove that the accuracy of the two left searching subspaces constructed is similar to that of the right subspace, and establish an important convergence result on the approximate generalized singular values.

2.1 The Construction of Left Searching Subspaces and Extraction Approach

At iteration \( k \), assume that a \( k \)-dimensional right searching subspace \( \mathcal{X} \) is available, from which we seek an approximation to the desired right generalized singular vector \( x \). For the left generalized singular vectors \( u \) and \( v \), since \( Ax = \alpha u \) and \( Bx = \beta v \), it is natural to construct \( \mathcal{U} := A \mathcal{X} \) and \( \mathcal{V} := B \mathcal{X} \) as left searching subspaces and seek approximations to \( u \) and \( v \) from them, respectively.

We shall present a basic result, which shows that such \( \mathcal{U} \) and \( \mathcal{V} \) contain the same accurate information on the desired \( u \) and \( v \) as \( \mathcal{X} \) does on \( x \). Consequently, it is expected that we can compute approximate left and right generalized singular vectors with similar accuracy.

**Theorem 2.1** Let \( \mathcal{X} \) be a given right searching subspace and \( \mathcal{U} = A \mathcal{X} \) and \( \mathcal{V} = B \mathcal{X} \). Then for the right and left generalized singular vectors \( x \) and \( u, v \) of \{A, B\} associated with the generalized singular value \( \sigma = \frac{\alpha}{\beta} \), it holds that

\[
\sin \angle(\mathcal{U}, u) \leq \frac{\|A\| \|x\|}{\alpha} \sin \angle(\mathcal{X}, x),
\]

\[
\sin \angle(\mathcal{V}, v) \leq \frac{\|B\| \|x\|}{\beta} \sin \angle(\mathcal{X}, x).
\]

**Proof** For an arbitrary vector \( x' \in \mathcal{X} \), by the definition of sine of the angle between arbitrary two nonzero vectors, we have
\[
\sin \angle (Ax', Ax) = \min_\mu \frac{\|Ax - \mu Ax'\|}{\|Ax\|} = \min_\mu \frac{\|A(x - \mu x')\|}{\|Ax\|} \\
\leq \frac{\|A\\|}{\|Ax\|} \min_\mu \frac{\|x - \mu x'\|}{\|x\|} \\
= \frac{\|A\\|}{\|Ax\|} \sin \angle (x', x),
\]
where the last relation holds since \( Ax = \alpha u \) with \( \|u\| = 1 \). Therefore, we obtain
\[
\sin \angle (U, u) = \sin \angle (A, \mathcal{X}, Ax) = \min_{x' \in \mathcal{X}} \sin \angle (Ax', Ax) \\
\leq \frac{\|A\\|}{\|Ax\|} \min_{x' \in \mathcal{X}} \sin \angle (x', x) \\
= \frac{\|A\\|}{\|Ax\|} \sin \angle (\mathcal{X}, x),
\]
i.e., bound (2.1) holds. Bound (2.2) can be proved similarly. \( \square \)

Theorem 2.1 shows that when \( \mathcal{X} \) contains good information on the desired \( x \), the qualities of \( \mathcal{U} \) and \( \mathcal{V} \) are determined by \( \alpha, \|A\|, \|x\| \) and \( \beta, \|B\|, \|x\| \), respectively. Theorem 2.3 in [8] states that
\[
\|X\| = \left\| \begin{bmatrix} A^\dagger \\ B \end{bmatrix} \right\| \quad \text{and} \quad \|X^{-1}\| = \left\| \begin{bmatrix} A \\ B \end{bmatrix} \right\|,
\]
where \(^\dagger\) denotes the Moore–Penrose generalized inverse. Therefore, by \( \|X^{-1}\| \leq \|x\| \leq \|X\| \), we have
\[
\left\| \begin{bmatrix} A \\ B \end{bmatrix} \right\|^{-1} \leq \|x\| \leq \left\| \begin{bmatrix} A^\dagger \\ B \end{bmatrix} \right\|.
\]
Since \( \begin{bmatrix} A \\ B \end{bmatrix} \) is supposed to be well conditioned and \( \begin{bmatrix} A^\dagger \\ B \end{bmatrix} \) is scaled, the size of \( \|x\| \) is modest. Therefore, the qualities of \( \mathcal{U} \) and \( \mathcal{V} \) are similar to that of \( \mathcal{X} \) provided that \( \alpha \) and \( \beta \) are not very small. The theorem also indicates that, for any \( \sigma \), at least one of \( \mathcal{U} \) and \( \mathcal{V} \) is as good as \( \mathcal{X} \) as \( \alpha \) and \( \beta \) cannot be small simultaneously.

Given \( \mathcal{X} \), we now propose an extraction approach that seeks an approximate generalized singular value pair \( \tilde{\alpha}, \tilde{\beta} \) with \( \tilde{\alpha}^2 + \tilde{\beta}^2 = 1 \) and corresponding approximate generalized singular vectors \( \tilde{u} \in \mathcal{U}, \tilde{v} \in \mathcal{V} \) with \( \|\tilde{u}\| = \|\tilde{v}\| = 1 \) and \( \tilde{x} \in \mathcal{X} \) satisfying the projection:
\[
\begin{cases}
A\tilde{x} - \tilde{a}\tilde{u} = 0, \\
B\tilde{x} - \tilde{b}\tilde{v} = 0, \\
\tilde{b}A^T\tilde{u} - \tilde{a}B^T\tilde{v} \perp \mathcal{X}.
\end{cases}
\]

Let the columns of \( \tilde{X} \in \mathbb{R}^{n \times k} \) form an orthonormal basis of \( \mathcal{X} \) and
\[
A\tilde{X} = \tilde{U}G \quad \text{and} \quad B\tilde{X} = \tilde{V}H
\]
be thin QR factorizations of \( A\tilde{X} \) and \( B\tilde{X} \), respectively, where \( G \in \mathbb{R}^{k \times k} \) and \( H \in \mathbb{R}^{k \times k} \) are upper triangular. Suppose that \( G \) and \( H \) are nonsingular. Then the columns of \( \tilde{U} \in \mathbb{R}^{m \times k} \) and \( \tilde{V} \in \mathbb{R}^{p \times k} \) form orthonormal bases of \( \mathcal{U} \) and \( \mathcal{V} \), respectively. Write \( \tilde{u} = \tilde{U}e, \tilde{v} = \tilde{V}f, \) and \( \tilde{x} = \tilde{X}d \). Then (2.6) is equivalent to
\[
\begin{cases}
Gd = \tilde{a}e, \\
Hd = \tilde{b}f, \\
\tilde{b}G^T e = \tilde{a}H^T f,
\end{cases}
\]
which show that the \( \{\tilde{\alpha}, \tilde{\beta}\} \) are the generalized singular value of the \( k \times k \) matrix pair \( \{G, H\} \), and \( e, f \) and \( d \) are the corresponding left and right generalized singular vectors. We pick up \( \theta = \frac{\tilde{\alpha}}{\tilde{\beta}} \) closest to the target \( \tau \), and take

\[
(\tilde{\alpha}, \tilde{\beta}, \tilde{u} = \tilde{U}e, \tilde{v} = \tilde{V}f, \tilde{x} = \tilde{X}d)
\]

as an approximation to the desired GSVD component \( (\alpha, \beta, u, v) \) of \( \{A, B\} \).

It is straightforward to justify that the extraction approach (2.6) mathematically amounts to realizing the standard Rayleigh–Ritz projection of the regular matrix pair \( \{A^T A, B^T B\} \) onto \( \mathcal{X} \). However, we do not form \( A^T A \) and \( B^T B \) explicitly, and thus avoid the potential accuracy loss of the computed GSVD components of \( \{G, H\} \).

For the accuracy of the approximate left generalized singular vectors \( \tilde{u} \) and \( \tilde{v} \), notice from (2.6) that \( \tilde{u}, u \) and \( \tilde{v}, v \) are collinear with \( A\tilde{x}, Ax \) and \( B\tilde{x}, Bx \) correspondingly. Applying (2.3) to the angles \( \angle(A\tilde{x}, Ax) \) and \( \angle(B\tilde{x}, Bx) \), we obtain the following result.

**Theorem 2.2** Let \( \tilde{u} \in \mathcal{U}, \tilde{v} \in \mathcal{V} \) and \( \tilde{x} \in \mathcal{X} \) satisfy (2.6) and be the approximations to the generalized singular vectors \( u, v \) and \( x \) of \( \{A, B\} \) corresponding to the generalized singular value \( \{\alpha, \beta\} \). Then

\[
\sin \angle(\tilde{u}, u) \leq \frac{||A||_1 ||x||_{\alpha}}{\alpha} \sin \angle(\tilde{x}, x),
\]

\[
\sin \angle(\tilde{v}, v) \leq \frac{||B||_1 ||x||_{\beta}}{\beta} \sin \angle(\tilde{x}, x).
\]

Theorem 2.2 indicates that, with the left researching subspaces \( \mathcal{U} = A\mathcal{X} \) and \( \mathcal{V} = B\mathcal{X} \), our extraction approach (2.6) can indeed obtain the approximate left and right generalized singular vectors \( \tilde{u}, \tilde{v} \) and \( \tilde{x} \) with similar accuracy if \( \alpha \) and \( \beta \) are not very small.

By (2.6), since the approximate GSVD component \( (\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x}) \) satisfies \( A\tilde{x} = \tilde{\alpha}\tilde{u} \) and \( B\tilde{x} = \tilde{\beta}\tilde{v} \), the (absolute) residual of \( (\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x}) \) is

\[
r = r(\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x}) := \tilde{\beta}A^T \tilde{u} - \tilde{\alpha}B^T \tilde{v},
\]

and \( r = 0 \) if and only if \( (\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x}) \) is an exact GSVD component of \( \{A, B\} \). In computations, for a prescribed tolerance \( tol > 0 \), \( (\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x}) \) is claimed to have converged if

\[
||r|| \leq \left( \tilde{\beta} ||A||_1 + \tilde{\alpha} ||B||_1 \right) \cdot tol,
\]

where \( || \cdot ||_1 \) denotes the 1-norm of a matrix.

In the following, we present one of our main results, which, in terms of \( ||r|| \), gives the accuracy estimate of the approximate generalized singular value \( \theta = \tilde{\alpha}/\tilde{\beta} \). To this end and also for our later use, we introduce the distance function of two scalars \( \theta \) and \( \varsigma \):

\[
h(\theta, \varsigma) = \frac{\varsigma^2 - \theta^2}{1 + \varsigma^2} \quad \text{for} \quad \theta \geq 0 \quad \text{and} \quad \varsigma \geq 0.
\]

By \( \alpha_i^2 + \beta_i^2 = 1 \) and \( 1 + \sigma_i^2 = (\beta_i^2 + \alpha_i^2)/\beta_i^2 = 1/\beta_i^2 \), we have

\[
\alpha_i^2 - \beta_i^2 \theta^2 = (\sigma_i^2 - \theta^2)\beta_i^2 = h(\theta, \sigma_i), \quad i = 1, \ldots, q.
\]

**Theorem 2.3** Let \( (\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x}) \) be an approximate GSVD component of \( \{A, B\} \) satisfying (2.6) with \( \theta = \tilde{\alpha}/\tilde{\beta} \) and \( r \) be the corresponding residual defined by (2.12). Then (i) if

\[
\frac{\sigma_{\min}}{\sqrt{2} + \sigma_{\min}^2} < \theta < \sqrt{1 + 2\sigma_{\max}^2}
\]
with $\sigma_{\text{max}}$ and $\sigma_{\text{min}}$ the largest and smallest nontrivial generalized singular values of $\{A, B\}$, respectively, then there exists a nontrivial generalized singular value $\sigma$ of $\{A, B\}$ such that

$$\frac{|\sigma^2 - \theta^2|}{(1 + \sigma^2)\theta} \leq \frac{\|X\|^2 \|r\|}{\|\tilde{x}\|};$$  \hspace{1cm} (2.17)

(ii) if $\theta \geq \sqrt{1 + 2\sigma_{\text{max}}^2}$, then

$$\frac{1}{\theta} \leq \frac{\|X\|^2 \|r\|}{\|\tilde{x}\|};$$  \hspace{1cm} (2.18)

(iii) if $\theta \leq \frac{\sigma_{\text{min}}}{\sqrt{2 + \sigma_{\text{min}}^2}}$, then

$$\theta \leq \frac{\|X\|^2 \|r\|}{\|\tilde{x}\|}.$$  \hspace{1cm} (2.19)

**Proof** By definition (2.12), $A\tilde{x} = \tilde{a}\tilde{u}$ and $B\tilde{x} = \tilde{b}\tilde{v}$, we have

$$\theta r = \tilde{a}A^T\tilde{u} - \frac{\tilde{a}^2}{\tilde{b}}B^T\tilde{v} = (A^T A - \theta^2 B^T B)\tilde{x}.$$  \hspace{1cm} (2.20)

Premultiplying the two hand sides of the above by $X^T$ and exploiting (1.1), we obtain

$$\theta X^T r = X^T (A^T A - \theta^2 B^T B)XX^{-1}\tilde{x} = \text{diag}(C^2 - \theta^2 S^2, -\theta^2 I_{q_1}, I_{q_2})X^{-1}\tilde{x}.$$  \hspace{1cm} (2.21)

Taking norms on the above two hand sides and exploiting (2.15) give

$$\theta \|X^T r\| \geq \min\{1, \theta^2, \min_{i=1,\ldots,q} |\alpha_i^2 - \beta_i^2 \theta^2|\} \|X^{-1}\tilde{x}\|$$

$$\geq \frac{\|\tilde{x}\|}{\|X\|} \min\{1, \theta^2, \min_{i=1,\ldots,q} |h(\theta, \sigma_i)|\}.$$  \hspace{1cm} (2.21)

By (2.14), for $\theta < \sigma_{\text{max}}$ and $\sigma_{\text{max}} \leq \theta < \sqrt{1 + 2\sigma_{\text{max}}^2}$, we have

$$|h(\theta, \sigma_{\text{max}})| = h(\theta, \sigma_{\text{max}}) = \frac{\sigma_{\text{max}}^2 - \theta^2}{1 + \sigma_{\text{max}}^2} < 1,$$

$$|h(\theta, \sigma_{\text{max}})| = -h(\theta, \sigma_{\text{max}}) = \frac{\theta^2 - \sigma_{\text{max}}^2}{1 + \sigma_{\text{max}}^2} < 1,$$

respectively, proving that $\min_{i=1,2,\ldots,q} |h(\theta, \sigma_i)| < 1$. For $\theta > \sigma_{\text{min}}$ and $\frac{\sigma_{\text{min}}}{\sqrt{2 + \sigma_{\text{min}}^2}} < \theta \leq \sigma_{\text{min}}$, we obtain

$$|h(\theta, \sigma_{\text{min}})| = -h(\theta, \sigma_{\text{min}}) = \frac{\theta^2 - \sigma_{\text{min}}^2}{1 + \sigma_{\text{min}}^2} < \theta^2,$$

$$|h(\theta, \sigma_{\text{min}})| = h(\theta, \sigma_{\text{min}}) = \frac{\sigma_{\text{min}}^2 - \theta^2}{1 + \sigma_{\text{min}}^2} < \theta^2,$$

respectively, proving that $\min_{i=1,2,\ldots,q} |h(\theta, \sigma_i)| < \theta^2$. Therefore, under condition (2.16), we have

$$\min\{1, \theta^2, \min_{i=1,\ldots,q} |h(\theta, \sigma_i)|\} = \min_{i=1,\ldots,q} |h(\theta, \sigma_i)| = |h(\theta, \sigma)|,$$

which, together with (2.21), proves (2.17).
Under the conditions in (ii) and (iii), it is straightforward to justify that
\[
\min\{1, \theta^2, \min_{i=1, \ldots, q} |h(\theta, \sigma_i)|\} = 1,
\]
\[
\min\{1, \theta^2, \min_{i=1, \ldots, q} |h(\theta, \sigma_i)|\} = \theta^2,
\]
respectively. Therefore, from (2.21) we obtain (2.18) and (2.19).

By assumption and (2.4), it is known that \( \|X\| \) is modest. From (2.7) and the orthonormality of \( \tilde{X} \), it is easily justified that
\[
\|G\| \leq \|A\|, \quad \|G^\top\| \leq \|B\|.
\] (2.22)

Exploiting Lemma 2.4 of [13] and adapting (2.4) and (2.5) to \( \{G, H\} \), for \( d \) defined by (2.8) we obtain
\[
\|G\|^{-1} \leq \|d\| \leq \|G^\top\|.
\] (2.23)

Therefore, from (2.4), (2.22), (2.23), and \( \|\tilde{x}\| = \|\tilde{X}d\| = \|d\| \), we have
\[
\frac{\|X\|^2}{\|\tilde{x}\|} \leq \|A\| \kappa\left(\frac{A}{B}\right).
\]

We should comment that assumption (2.16) holds generally, which roughly states that \( \theta \) lies between \( \sigma_{\text{min}} \) and \( \sigma_{\text{max}} \). Under the assumption, bound (2.17) shows that \( \theta \) must converge to a nontrivial \( \sigma \) as \( \|r\| \) tends to zero.

For the scaled matrix pair \( \{\gamma A, \gamma B\} \) with any given nonzero constant \( \gamma \), the approximate GSVD components becomes \( (\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \frac{1}{\gamma}\tilde{x}) \), the residual is \( \gamma r \), and the right generalized singular vector matrix of \( \{\gamma A, \gamma B\} \) is \( \frac{1}{\gamma}X \). Inserting them into (2.17)–(2.19) yields the same results. These indicate that \( \frac{\|X\|^2}{\|\tilde{x}\|} \) in the right-hand sides of (2.17)–(2.19) is invariant under the scaling of \( \{A, B\} \). Notice that \( \|r\| \) is a backward error and the left-hand side of (2.17) is a forward error of \( \sigma \). It is instructive to regard the factor \( \frac{\|X\|^2}{\|\tilde{x}\|} \) as a condition number of \( \sigma \) when bounding the error of \( \sigma \) in terms of the residual norm \( \|r\| \).

### 2.2 Subspace Expansion

If the current GSVD approximation \( (\alpha, \beta, u, v, x) \) does not yet converge, one needs to expand the searching subspaces \( \mathcal{X} \) and \( \mathcal{V} \) in order to obtain a more accurate approximate GSVD component with respect to them. Since we construct the left searching subspaces by \( \mathcal{V} = A\mathcal{X} \) and \( \mathcal{V}' = B\mathcal{X}' \), the key is to expand \( \mathcal{X} \) effectively.

Keep in mind that \( (\sigma^2, x) \) is an eigenpair of \( \{A^T A, B^T B\} \) with \( \sigma = \alpha/\beta \), Suppose that an approximate right generalized singular vector \( \tilde{x} \in \mathcal{X} \) is available. We aim to seek for a correction vector \( t \) satisfying
\[
t \bot \tilde{y} := (A^T A + B^T B)\tilde{x} = \tilde{\alpha}A^T \tilde{u} + \tilde{\beta}B^T \tilde{v}
\] (2.24)
such that \( \tilde{x} + t \) is an unnormalized right generalized singular vector of \( \{A, B\} \); that is, \( (\sigma^2, \tilde{x} + t) \) is an exact eigenpair of \( \{A^T A, B^T B\} \):
\[
A^T A(\tilde{x} + t) = \sigma^2 B^T B (\tilde{x} + t).
\] (2.25)

Rearranging this equation, we obtain
\[
(A^T A - \theta^2 B^T B)t = -(A^T A - \theta^2 B^T B)\tilde{x} + (\sigma^2 - \theta^2)B^T Bx + (\sigma^2 - \theta^2)B^T Bt.
\] (2.26)
where $\theta = \tilde{\alpha}/\tilde{\beta}$ is the current approximation to $\sigma$.

Assume that $\tilde{x}$ is already reasonably accurate with the normalization $\tilde{x}^T (A^T A + B^T B) \tilde{x} = 1$, which means that $\|t\|$ is small relative to $\|\tilde{x}\|$. Premultiplying $\tilde{x}^T$ both hand sides of (2.25), we have

$$
\sigma^2 = \frac{\tilde{x}^T A^T A(x + t)}{\tilde{x}^T B^T B(x + t)} = \frac{\tilde{\alpha}^2 + \tilde{x}^T A^T At}{\tilde{\beta}^2 + \tilde{x}^T B^T Bt} = \frac{\tilde{\alpha}^2 + \tilde{\beta}^2 \tilde{x}^T A^T At - \tilde{\alpha}^2 \tilde{x}^T B^T Bt}{\tilde{\beta}^2 (\tilde{\beta}^2 + \tilde{x}^T B^T Bt)}
$$

$$
\theta^2 + \frac{\tilde{x}^T A^T At - \theta^2 \tilde{x}^T B^T Bt}{\tilde{\beta}^2 + \tilde{x}^T B^T Bt} = \theta^2 + \frac{(A^T \tilde{x} - \theta^2 B^T B \tilde{x})^T t}{\tilde{\beta}^2 + \tilde{x}^T B^T Bt}.
$$

(2.27)

It is immediate from (2.25) that $\|(A^T A - \theta^2 B^T B) \tilde{x}\| = \mathcal{O}(\|t\|)$. Inserting this into (2.27), we obtain

$$
\sigma^2 = \theta^2 + \mathcal{O}(\|t\|^2) = \theta^2 (1 + \mathcal{O}(\|t\|^2))
$$

for $\|t\|$ sufficiently small. Therefore, the sizes of the second and third terms in the right-hand side of (2.26) are higher order small $\mathcal{O}(\|t\|^2)$ and $\mathcal{O}(\|t\|^3)$, respectively.

Note from (2.20) that the first term in the right-hand side of (2.26) is collinear with the residual $r$ of $(\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x})$, which is orthogonal to $\mathcal{X}$ by the third condition in (2.6). Therefore, the first term in the right-hand side of (2.26) is orthogonal to $\tilde{x} \in \mathcal{X}$. Moreover, we know from (2.6) and (2.24) that $\tilde{y}^T \tilde{x} = 1$ and $(I - \tilde{y}\tilde{y}^T)$ is an oblique projector onto the orthogonal complement $\tilde{x}^\perp$ of $\text{span}([\tilde{x}])$. Neglecting the last two $\mathcal{O}(\|t\|^2)$ and $\mathcal{O}(\|t\|^3)$ terms in the right-hand side of (2.26), we obtain

$$
(I - \tilde{y}\tilde{x}^T) (A^T A - \theta^2 B^T B) t = -\theta r \quad \text{with} \quad t \perp \tilde{y}.
$$

(2.28)

Since requirement $t \perp \tilde{y}$ means $t = (I - \tilde{y}\tilde{x}^T)t$, we can replace $t$ with $(I - \tilde{y}\tilde{x}^T)t$ in (2.28). Notice that it is the direction other than the size of $t$ that matters when expanding $\mathcal{X}$ by adding $t$ to it. Therefore, it makes no difference when solving (2.28) with the right-hand side $-\theta r$ or $-r$. As a consequence, we have ultimately derived the correction equation

$$
(I - \tilde{y}\tilde{x}^T)(A^T A - \theta^2 B^T B)(I - \tilde{y}\tilde{x}^T)t = -r \quad \text{with} \quad t \perp \tilde{y}.
$$

(2.29)

Solving it for $t$ and orthonormalizing $t$ against $\tilde{X}$ yield the subspace expansion vector $x_+ = \frac{(I - \tilde{y}\tilde{x}^T)t}{\|(I - \tilde{y}\tilde{x}^T)t\|}$. The columns of the updated $\tilde{X} := [\tilde{X}, x_+]$ form an orthonormal basis of the expanded $(k + 1)$-dimensional right searching subspace $\mathcal{X} := \mathcal{X} + \text{span}(x_+)$. The coefficient matrix in (2.29) dynamically depends on $\theta$ as the outer iterations proceed. In practice, $\theta$ may have very limited or little accuracy as approximations to $\sigma$ in an initial stage, so that solving (2.29) with varying $\theta$ may not gain and even may misconverge. To this end, a better way is to solve (2.29) with $\theta$ replaced by the target $\tau$ (cf. [26, Sect. 6.2] and [28, Chapter IX, Sect. 45]):

$$
(I - \tilde{y}\tilde{x}^T)(A^T A - \tau^2 B^T B)(I - \tilde{y}\tilde{x}^T)t = -r \quad \text{with} \quad t \perp \tilde{y}
$$

(2.30)

in the initial stage and then switch to solving (2.29) when $\|r\|$ becomes fairly small; that is, $\theta$ has had some accuracy. Approximately solving (2.30) or (2.29) iteratively is called the inner iterations in CPF-JDGSVD. In computations, if

$$
\|r\| \leq (\tilde{\beta}\|A\|_1 + \tilde{\alpha}\|B\|_1) \cdot \text{fixtol}
$$

(2.31)

with fixtol fairly small but bigger than the stopping tolerance tol of outer iterations, then we switch to solving (2.29).
3 Properties of the Correction Equations and Stopping Criteria for the Inner Iterations

For the large $A$ and $B$, suppose that only iterative solvers are computationally viable to solve the correction equations (2.29) and (2.30). Since the coefficient matrices in them are symmetric and typically indefinite, the minimal residual method (MINRES) is a most commonly used choice [7, 24]. We derive some compact estimates for the condition numbers of the correction equations (2.29) and (2.30) when $\theta = \sigma$ and $\bar{x} = x$ ideally. Meanwhile, we make an analysis on the least solution accuracy requirement on the correction equations in order to maximize the overall efficiency of CPF-JDGSVD as much as possible. Based on these results, we are able to propose practical stopping criteria for the inner iterations. We focus on (2.29), and, as it will turn out, the results are directly applicable to (2.30).

3.1 Condition Numbers of the Correction Equations

The coefficient matrix in the correction equation (2.29) is restricted to the orthogonal complement $\tilde{y}^\perp$ of span{$\tilde{y}$}, and maps $\tilde{y}^\perp$ to $\tilde{x}^\perp$. We denote this restricted linear operator by

$$ M = (A^T A - \theta^2 B^T B)|_{\tilde{y}^\perp \rightarrow \tilde{x}^\perp}. \quad (3.1) $$

As will be clear, the condition number $\kappa(M) = \|M\|\|M^T\|$ determines the reliability of adopting the relative residual norm of the correction equation as the measurement of inner iteration accuracy. Therefore, it is important to derive sharp estimates for $\kappa(M)$. However, it is generally not possible to do so for a general approximation $\tilde{x}$. Fortunately, sharp estimates for the ideal case that $\theta = \sigma$ and $\bar{x} = x$ suffice and are insightful since, by a continuity argument, they will exhibit the asymptotic behavior of $\kappa(M)$ when $\theta \to \sigma$ and $\bar{x} \to x$.

Based on the GSVD (1.1) of $\{A, B\}$ and (1.2), we partition

$$ U = [u, U_2], \quad V = [v, V_2], \quad X = [x, X_2], \quad \Sigma_A = \begin{bmatrix} \alpha & \Sigma_{A, 2} \end{bmatrix}, \quad \Sigma_B = \begin{bmatrix} \beta & \Sigma_{B, 2} \end{bmatrix}, \quad (3.2) $$

where the matrices

$$ \Sigma_{A, 2} = \text{diag}\{C_2, 0_{l_1 q_1}, I_{q_2}\} \quad \text{and} \quad \Sigma_{B, 2} = \text{diag}\{S_2, I_{q_1}, 0_{l_2 q_2}\} \quad (3.3) $$

with $C_2 = \text{diag}\{\alpha_2, \ldots, \alpha_q\}$ and $S_2 = \text{diag}\{\beta_2, \ldots, \beta_q\}$. By $X^T (A^T A + B^T B) X = I$, we obtain

$$ Y = X^{-T} = (A^T A + B^T B) X = [y, Y_2] \quad (3.4) $$

with $y = (A^T A + B^T B) x$ and $Y_2 = (A^T A + B^T B) X_2$. Then $X_2^T y = 0$ and $Y_2^T x = 0$, and the columns of $Y_2$ form a basis of $x^\perp$. Let

$$ Y_2 = Q_y R_y \quad (3.5) $$

be the thin QR factorization of $Y_2$. Then the columns of $Q_y$ form an orthonormal basis of $x^\perp$. It is obvious from (2.24) and (3.4) that $y = y$ when $\tilde{x} = x$.

**Theorem 3.1** Set $\tilde{x} = x$ and $\tilde{y} = y$ in (3.1), and assume that $\sigma$ is a simple nontrivial generalized singular value of $\{A, B\}$. Then

$$ M' = X_2^T (A^T A - \sigma^2 B^T B) X_2 = \Sigma_{A, 2}^T \Sigma_{A, 2} - \sigma^2 \Sigma_{B, 2} \Sigma_{B, 2}^T \quad (3.6) $$


is nonsingular with $X_2$ and $\Sigma_{A,2}, \Sigma_{B,2}$ defined by (3.2) and (3.3), respectively. Furthermore,

$$
\kappa(M) = \kappa(R_y M' R_y^T) \leq \kappa^2 \left( \left[ \begin{array}{c} A \\ B \end{array} \right] \right) \max \left\{ \frac{\max[1, \sigma_i^2]}{1 + \sigma_i^2}, \frac{\max[1, \sigma_i^2]}{1 + \sigma_i^2} \right\}, \tag{3.7}
$$

where $R_y$ is defined in (3.5) and $\sigma_i$ is the minimizer of $\min_{i=2,3,\ldots,q} |\alpha_i^2 - \beta_i^2 \sigma_i^2|$. 

**Proof** For $\bar{x} = x$ and $\bar{y} = y$, we have $\theta = \sigma$ in (3.1). For $Y = X^{-T}$ in (3.4), we have $YX^T = XY^T = I$. Therefore, from (3.5) and (3.6), the coefficient matrix in (2.29) is

$$(I - xy^T)(A^T - \sigma^2 B^T B)(I - xy^T) = (YX^T - yx^T)(A^T - \sigma^2 B^T B)(XY^T - xy^T)$$

$$= Y_2X_2^T(A^T - \sigma^2 B^T B)X_2Y_2^T$$

$$= Q_y R_y M' R_y^T Q_y^T. \tag{3.8}$$

By the GSVD (1.1) of $[A, B]$ and (3.3), we obtain

$$M' = X_2^T A^T AX_2 - \sigma^2 X_2^T B^T BX_2$$

$$= \Sigma_{A,2}^T \Sigma_{A,2} - \sigma^2 \Sigma_{B,2} \Sigma_{B,2}$$

$$= \text{diag}(C_2^2 - \sigma^2 S_2^2, -\sigma^2 I_{q_1}, I_{q_2}). \tag{3.9}$$

which, by the assumption, proves that $M'$ is nonsingular. Since $Q_y$ is column orthonormal, it follows from (3.1) and (3.8) that $M$ is nonsingular and

$$\kappa(M) = \kappa(R_y M' R_y^T) \leq \kappa^2(R_y)\kappa(M'). \tag{3.10}$$

Since $Y_2$ consists of the second to the last columns of $Y$, from (3.5) and $Y = X^{-T}$ we obtain

$$\|R_y\| = \|Y_2\| \leq \|Y\| = \|X^{-1}\|, \quad \|R_y^{-1}\| = \|Y_2^+\| \leq \|Y^{-1}\| = \|X\|,$$

showing that $\kappa(R_y) \leq \kappa(X)$. Therefore, it follows from (2.4) that

$$\kappa(R_y) \leq \kappa(\left[ \begin{array}{c} A \\ B \end{array} \right]). \tag{3.11}$$

From (2.15), the diagonal elements of $M'$ are $\alpha_i^2 - \beta_i^2 \sigma_i^2 = h(\sigma, \sigma_i), i = 2, \ldots, q$ with $h(\sigma, \sigma_i)$ defined by (2.14). By definition, it is straightforward that

$$|h(\sigma, \sigma_i)| \leq \max \left\{ \frac{\sigma_i^2}{1 + \sigma_i^2}, \frac{\sigma_i^2}{1 + \sigma_i^2} \right\} \leq \max[1, \sigma_i^2]. \tag{3.12}$$

Applying it and (2.15) to (3.9) yields

$$\sigma_{\max}(M') = \max_{i=2,\ldots,q} \{1, \sigma_i^2, |\alpha_i^2 - \beta_i^2 \sigma_i^2|\} \leq \max[1, \sigma_i^2]. \tag{3.13}$$

Note that

$$\sigma_{\min}(M') = \min_{i=2,\ldots,q} \min \{1, \sigma_i^2, |\alpha_i^2 - \beta_i^2 \sigma_i^2|\}. \tag{3.14}$$

We next consider the following two cases.

Case (i): If

$$\sigma_{\min}(M') = \min_{i=2,\ldots,q} |\alpha_i^2 - \beta_i^2 \sigma_i^2| = |\alpha_x^2 - \beta_x^2 \sigma_x^2| = \frac{\sigma_x^2 - \sigma^2}{1 + \sigma_x^2}. \tag{3.15}$$
that is, $\sigma$ is comparatively clustered with $\sigma^\ast$, then by (3.13) we obtain

$$
\kappa(M') = \frac{\max\{1, \sigma^2\}(1 + \sigma^2)}{|\sigma^2 - \sigma^2|}.
$$

(3.14)

Case (ii): If $\sigma_{\min}(M') = \min\{1, \sigma^2\}$, that is, $\sigma$ is comparatively well separated from $\sigma^\ast$, then it follows from (3.13) that

$$
\kappa(M') = \frac{\max\{1, \sigma^2\}}{\min\{1, \sigma^2\}}.
$$

Bound (3.7) follows from the application of this relation, (3.14) and (3.11) to (3.10).

In an analogous manner, for the correction equation (2.30), define $M^\tau$ by replacing $\theta$ with $\tau$ in (3.1). Then for $\tilde{x} = x$ and $\tilde{y} = y$, we have

$$
\kappa(M^\tau) \leq \kappa^2 \left[\begin{array}{c} A \\ B \end{array}\right] \max \left\{ \frac{\max\{1, \tau^2\}(1 + \sigma^2_{\ast, \tau})}{|\sigma^2_{\ast, \tau} - \tau^2|}, \frac{\max\{1, \tau^2\}}{\min\{1, \tau^2\}} \right\},
$$

(3.15)

where $\sigma^\ast_{\ast, \tau}$ is the minimizer of $\min_{i=2,3,\ldots,q} |\sigma^2_{\ast, \tau} - \tau^2|$.

We remark that, by a continuity argument, bounds (3.7) and (3.15) asymptotically hold as $\tilde{x} \to x$ and $\theta \to \sigma$. Therefore, (3.7) and (3.15) give good estimates for $\kappa(M)$ and $\kappa(M^\tau)$, respectively, once $\tilde{x}$ becomes a reasonably good approximation to $x$.

### 3.2 Least Accuracy Requirements on the Inner Iterations

We make an analysis on the inner iterations and seek for least accuracy requirements on them in order to make the outer iterations of the resulting inexact CPF-JDGSDV method mimic those of the exact CPF-JDGSDV where the correction equations are solved accurately. This will maximize the overall efficiency of the inexact method as much as possible.

Assume that $\theta \neq \sigma$. Then $A^T A - \theta^2 B^T B$ is nonsingular. Denote the matrices

$$
L = (A^T A - \theta^2 B^T B)^{-1} \quad \text{and} \quad K = L(A^T A + B^T B).
$$

(3.16)

The eigenpairs $(\sigma^2, x)$, $(0, x)$ and $(+\infty, x)$ of $\{A^T A, B^T B\}$ are transformed into the eigenpairs $(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2}, x)$, $(-\frac{1}{\theta^2}, x)$ and $(1, x)$ of $K$, respectively.

By (2.20) and $(I - \tilde{x}\tilde{y}^T)t = t$, equation (2.29) can be rearranged as

$$
(I - \tilde{y}\tilde{x}^T)L^{-1}(\theta t) = -L^{-1}\tilde{x},
$$

whose solution is

$$
\theta t = -\tilde{x} + \nu L\tilde{y}
$$

(3.17)

with $\nu = \tilde{x}^T L^{-1}(\theta t)$. Premultiplying both hand sides of (3.17) by $\tilde{y}^T$ and making use of the orthogonality $t \perp \tilde{y}$ and the normalization $\tilde{y}^T\tilde{x} = 1$, we obtain

$$
\nu = \frac{1}{\tilde{y}^T L\tilde{y}}.
$$

(3.18)

Let $\tilde{t}$ be an approximate solution of (2.29) with the relative error $\varepsilon = \frac{||\tilde{t} - t||}{||t||}$. Then

$$
\tilde{t} = t + \varepsilon ||t|| s,
$$

(3.19)
where $s$ is the error direction vector with $\|s\| = 1$, and the exact and inexact expansion vectors are $x_+ = \frac{(I-\tilde{X}\tilde{X}^T)y}{\|I-\tilde{X}\tilde{X}^T\|y}$ and $\tilde{x}_+ = \frac{(I-\tilde{X}\tilde{X}^T)y}{\|I-\tilde{X}\tilde{X}^T\|y}$, respectively. The relative error of $x_+$ and $\tilde{x}_+$ can be defined as

$$
\bar{\varepsilon} = \frac{\|(I - \tilde{X}\tilde{X}^T)t - (I - \tilde{X}\tilde{X}^T)t\|}{\|\|(I - \tilde{X}\tilde{X}^T)t\|\}}. \tag{3.20}
$$

As has been shown in [12, 15, 16], in order to make the ratio of the distance between $x$ and $\mathcal{X} + \text{span}\{x_+\}$ and that between $x$ and $\mathcal{X} + \text{span}\{\tilde{x}_+\}$ lie in $[0.999, 1.001]$, which means that $\mathcal{X} + \text{span}\{x_+\}$ and $\mathcal{X} + \text{span}\{\tilde{x}_+\}$ contain almost the same information on $x$, it generally suffices to take a fairly small

$$\bar{\varepsilon} \in [10^{-4}, 10^{-3}], \tag{3.21}$$

which will be utilized when designing robust stopping criteria for the inner iterations.

We next relate $\varepsilon$ to $\bar{\varepsilon}$, which is the third main result of this paper.

**Theorem 3.2** Let $\tilde{t}$ be an approximation to the exact solution $t$ of (2.29) with the relative error $\varepsilon$ satisfying (3.19), and $\tilde{\varepsilon}$ be defined by (3.20). Let $L$ and $K$ be defined by (3.16) and $K' = X^T K X_\perp$ with $X_\perp$ such that $\{x, x_+\}$ is orthogonal, and assume that

$$\text{sep}(\rho, K') = \|(K' - \rho I)^{-1}\| > 0 \quad \text{with} \quad \rho = \tilde{y}^T L\tilde{y} = 1/\nu. \tag{3.22}$$

Then

$$\nu \leq \frac{2\|K\|}{\text{sep}(\rho, K')\|s_\perp\|} \bar{\varepsilon}, \tag{3.23}$$

where $s_\perp = (I - \tilde{X}\tilde{X}^T)s$ with the vector $s$ defined by (3.19).

**Proof** Premultiplying both hand sides of (3.19) by $(I - \tilde{X}\tilde{X}^T)$ and taking norms give

$$\varepsilon = \frac{\|(I - \tilde{X}\tilde{X}^T)\tilde{t} - (I - \tilde{X}\tilde{X}^T)t\|}{\|\|t\|\|(I - \tilde{X}\tilde{X}^T)s\|\}} = \frac{\|(I - \tilde{X}\tilde{X}^T)t\|}{\|\|t\|\|s_\perp\|\}}. \tag{3.24}$$

By (3.17), substituting $t = \frac{1}{\nu}(-\tilde{x} + vL\tilde{y})$ into the above relation and making use of $\tilde{x} \in \mathcal{X}$ and $\tilde{y} = (A^T A + B^T B)\tilde{x}$, we obtain

$$\varepsilon = \frac{\|(I - \tilde{X}\tilde{X}^T)(-\tilde{x} + vL\tilde{y})\|_{s_\perp}}{\|\|s_\perp\|\|\nu L\tilde{y} - \tilde{x}\|\|s_\perp\|\}} = \frac{\|(I - \tilde{X}\tilde{X}^T)(vL\tilde{y})\|_{s_\perp}}{\|\|s_\perp\|\|\nu L\tilde{y} - \tilde{x}\|\|s_\perp\|\}} = \frac{\|(I - \tilde{X}\tilde{X}^T)K\tilde{x}\|_{s_\perp}}{\|\|s_\perp\|\|\nu L\tilde{y} - \tilde{x}\|\|s_\perp\|\}}. \tag{3.25}$$

where $\nu = \frac{1}{\tilde{y}^T L\tilde{y}}$ by definition (3.18).

Since $\tilde{x} \rightarrow x$ and $\tilde{y} \rightarrow y = (A^T A + B^T B)x$, from (3.16) and $x^T y = 1$ we have

$$\rho \rightarrow y^T L(A^T A + B^T B)x = y^T Kx = \frac{\sigma^2 + 1}{\sigma^2 - \theta^2} x^T x = \frac{\sigma^2 + 1}{\sigma^2 - \theta^2}. \tag{3.26}$$

Therefore, the pair $(\rho, \tilde{x})$ is an approximation to the simple eigenpair $(\sigma^2 + 1, x)$ of $K$. Set $x_s = x/\|x\|$, and note that $[x_s, X_\perp]$ is orthogonal. Then we have a Schur like decomposition:

$$[x_s^T X_\perp^T] K [x_s X_\perp] = \begin{bmatrix} \sigma^2 + 1 & x_s^T K X_\perp \\ 0 & K' \end{bmatrix}. \tag{3.26}$$

Springer
with $K' = X_T^T K X_\perp$. By Theorem 6.1 of [18], we obtain
\[
\|K\tilde{x}_s - \rho\tilde{x}_s\| \geq \sin \psi \cdot \text{sep}(\rho, K'),
\] (3.27)
where \(\text{sep}(\rho, K') = \|(K' - \rho I)^{-1}\|^{-1}\) and \(\psi = \angle(\tilde{x}, x)\) is the acute angle of \(\tilde{x}\) and \(x\).

Let us decompose \(\tilde{x}_s\) and \(x_s\) into the orthogonal direct sums:
\[
\tilde{x}_s = x_s \cos \psi + w_1 \sin \psi \quad \text{and} \quad x_s = \tilde{x}_s \cos \psi + w_2 \sin \psi,
\]
where \(w_1 \perp x_s\) and \(w_2 \perp \tilde{x}_s\) with \(\|w_1\| = \|w_2\| = 1\). From \((I - \tilde{X}\tilde{X}^T)\tilde{x}_s = 0\), we obtain
\[
(I - \tilde{X}\tilde{X}^T)K\tilde{x}_s = (I - \tilde{X}\tilde{X}^T)K(x_s \cos \psi + w_1 \sin \psi)
= (I - \tilde{X}\tilde{X}^T)\left(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2} x_s \cos \psi + K w_1 \sin \psi\right)
= (I - \tilde{X}\tilde{X}^T)\left(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2} \tilde{x}_s \cos \psi + \omega \sin \psi + K w_1 \sin \psi\right)
= (I - \tilde{X}\tilde{X}^T)\left(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2} \omega \sin \psi + K w_1 \sin \psi\right) \sin \psi.
\]

Since \(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2}\) is an eigenvalue of \(K\), we have \(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2} \leq \|K\|\), and obtain
\[
\|(I - \tilde{X}\tilde{X}^T)K\tilde{x}_s\| \leq \|(I - \tilde{X}\tilde{X}^T)\| \|K\| (\|\omega \sin \psi + \|w_1\|) \sin \psi
\leq 2\|(I - \tilde{X}\tilde{X}^T)\| \|K\| \sin \psi
\leq 2\|K\| \sin \psi.
\] (3.28)
Relation (3.23) then follows by applying (3.27) and (3.28) to (3.24).

Theorem 3.2 reveals a key relationship between the solution accuracy \(\epsilon\) of (2.29) and the accuracy \(\tilde{\epsilon}\) of the expansion vector \(\tilde{x}_+\). For (2.30), we can analogously prove that
\[
\epsilon \leq \frac{2\|K_\tau\|}{\text{sep}(\rho_\tau, K'_\tau)} \|s_\perp\| \tilde{\epsilon} \quad \text{with} \quad \rho_\tau = \tilde{y}^T L_\tau \tilde{y},
\] (3.29)
where
\[
L_\tau = (A^T A - \tau^2 B^T B)^{-1}, \quad K_\tau = L_\tau (A^T A + B^T B), \quad K'_\tau = X^T_\perp K_{\tau} X_\perp.
\] (3.30)

Bound (3.23) (resp. (3.29)) indicates that once \(\tilde{\epsilon}\) is given, we are able to determine the least accuracy requirement \(\epsilon\) for (2.29) (resp. (2.30)) from (3.23) (resp. (3.29)).

### 3.3 Practical Criteria for the Inner Iterations

Our goal is to derive a practical least solution accuracy requirement for the relevant correction equation, so that the inexact CPF-JDGSVD method and the exact CPF-JDGSVD method use almost the same outer iterations to achieve a prescribed stopping tolerance.

From (3.23) and (3.29), since \(\|s_\perp\|\) is uncomputable in practice, we simply replace it by its upper bound one. This makes \(\epsilon\) as small as possible, so that the inexact CPF-JDGSVD method is more reliable to mimic its exact counterpart.

By (3.25), we can use \(\text{sep}(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2}, K')\) to estimate \(\text{sep}(\rho, K')\). From (3.16) and the GSVD (1.1) of \(\{A, B\}\), the other eigenvalues of \(K\) than \(\frac{\sigma^2 + 1}{\sigma^2 - \theta^2}\) are \(q_1\)-multiple \(-\frac{1}{\theta^2}\), \(q_2\)-multiple 1.
and \( \frac{\sigma_i^2 + 1}{\sigma_i^2 - \theta^2} \), \( i = 2, \ldots, q \). By (3.26), they are also the eigenvalues of \( K' \). Therefore,

\[
\text{sep}(\rho, K') \approx \min \left\{ \frac{\sigma^2 + 1}{\sigma^2 - \theta^2} + \frac{1}{\rho^2}, \frac{\sigma^2 + 1}{\sigma^2 - \theta^2} - 1 \right\}, \text{min}_{i=2,\ldots,q} \left\{ \frac{\sigma^2 + 1}{\sigma_i^2 - \theta^2} - 1 \right\}
\]

\[
\approx \frac{\sigma^2 + 1}{|\sigma^2 - \theta^2|},
\]

where we have used \( \theta \approx \sigma \). Since the eigenvalue \( \frac{\sigma^2 + 1}{\sigma^2 - \theta^2} \) is the largest one in magnitude of \( K \), it is reasonable to use \( \frac{\sigma^2 + 1}{|\sigma^2 - \theta^2|} \) to estimate \( \| K \| \). Applying these estimates for \( \| K \| \) and \( \text{sep}(\rho, K') \) to (3.23), we obtain

\[
\varepsilon \leq 2\tilde{\varepsilon}.
\]

(3.31)

Thus we terminate the inner iterations of solving the correction equation (2.29) once (3.31) is met for a given \( \tilde{\varepsilon} \in [10^{-4}, 10^{-3}] \); see (3.21).

If \( \theta \) is replaced by the fixed target \( \tau \), the other eigenvalues of \( K_\tau \) than \( \frac{\sigma^2 + 1}{\sigma^2 - \tau^2} \) are \( q_1 \)-multiple \( -\frac{1}{\tau^2}, q_2 \)-multiple \( 1 \) and \( \frac{\sigma_i^2 + 1}{\sigma_i^2 - \tau^2}, i = 2, \ldots, q \), which are also the eigenvalues of \( K'_\tau \).

For (3.29), since the parameter \( \rho_1 = y^T L_\tau y \approx \frac{\sigma^2 + 1}{\sigma^2 - \tau^2} \approx \frac{\sigma_i^2 + 1}{\sigma_i^2 - \tau^2} \), we use \( \theta^2 + 1 \)

\[
\text{se}
(\theta^2 - \tau^2, K'_\tau) \approx \min \left\{ \left\| \frac{\theta^2 + 1}{\theta^2 - \tau^2} + \frac{1}{\tau^2} \right\|, \left\| \frac{\theta^2 + 1}{\theta^2 - \tau^2} - 1 \right\|, \min_{i=2,\ldots,q} \left\{ \left\| \frac{\theta^2 + 1}{\theta_i^2 - \tau^2} - 1 \right\| \right\}
\]

Observe the absolute value of the largest eigenvalue in magnitude of \( K_\tau \) is

\[
\max_{i=1,\ldots,q} \left\{ \frac{1}{\tau^2}, 1, \frac{\sigma_i^2 + 1}{|\sigma_i^2 - \tau^2|} \right\}.
\]

We use it as an estimate for \( \| K_\tau \| \). Since the eigenvalues of \( K_\tau \) and \( K'_\tau \) are unknown, we can further replace the above two a-priori estimates by exploiting the information available in computations as follows: Let \( \theta_i, i = 1, 2, \ldots, k \) be the generalized singular values of \( \{ G, H \} \), and suppose that \( \theta = \theta_1 \) approximates the desired \( \sigma \). Then

\[
\| K_\tau \| \approx \max_{i=1,\ldots,k} \left\{ \frac{1}{\tau^2}, 1, \frac{\theta_i^2 + 1}{|\theta_i^2 - \tau^2|} \right\}
\]

(3.32)

and

\[
\text{sep} \left( \frac{\theta^2 + 1}{\theta^2 - \tau^2}, K'_\tau \right) \approx \min \left\{ \left\| \frac{\theta^2 + 1}{\theta^2 - \tau^2} + \frac{1}{\tau^2} \right\|, \left\| \frac{\theta^2 + 1}{\theta^2 - \tau^2} - 1 \right\|, \min_{i=2,\ldots,k} \left\{ \left\| \frac{\theta^2 + 1}{\theta_i^2 - \tau^2} - 1 \right\| \right\}
\]

Define \( c_\tau \) to be the ratio of the right-hand sides of (3.32) and the above quantity. Then we terminate the inner iterations of solving the correction equation (2.30) provided that

\[
\varepsilon \leq 2c_\tau \tilde{\varepsilon}.
\]

In computations, in order to guarantee that \( \mathcal{X} + \text{span}\{\mathcal{X}_+\} \) has some improvement over \( \mathcal{X} \), as a safeguard, we propose to take

\[
\varepsilon \leq \min\{2c_\tau \tilde{\varepsilon}, 0.01\}.
\]

(3.33)
However, $\epsilon = \frac{\|\tilde{r} - r\|}{\|\tilde{r}\|}$ is an a-priori error and uncomputable in practice, which makes us impossible to determine if (3.31) and (3.33) are met. As a practical alternative, denote by

$$\|r_{in}\| = \frac{1}{\|r_t\|} \|r - (I - \tilde{y}^T \tilde{x}) (A^T A - \theta^2 B^T B) (I - \tilde{x}^T \tilde{y}) \|$$

the relative residual norm of approximate solution $\tilde{r}$ of the correction equation (2.29), and by $\|r_{in, r}\|$ the relative residual norm of approximate solution $\tilde{r}$ of the correction equation (2.30). Then it is straightforward to justify that

$$\frac{\epsilon}{\kappa(M)} \leq \|r_{in}\| \leq \kappa(M) \epsilon \quad \text{and} \quad \frac{\epsilon}{\kappa(M_{\tau})} \leq \|r_{in, r}\| \leq \kappa(M_{\tau}) \epsilon,$$  \hspace{1cm} (3.34)

where $M$ and $M_{\tau}$ are $A^T A - \theta^2 B^T B$ and $A^T A - \tau^2 B^T B$ restricted to $\tilde{y}^\perp$ and map $\tilde{y}^\perp$ to $\tilde{x}^\perp$. The bounds in (3.34) motivate us to replace $\epsilon$ by $\|r_{in}\|$ and $\|r_{in, r}\|$ and stop the inner iterations of solving (2.29) and (2.30) when

$$\|r_{in}\| \leq 2\tilde{\epsilon} \quad \text{and} \quad \|r_{in, r}\| \leq \min\{2c_{\tau} \tilde{\epsilon}, 0.01\}$$  \hspace{1cm} (3.35)

for a given $\tilde{\epsilon}$. Provided that $\kappa(M)$ or $\kappa(M_{\tau})$ is modest, $\|r_{in}\|$ or $\|r_{in, r}\|$ is a reliable replacement of $\epsilon$, so that criterion (3.35) is robust. The asymptotic bounds for $\kappa(M)$ and $\kappa(M_{\tau})$ have been given in (3.7) and (3.15).

### 4 A Thick-Restart CPF-JDGSVD Algorithm with Deflation and Purgation

To be practical, we consider several key issues on CPF-JDGSVD and its extension, which include efficient implementations of thick-restart, deflation and purgation. The resulting algorithm can compute more than one GSVD components of $\{A, B\}$ efficiently and reliably.

#### 4.1 Thick-Restart

As the searching subspaces become large, the basis matrices $\tilde{U}$, $\tilde{V}$ and $\tilde{X}$ are large, and CPF-JDGSVD will be prohibitive due to the excessive computational complexity. A common approach is to restart the basic algorithm after a maximum subspace dimension $k_{\max}$ is reached. We will adapt the thick-restart technique [25] for the eigenvalue problem to our CPF-JDGSVD method. A main ingredient is to retain minimal $k_{\min}$ dimensional left and right searching subspaces for restart. We next present a novel approach to achieving thick-restart in our method effectively and efficiently.

At the extraction stage, let the GSVD of $\{G, H\}$ be partitioned as

$$\begin{align*}
(\Sigma_G, \Sigma_H, E, F, D) &= \left([\Sigma_{G,1} \Sigma_{G,2}], \begin{bmatrix} \Sigma_{H,1} & \Sigma_{H,2} \end{bmatrix}, [E_1, E_2], [F_1, F_2], [D_1, D_2]\right),
\end{align*}$$  \hspace{1cm} (4.1)

such that $(\Sigma_{G,1}, \Sigma_{H,1}, E_1, F_1, D_1)$ is the partial GSVD associated with the $k_{\min}$ generalized singular values of $\{G, H\}$ closest to the target $\tau$, i.e.,

$$GD_1 = E_1 \Sigma_{G,1} \quad \text{and} \quad HD_1 = F_1 \Sigma_{H,1}.$$  \hspace{1cm} (4.2)

Let the new starting right searching subspace, denoted by $\mathcal{Y}_{\text{new}}$ be spanned by the columns of $\tilde{X} D_1$. Then the corresponding starting left searching subspaces, denoted by $\mathcal{X}_{\text{new}}$ and $\gamma_{\text{new}}$, are spanned by the columns of $A \tilde{X} D_1$ and $B \tilde{X} D_1$, respectively. Let $D_1 = Q_{1} R_{1}$ be the thin QR factorization of $D_1$, whose computation costs $\mathcal{O}(k_{\max} k_{\min}^2)$ flops. Then the columns of
\[ \tilde{X}_{\text{new}} = \tilde{X} Q_r \] form an orthonormal basis of \( \mathcal{X}_{\text{new}} \), and its computation costs \( 2nk_{\text{max}}k_{\text{min}} \) flops. Combining this with (2.7) and (4.2), we obtain the thin QR factorizations:

\[
\begin{align*}
A \tilde{X}_{\text{new}} &= A \tilde{X} Q_r = \tilde{U} G D_1 R_{\tau}^{-1} = \tilde{U} E_1 \cdot \Sigma_{G,1} R_{\tau}^{-1} = \tilde{U}_{\text{new}} G_{\text{new}}, \\
B \tilde{X}_{\text{new}} &= B \tilde{X} Q_r = \tilde{V} H D_1 R_{\tau}^{-1} = \tilde{V} F_1 \cdot \Sigma_{H,1} R_{\tau}^{-1} = \tilde{V}_{\text{new}} H_{\text{new}},
\end{align*}
\]

where the columns of \( \tilde{U} E_1 \) and \( \tilde{V} F_1 \) are orthonormal, and \( \Sigma_{G,1} R_{\tau}^{-1} \) and \( \Sigma_{H,1} R_{\tau}^{-1} \) are upper triangular. The computations of \( \tilde{U}_{\text{new}} \) and \( \tilde{V}_{\text{new}} \) cost \( 2mk_{\text{max}}k_{\text{min}} \) and \( 2pk_{\text{max}}k_{\text{min}} \) flops, and those of \( G_{\text{new}} \) and \( H_{\text{new}} \) consume \( \mathcal{O}(k_{\text{min}}^3) \) flops, respectively. As a result, the main cost of thick-restart is \( \mathcal{O}(\max\{m,p\}k_{\text{max}}) \) as \( k_{\text{min}} = \mathcal{O}(1) \).

Rewriting \( \tilde{U}, \tilde{V}, \tilde{X} \) and \( G, H \) as \( \tilde{U}_{\text{new}}, \tilde{V}_{\text{new}}, \tilde{X}_{\text{new}} \) and \( G_{\text{new}}, H_{\text{new}} \), respectively, we expand the subspaces in a regular way until they reach the dimension \( k_{\text{max}} \) or the algorithm converges. In such a way, we have developed a thick-restart CPF-JDGSVD algorithm.

### 4.2 Deflation

Suppose that we are interested in the GSVD components \((\sigma_i, u_i, v_i, x_i)\) of \( \{A, B\} \) with \( \sigma_i \) closest to \( \tau, i = 1, \ldots, \ell \). We aim to introduce an appropriate deflation technique into the thick-restart CPF-JDGSVD algorithm for such purpose. The following result is straightforward to justify, and forms the basis of our deflation technique. We need to consider several nontrivial issues, including modifications of the correction equations and their right-hand sides and the least solution accuracy requirements \( \varepsilon \) on the inner iterations, etc.

**Proposition 4.1** For integer \( 1 \leq j < \ell \), let the partial GSVD

\[
(C_j, S_j, U_j, V_j, X_j) = \left(\begin{array}{ccc}
\alpha_1 & \cdots & \alpha_j \\
\beta_1 & \cdots & \beta_j
\end{array}\right), \left[\begin{array}{c}
u_1, \ldots, u_j \\
v_1, \ldots, v_j \\
x_1, \ldots, x_j
\end{array}\right],
\]

of \( \{A, B\} \) be defined by (1.1) and (1.2), and define

\[
Y_j = (A^T A + B^T B) X_j = A^T U_j C_j + B^T V_j S_j.
\]

Then \((\alpha_i, \beta_i, u_i, v_i, x_i), i = j + 1, \ldots, q \) are the nontrivial GSVD components of the matrix pair

\[
\{A_j, B_j\} := \{A(I - X_j Y_j^T), B(I - X_j Y_j^T)\}
\]

restricted to the orthogonal complement of \( \text{span}\{Y_j\} \).

Assume that \((\tilde{\alpha}_i, \tilde{\beta}_i, \tilde{u}_i, \tilde{v}_i, \tilde{x}_i), i = 1, 2, \ldots, j \) are converged approximations to the GSVD components \((\alpha_i, \beta_i, u_i, v_i, x_i)\) of \( \{A, B\} \) that satisfy the stopping criteria

\[
\|r_i\| = \|\tilde{\beta}_i A^T \tilde{u}_i - \tilde{\alpha}_i B^T \tilde{v}_i\| \leq (\tilde{\beta}_i \|A\|_1 + \tilde{\alpha}_i \|B\|_1) \cdot \text{tol}, \quad i = 1, \ldots, j.
\]

Then

\[
(C_c, S_c, U_c, V_c, X_c) = \left(\begin{array}{ccc}
\tilde{\alpha}_1 & \cdots & \tilde{\alpha}_j \\
\tilde{\beta}_1 & \cdots & \tilde{\beta}_j
\end{array}\right), \left[\begin{array}{c}	ilde{u}_1, \ldots, \tilde{u}_j \\
\tilde{v}_1, \ldots, \tilde{v}_j \\
\tilde{x}_1, \ldots, \tilde{x}_j
\end{array}\right],
\]

is a converged approximation to the partial GSVD \((C_j, S_j, U_j, V_j, X_j)\) of \( \{A, B\} \) that satisfies

\[
A^T U_c S_c - B^T V_c C_c \|_F = \|r_1, \ldots, r_j\|_F \leq \sqrt{j(\|A\|_1^2 + \|B\|_1^2)} \cdot \text{tol},
\]

\( \odot \) Springer
where the last inequality holds since \( \|r_j\| \leq \sqrt{\|A\|^2_1 + \|B\|^2_1} \cdot \text{tol} \) from (4.6).

Write

\[
Y_c = (A^T A + B^T B)X_c = A^T U_c C_c + B^T V_c S_c.
\]

Then \( Y_c^T X_c = X_c^T (A^T A + B^T B)X_c = I \), and \( I - X_c Y_c^T \) is an oblique projector onto the orthogonal complement of span\(\{Y_c\}\). Proposition 4.1 indicates that, in order to compute the next GSVD component \((\alpha, \beta, u, v, x) := (\alpha_{j+1}, \beta_{j+1}, u_{j+1}, v_{j+1}, x_{j+1})\) of \(\{A, B\}\), one can apply CPF-JDGSVD to the matrix pair

\[
(\tilde{A}_j, \tilde{B}_j) = (A(I - X_c Y_c^T), B(I - X_c Y_c^T)).
\]

Let \((\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x})\) be the current approximate GSVD component of \(\{A, B\}\) that is obtained by applying CPF-JDGSVD to \((\tilde{A}_j, \tilde{B}_j)\) and satisfies (2.6). Suppose that it does not yet converge to \((\alpha, \beta, u, v, x)\). Then at the expansion stage, depending on if criterion (2.31) is met, we formally solve the correction equation

\[
(I - \tilde{\gamma}\tilde{x}^T)(\tilde{A}_j^T \tilde{A}_j - \theta^2 \tilde{B}_j^T \tilde{B}_j)(I - \tilde{x}\tilde{y}^T)t = -\tilde{r} \quad \text{for} \quad t \perp \text{span}\{\tilde{y}, Y_c\}
\]

or

\[
(I - \tilde{\gamma}\tilde{x}^T)(\tilde{A}_j^T \tilde{A}_j - \tau^2 \tilde{B}_j^T \tilde{B}_j)(I - \tilde{x}\tilde{y}^T)t = -\tilde{r} \quad \text{for} \quad t \perp \text{span}\{\tilde{y}, Y_c\},
\]

where \(\tilde{y}\) is defined by (2.24) and \(\tilde{r}\) is the residual of \((\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x})\) as an approximate GSVD component of \((\tilde{A}_j, \tilde{B}_j)\):

\[
\tilde{r} = \tilde{\beta}\tilde{A}_j^T \tilde{u} - \tilde{\alpha}\tilde{B}_j^T \tilde{v} = (I - Y_c X_c^T)(\tilde{\beta}\tilde{A}_j^T \tilde{u} - \tilde{\alpha}\tilde{B}_j^T \tilde{v}) = (I - Y_c X_c^T)r.
\]

Since \(I - X_c Y_c^T\) is an oblique projector and \((\tilde{A}_j, \tilde{B}_j)\) is restricted to the orthogonal complement of span\(\{Y_c\}\), it is evident that \(\tilde{x} \perp\) span\(\{Y_c\}\) automatically when CPF-JDGSVD is applied to such matrix pair. Set \(X_p = \{X_c, \tilde{x}\} \) and \(Y_p = \{Y_c, \tilde{y}\} \). Then from \(Y_c^T X_c = I_j\) and the above, it is known that \(Y_p^T X_p = I_{j+1}\) and \(I - X_p Y_p^T\) are oblique projectors onto the orthogonal complements of span\(\{Y_p\}\) and span\(\{X_p\}\), respectively. By the definition (4.8) of \((\tilde{A}_j, \tilde{B}_j)\), we have

\[
(I - \tilde{\gamma}\tilde{x}^T)(\tilde{A}_j^T \tilde{A}_j - \theta^2 \tilde{B}_j^T \tilde{B}_j)(I - \tilde{x}\tilde{y}^T) = (I - Y_p X_p^T)(A^T A - \theta^2 B^T B)(I - X_p Y_p^T)
\]

and

\[
(I - \tilde{\gamma}\tilde{x}^T)(\tilde{A}_j^T \tilde{A}_j - \tau^2 \tilde{B}_j^T \tilde{B}_j)(I - \tilde{x}\tilde{y}^T) = (I - Y_p X_p^T)(A^T A - \tau^2 B^T B)(I - X_p Y_p^T).
\]

Therefore, together with (4.11), the correction equations (4.9) and (4.10) are now exactly

\[
(I - Y_p X_p^T)(A^T A - \theta^2 B^T B)(I - X_p Y_p^T)t = -(I - Y_c X_c^T)r \quad \text{for} \quad t \perp Y_p
\]

and

\[
(I - Y_p X_p^T)(A^T A - \tau^2 B^T B)(I - X_p Y_p^T)t = -(I - Y_c X_c^T)r \quad \text{for} \quad t \perp Y_p,
\]

respectively. Let \(\hat{t}\) be an approximate solution of the correction equation (4.12) or (4.13) obtained by some iterative solver, and set \(t := (I - X_p Y_p^T)\hat{t}\). Since \((I - X_p Y_p^T)^2 = I - X_p Y_p^T\), it is easy to see that the residual norms of \(\hat{t}\) and \(\tilde{t}\) as approximate solutions of (4.12) or (4.13) are identical. Clearly, by the definition of the oblique \(I - X_p Y_p^T\), we have \(\tilde{t} \perp Y_p\).

Next we show that we can always perform CPF-JDGSVD only on \(A\) and \(B\) rather than on \(\tilde{A}_j\) and \(\tilde{B}_j\) explicitly. This is particularly desirable because of its higher computational efficiency. Assume that the current searching subspace \(\mathcal{X}\) is \((A^T A + B^T B)\)-orthogonal to the
converged approximate singular vectors $\tilde{x}_1, \ldots, \tilde{x}_j$, i.e., $V_j \perp \text{span}\{Y_c\}$. Then $A_j X_j = \Sigma_0 \tilde{A}_j$ and $B_j B_j = \Sigma_0$, $V_j = B_j X_j$. Our task is to prove that, under this assumption, the expanded subspace $V_{\text{new}}$ at each outer iteration meets this orthogonality too, so that $A_j V_{\text{new}} = \Sigma_0 \tilde{A}_j$ and $B_j V_{\text{new}} = \Sigma_0 B_j V_{\text{new}}$. Suppose at this moment that we have three $k$-dimensional subspaces $V_j$ and $V_j = A_j V_j$, $V_j = B_j V_j$ at hand, where $V_j$ satisfies the orthogonality requirement $V_j \perp \text{span}\{Y_c\}$ of $V_{\text{new}}$. Obviously, it suffices to maintain this orthogonality as $V_{\text{new}}$ is expanded. Let the columns of $\tilde{X}$ be an orthonormal basis of $V_j$. With the approximate solution $\tilde{X}$ of either (4.13) or (4.12) that satisfies $\tilde{X} \perp Y_p$, we orthonormalize it against $\tilde{X}$ to generate the expansion vector $\tilde{x}_+ = \frac{(I - \tilde{X} \tilde{X}^T) \tilde{X}^T}{\| (I - \tilde{X} \tilde{X}^T) \|}$ and update $\tilde{X}_{\text{new}} := [\tilde{X}, \tilde{x}_+]$. Since $\tilde{X} \perp Y_c$ and $\tilde{X} \perp Y_c$, we have $\tilde{x}_+ \perp Y_c$ and thus the new right searching subspace $V_{\text{new}} = \text{span}\{\tilde{x}_{\text{new}}\} \perp Y_c$. Moreover, by (4.8), we have $A_j \tilde{X}_{\text{new}} = A_j \tilde{X}_{\text{new}}$ and $B_j \tilde{X}_{\text{new}} = B_j \tilde{X}_{\text{new}}$. Therefore, in computations, we form the orthonormal basis matrices of the two new left searching subspace by updating the QR factorizations $A \tilde{X}_{\text{new}} = \tilde{U}_{\text{new}} G_{\text{new}}$ and $B \tilde{X}_{\text{new}} = \tilde{V}_{\text{new}} H_{\text{new}}$. We then compute a new approximation to the desired $(\alpha, \beta, u, v, x)$ with respect to $V_{\text{new}}$ and $V_{\text{new}} = \text{span}\{U_{\text{new}}\}$, $V_{\text{new}} = \text{span}\{V_{\text{new}}\}$.

Based on the above, provided that the starting one-dimensional $V_j$ satisfies the orthogonality requirement that $V_j \perp \text{span}\{Y_c\}$, we can always perform CPF-JDGSVD on $A$ and $B$ rather than on $A_j$ and $B_j$ when computing $\ell$ GSVD components of $(A, B)$. As we will see, in Algorithm 1 to be proposed, the starting subspace $V_j$ for computing the $(j + 1)$th $x$ is the one that purges the currently converged $x_j$ from the current subspace, which itself satisfies $V_j \perp \text{span}\{Y_c\}$ automatically; see the next subsection. Consequently, the orthogonality requirement $V_j \perp \text{span}\{Y_c\}$ is always met at each outer iteration.

Once $(\alpha, \tilde{V}, u, v, x)$ has converged in the sense of (2.13), we add it to the already converged partial GSVD $(C, S, U, V, X, Y)$, and set $j := j + 1$ until $j = \ell$.

Finally, we make comments on the ideal case that $(C, S, U, V, X, Y) = (C_j, S_j, U_j, V_j, X_j, Y_j)$, that is, $tol = 0$ in (4.6) and we have computed the $j$ desired GSVD components exactly, following the same derivations as those in Sects. 3.2–3.3, we can directly obtain (3.23) and (3.29) for the accuracy $\tilde{e}$ of the expansion vectors and for the solution accuracy $e$ of (4.13) and (4.12) with $K' = X_j^T K X_j$ and $K_j' = X_j^T K_j X_j$, respectively, where the columns of $X_j$ form an orthonormal basis of the orthogonal complement $X_j^\perp$ of $\text{span}\{X_j\}$. For $tol > 0$ in (4.6), a tedious but routine derivation shows that the new corresponding bounds in (3.23) and (3.29) are simply the counterparts established for $tol = 0$ plus $O(tol)$, and we omit details. Following the same discussions in Sects. 3.2–3.3, we stop the inner iterations when the inner relative residual norms $r_{in}$ and $r_{in, \tau}$ satisfy (3.35) for a given $\tilde{e}$.

### 4.3 Purification of the Converged $\tilde{x}_j$ from the Current $V_j$

It is remarkable that the current $V_j$ generally provides reasonably good information on the desired generalized singular vector(s) to be computed. Therefore, we should fully exploit $V_j$ by purging the newly converged $\tilde{x}_j$ from $V_j$ and retaining the resulting reduced subspace, denoted by $V_{\text{new}}$, as a good initial searching subspace for the next desired GSVD component. This motivates us to further improve the afore-described thick-restart CPF-JDGSVD algorithm with deflation so as to compute the $(j + 1)$th GSVD component more efficiently. We can achieve purification in the following efficient and numerically stable way.

Let $(\Sigma_G, \Sigma_H, E, F, D)$ be the GSVD of $(G, H)$ partitioned as (4.1) and

$$(\Sigma_{G,1}, \Sigma_{H,1}, E_1, F_1, D_1) = (\tilde{\alpha}, \tilde{\beta}, e, f, d)$$
be the GSVD component corresponding to the converged GSVD component \((\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x})\) of \((A, B)\). Since the columns of \(D\) are \((G^T G + H^T H)\)-orthonormal, we obtain
\[
D_2^T \tilde{X} \tilde{y} = D_2^T \tilde{X} (A^T A + B^T B) \tilde{x} = D_2^T \tilde{X} (A^T A + B^T B) \tilde{X} d = D_2 (G^T G + H^T H) d = 0,
\]
proving that the columns of \(\tilde{X} D_2\) are orthogonal to \(\tilde{y}\). Therefore, \(\mathcal{X}_{\text{new}} = \text{span}\{\tilde{X} D_2\}\). Let \(D_2 = Q_d R_d\) be the thin QR factorization of \(D_2\). Then the columns of \(\tilde{X}_{\text{new}} = \tilde{X} Q_d\) form an orthonormal basis of \(\mathcal{X}_{\text{new}}\), so that from (4.1) we can obtain thin QR factorizations:
\[
A \tilde{X}_{\text{new}} = A \tilde{X} Q_d = \tilde{U} G D_2 R_d^{-1} = \tilde{U} E_2 \cdot \Sigma_{G,2} R_d^{-1} = \tilde{U} G_{\text{new}},
\]
\[
B \tilde{X}_{\text{new}} = B \tilde{X} Q_d = \tilde{V} H D_2 R_d^{-1} = \tilde{V} F_2 \cdot \Sigma_{H,2} R_d^{-1} = \tilde{V} H_{\text{new}},
\]
where the columns of \(\tilde{U}_{\text{new}}\) and \(\tilde{V}_{\text{new}}\) form orthonormal bases of the new initial left searching subspaces \(\mathcal{X}_{\text{new}}\) and \(\mathcal{Y}_{\text{new}}\).

### 4.4 The Thick-Restart CPF-JDGSVD Algorithm with Deflation and Purgation: A Pseudocode

For the given target \(\tau\), the thick-restart CPF-JDGSVD algorithm with deflation and purgation requires the devices to compute \(A^T u, B^T v\) and \(A x, B x\) for arbitrary vectors \(u, v\) and \(x\), a unit-length starting vector \(x_0\) to generate one-dimensional \(\mathcal{X}\), the target \(\tau\), the number \(\ell\) of the desired GSVD components, and the convergence tolerance \(\text{tol}\). It outputs a converged approximation \((C_{\ell}, S_{\ell}, U_{\ell}, V_{\ell}, X_{\ell})\) to the desired GSVD \((C, S, U, V, X)\) of \((A, B)\) associated with the \(\ell\) generalized singular values closest to the target \(\tau\). The other parameters are the minimum and maximum dimensions \(k_{\text{min}}\) and \(k_{\text{max}}\), the number of searching subspaces, the switching tolerance \(\text{fixtol}\) from (2.30) to (2.29), and the solution accuracy \(\varepsilon\) of the expansion vectors in (3.35). By defaults, we set \(k_{\text{min}} = 3, k_{\text{max}} = 30, \text{fixtol} = 10^{-4}\), and \(\varepsilon = 10^{-3}\). Algorithm 1 sketches our thick-restart CPF-JDGSVD algorithm with deflation and purgation.

### 5 Numerical Examples

We report numerical results to illustrate the robustness, accuracy and overall efficiency of Algorithm 1. The experiments were performed on an Intel (R) Core (TM) i7-7700 CPU 3.60 GHz with the main memory 8 GB and 4 cores using the MATLAB R2020b with the machine precision \(\epsilon_{\text{mach}} = 2.22 \times 10^{-16}\) under the Windows 10 64-bit system.

Table 1 lists the test matrix pairs with some of their basic properties, where we use matrices from the SuiteSparse Matrix Collection [5] or their transposes, denoted by the matrix names with the superscript \(T\), as our test matrices \(A\), so as to ensure \(m \geq n\), and the matrices \(B\) are the \(n \times n\) tridiagonal Toeplitz matrix \(B_0\) with 3 and 1 being the main and off diagonal elements and the \((n - 1) \times n\) scaled discrete approximation \(B_1\) of the first order derivative operator of dimension one [9]:

\[
B_0 = \begin{bmatrix}
3 & 1 & & \\
1 & \ddots & \ddots & \\
& \ddots & \ddots & 1 \\
& & 1 & 3
\end{bmatrix}
\quad \text{and} \quad
B_1 = \begin{bmatrix}
1 & -1 & & \\
& \ddots & \ddots & \\
& & 1 & -1
\end{bmatrix}.
\]

(5.1)
Algorithm 1 The thick-restart CPF-JDGSVD with deflation and purgation for the target $\tau$.

1: Initialization: Set $k = 1$, $k_c = 0$, $C_c = [ ]$, $S_c = [ ]$, $U_c = [ ]$, $V_c = [ ]$, $X_c = [ ]$ and $Y_c = [ ]$. Let $\bar{U} = [ ]$, $\bar{V} = [ ]$, $\bar{X} = [ ]$, and $x_+ = x_0$.

2: while $k \geq 0$ do
3:     Set $\bar{X} = [\bar{X}, x_+]$, and update the QR factorizations $A\bar{X} = \bar{U} G$ and $B\bar{X} = \bar{V} H$.
4:     Compute the GSVD of $\{G, H\}$, label the generalized singular values in increasing order according to their distances from the target $\tau$, and pick up $(\bar{u}, \bar{v}, \bar{e}, f, d)$ with the generalized singular value $\theta = \frac{\bar{u}}{\bar{v}}$ closest to $\tau$.
5:     Compute the approximate generalized singular vectors $\tilde{u} = \tilde{U} e$, $\tilde{v} = \tilde{V} f$, $\tilde{x} = \tilde{X} d$, the residual $r = \bar{b} A^T \tilde{u} - \bar{a} B^T \tilde{v}$, and $\tilde{y} = \bar{a} A^T \tilde{u} + \bar{b} B^T \tilde{v}$.
6:     if $||r|| \leq (\bar{b} ||A||_1 + \bar{a} ||B||_1) \cdot \text{fixtol}$ then
7:         (i) set $k_c = k_c + 1$ and update $C_c = [C_c \bar{a}]$, $S_c = [S_c \bar{b}]$, $U_c = [U_c \tilde{u}]$, $V_c = [V_c \tilde{v}]$, $X_c = [X_c \tilde{x}]$, and $Y_c = [Y_c \tilde{y}]$; (ii) if $k_c = \ell$ then return $(C_c, S_c, U_c, V_c, X_c)$ and stop; (iii) set $k = k - 1$, purge $\tilde{x}$ from the current $\mathcal{X}$, obtain the reduced subspaces, and go to step 3.
8:     end if
9: end while

Table 1 Properties of the test matrix pairs $\{A, B\}$, where $B_0$ and $B_1$ are defined by \text{(5.1)}, ‘tmgpc1’, ‘wstn_1’ and ‘flower54’ are abbreviations of ‘tomographic1’, ‘waston_1’, and ‘flower5_4’, respectively, $\text{nnz}$ is the total number of nonzero entries in $A$ and $B$, and $\sigma_{\text{max}}$ and $\sigma_{\text{min}}$ are the largest and smallest generalized singular values of $\{A, B\}$, respectively.

| $A$   | $B$   | $m$   | $n$   | $p$   | $\text{nnz}$ | $\kappa([A\ B])$ | $\sigma_{\text{max}}$ | $\sigma_{\text{min}}$ |
|-------|-------|-------|-------|-------|--------------|-------------------|----------------------|----------------------|
| r05T  | B0    | 9690  | 5190  | 5190  | 119713       | 13.2              | 17.4                 | 3.10e−2              |
| deter4T | B0    | 9133  | 3235  | 3235  | 28934        | 7.07              | 8.56                 | 5.62e−3              |
| lp_bnl2T | B0    | 4486  | 2324  | 2324  | 21966        | 1.93e+2           | 1.10e+2              | 1.20e−2              |
| largeT | B0    | 8617  | 4282  | 4282  | 33479        | 3.53e+3           | 2.40e+3              | 2.25e−3              |
| gemat1T | B0    | 10595 | 4929  | 4929  | 61376        | 1.82e+4           | 1.21e+4              | 5.97e−5              |
| tmgpc1 | B0    | 73159 | 59498 | 59498 | 825987       | 7.94              | 2.78                 | 0                    |
| wstn_1T | B0    | 386992| 201155| 201155| 1658556      | 15.1              | 11.1                 | 5.19e−3              |
| degmeT | B0    | 659415| 185501| 185501| 8684029      | 2.04e+2           | 9.24e+2              | 1.24                 |
| slptskT | B1    | 3347  | 2861  | 2860  | 78185        | 3.29e+2           | 1.16e+4              | 1.81e−1              |
| rosen10T | B1    | 6152  | 2056  | 2055  | 68302        | 1.57e+2           | 2.02e+4              | 1.24                 |
| flower54T | B1    | 14721 | 5226  | 5225  | 54392        | 4.52              | 7.73e+3              | 2.42e−1              |
| l30T   | B1    | 16281 | 2701  | 2700  | 57470        | 7.76              | 2.53e+3              | 2.22e−3              |
| cq5T   | B1    | 11748 | 5048  | 5047  | 61665        | 8.91e+3           | 7.34e+4              | 3.78e−2              |
| stat96v5T | B1    | 75779 | 2307  | 2306  | 238533       | 31.7              | 5.55e+3              | 0                    |
In order to verify the performance of CPF-JDGSVD, we have used the MATLAB built-in functions gsvd and eig to compute the GSVD of the first five and last six test problems for \( m \approx n \) roughly and \( m \gg n \), respectively, with eig applied to \( \{A^T A, B^T B\} \). For the three large matrix pairs \( \{A, B\} = \{\text{tmgpc1}, B_0\}, \{\text{wstn}_1^T, B_0\} \) and \( \{\text{degme}_1^T, B_0\} \), we have applied the MATLAB built-in function svds to \( \{\tilde{A}, \tilde{B}\} \) to compute its largest and smallest singular values, and obtained \( \kappa \left( \frac{A}{\tilde{B}} \right) \). We have applied the MATLAB built-in function eigs to \( \{A^T A, B^T B\} \) to compute the largest and smallest generalized singular values of \( \{A, B\} \). Particularly, the smallest generalized singular values \( \sigma_{\min} = 0 \) of the matrix pairs \( \{\text{tmgpc1}, B_0\} \) and \( \{\text{stat96v5}^T, B_1\} \) since tmgpc1 and stat96v5 are known to be rank deficient [5].

For each matrix pair \( \{A, B\} \) with a given target \( \tau \), we compute the GSVD components of \( \{A, B\} \) corresponding to the \( \ell \) generalized singular values closest to \( \tau \), where \( \ell = 1, 5, \) and 9. We take the initial vector \( x_0 \) for the problems with the full column rank \( B \) to be \( \frac{1}{\sqrt{n}} \{1, 1, \ldots, 1\}^T \), and \( x_0 \) for the problems with \( B \) rank deficient to be the unit-length vector whose primitive \( i \)-th element is \( i \mod 4 \), \( i = 1, \ldots, n \). An approximate GSVD component \( (\tilde{a}, \tilde{B}, \tilde{u}, \tilde{v}, \tilde{x}) \) is claimed to have converged if its residual norm satisfies \( (4.6) \) with \( \text{tol} = 10^{-10} \). For the inner iterations, we use the unpreconditioned MINRES to solve the correction equation (4.12) or (4.13), where the code minres is from MATLAB R2020b. We always take the initial approximate solutions to be zero vectors, and stop the inner iterations when the stopping criterion (3.35) is fulfilled for a fixed \( \tilde{\varepsilon} \). Unless specified otherwise, we take the parameters in CPF-JDGSVD to be the defaults in Sect. 4.4.

As a comparison, we also compute the desired GSVD components of \( \{A, B\} \) with \( B \) of full column rank using the JDGSVD algorithm [10] with the parameters the same as in the thick-restart CPF-JDGSVD as far as possible. We always take \( u_0 = \frac{1}{\sqrt{m}} \{1, \ldots, 1\}^T \) as the initial left vector for JDGSVD, which works on the generalized eigenvalue problem of the matrix pair \( \{\left[ A^T A \right], \left[ I_{B^T B} \right] \} \) for \( B \) of full column rank. At each step, JDGSVD computes an approximation \( (\vartheta, \hat{u}, \hat{w}) \) to the desired triplet \( (\sigma, u, w := x/\beta) \). In our implementations, an approximate \( (\vartheta, \hat{u}, \hat{w}) \) is claimed to have converged if the residual

\[
 r = \begin{bmatrix} A \hat{w} - \vartheta \hat{u} \\ A^T \hat{u} - \vartheta B^T B \hat{w} \end{bmatrix} = \left( \begin{bmatrix} A^T A \\ I_{B^T B} \end{bmatrix} - \vartheta \begin{bmatrix} I \\ B^T B \end{bmatrix} \right) \begin{bmatrix} \hat{u} \\ \hat{w} \end{bmatrix}
\]

of the approximate generalized eigenpair \( (\vartheta, (\hat{u}^T, \hat{w}^T)^T) \) satisfies

\[
 \text{relres} := \frac{\|r\|}{(\|A\|_1 + \vartheta \|B^T B\|_1)\sqrt{1 + \|\hat{w}\|^2}} \leq \text{tol}
\]

with \( \text{tol} \) the same as in (2.13). For the correction equations (cf. the equation after equation (13) in [10]), we take \( \vartheta = \tau \) in the initial steps and then switch to the approximate generalized singular value \( \vartheta \) when the relative residual norm \( \text{relres} \) is smaller than the same switching tolerance \( \text{fixtol} \) in CPF-JDGSVD. We always take zero vector as the initial guess and use minres to solve the symmetric correction equations until the relative residual norm of the inner iterations is smaller than \( 2\tilde{\varepsilon} \), where \( \tilde{\varepsilon} \) is the same as in (3.31) and (3.33). To make a fair comparison, we have introduced thick-restart and deflation technique similar to those described in Sect. 4 into JDGSVD for computing \( \ell \) GSVD components of \( \{A, B\} \). Once a triplet \( (\vartheta, \hat{u}, \hat{w}) \) has converged, the converged GSVD component is recovered by

\[
 \left( \tilde{a}, \tilde{B}, \hat{u}, \hat{v}, \tilde{x} \right) = \left( \frac{\vartheta}{\sqrt{1 + \vartheta^2}}, \frac{1}{\sqrt{1 + \vartheta^2}}, \hat{u}, \frac{1}{\|B \hat{w}\|} B \hat{w}, \frac{1}{\sqrt{\|A \hat{w}\|^2 + \|B \hat{w}\|^2}} \hat{w} \right).
\]
As an approximation to the desired GSVD component \((\alpha, \beta, u, v, x)\) of \(\{A, B\}\), for the original GSVD problem, an associated true relative residual norm of \((\tilde{\alpha}, \tilde{\beta}, \tilde{u}, \tilde{v}, \tilde{x})\) is

\[
\text{relrest}_i = \frac{\|A\tilde{x} - \tilde{\alpha}\hat{u}\|}{\|A\|_1\|\tilde{x}\| + \tilde{\alpha}} + \frac{\|B\tilde{x} - \tilde{\beta}\hat{v}\|}{\|B\|_1\|\tilde{x}\| + \tilde{\beta}} + \frac{\|\tilde{\beta}A^T\hat{u} - \tilde{\alpha}B^T\hat{v}\|}{\tilde{\beta}\|A\|_1 + \tilde{\alpha}\|B\|_1}.
\] (5.3)

In all the tables, we denote by \(I_{out}\) and \(I_{in}\) the total numbers of outer and inner iterations, and by \(T_{cpu}\) the CPU time in seconds counted by the MATLAB commands tic and toc.

**Experiment 5.1** We compute the GSVD components of \(\{A, B\} = \{r05^T, B_0\}\) with \(\ell = 1, 5, 9\) corresponding to the generalized singular values closest to \(\tau = 4\) using Algorithm 1 with fixtol = \(+\infty, 10^{-2}, 10^{-4}\) and 0, respectively. Here fixtol = \(+\infty\) or 0 is a virtual value and means that we always solve the modified correction equation (4.12) or (4.13) only. The desired generalized singular values of \(\{A, B\}\) are clustered interior ones. We aim to show that fixtol = \(10^{-4}\) is a very best choice for the robustness and overall efficiency of Algorithm 1.

Table 2 reports the results. Clearly, for the three \(\ell\), CPF-JDGSVD with fixtol = \(+\infty\) uses much more outer and inner iterations and much more CPU time to converge than it does for the other three fixtol. What is worse, none of the converged generalized singular values \(\sigma_{c,i} \in [1.93, 2.17]\) is a desired one since the \(\ell\) desired \(\sigma_i \in [3.68, 4.47]\), meaning that the algorithm misconverges. CPF-JDGSVD with fixtol = \(10^{-2}\) is successful but has some irregular behavior: The first converged generalized singular value \(\sigma_{c,1} \approx 4.12\) is not the closest to \(\tau\) but the second converged \(\sigma_{c,2} \approx 3.92\) is. In contrast, CPF-JDGSVD with fixtol = 0 converges correctly, but it uses more outer iterations than CPF-JDGSVD with fixtol = \(10^{-2}\). CPF-JDGSVD with fixtol = \(10^{-4}\) works reliably and uses much fewer outer iterations than it does with fixtol = 0 for \(\ell = 5\) and 9; it also computes the desired GSVD components in the correct order. Obviously, with an improper larger or smaller fixtol, CPF-JDGSVD may either misconverge or converge slowly. A good choice of fixtol must guarantee the reliability of the computed GSVD components and, meanwhile, should reduce the total computational costs as much as possible. Such a choice is obviously problem dependent. In order to make the algorithm reliable and efficient, one should take a fairly small fixtol.

We have also observed the same phenomena on other test matrix pairs. The experiments have indicated that, for the reliability and overall efficiency of CPF-JDGSVD, fixtol = \(10^{-4}\) is a good choice and can be used as a default.

**Experiment 5.2** We compute the GSVD components of \(\{A, B\} = \{\text{deter}4^T, B_0\}\) with \(\ell = 1, 5, 9\) corresponding to the clustered interior generalized singular values closest to \(\tau = 0.08\) using Algorithm 1 with \(\tilde{\epsilon} = 10^{-5}, 10^{-4}, 10^{-15}\) in (3.35), where \(\tilde{\epsilon} = 10^{-15}\) means that all the correction equations have been solved exactly in finite precision arithmetic. For the experimental purpose, we have also used the so-called “exact” CPF-JDGSVD algorithm to...
Table 3 \( \{A, B\} = \{\text{deter}^4 T, B_0\} \) with \( \tau = 0.08 \)

| \( \tilde{\epsilon} \) | \( \ell = 1 \) | \( \ell = 5 \) | \( \ell = 9 \) |
|---|---|---|---|
|   | \( t_{\text{out}} \) | \( t_{\text{in}} \) | \( t_{\text{cpu}} \) | \( t_{\text{out}} \) | \( t_{\text{in}} \) | \( t_{\text{cpu}} \) | \( t_{\text{out}} \) | \( t_{\text{in}} \) | \( t_{\text{cpu}} \) |
| \( 10^{-3} \) | 25 | 2082 | 0.29 | 43 | 8790 | 1.02 | 58 | 13711 | 2.03 |
| \( 10^{-4} \) | 25 | 2586 | 0.27 | 38 | 8494 | 0.93 | 56 | 13748 | 2.03 |
| \( 10^{-15} \) | 24 | 7192 | 0.72 | 42 | 19199 | 2.00 | 55 | 25972 | 3.36 |
| exact | 24 | – | 2.04 | 42 | – | 3.32 | 55 | – | 4.22 |

Fig. 1 Computing the GSVD components of \( \{A, B\} = \{\text{deter}^4 T, B_0\} \) corresponding to the nine generalized singular values closest to \( \tau = 0.08 \)

compute the desired GSVD components, where “exact” means, as indicated by (3.17) and (3.18), that the correction equations (4.12) and (4.13) are solved by the LU factorizations of \( L = A^T A - \tau^2 B^T B \) and \( L_{\tau} = A^T A - \tau^2 B^T B \), respectively. We aim to illustrate that \( \tilde{\epsilon} \in [10^{-4}, 10^3] \) is a very best choice for the robustness and overall efficiency of Algorithm 1.

Table 3 reports the results, and Fig. 1 depicts the convergence curves of CPF-JDGSVD with \( \tilde{\epsilon} = 10^{-3}, 10^{-4} \) and the “exact” CPF-JDGSVD algorithm for computing nine GSVD components of \( \{A, B\} \). Since CPF-JDGSVD computes the GSVD components one by one, its convergence curve has nine different stages for each \( \tilde{\epsilon} \), each of which indicates the convergence process of one GSVD component. We observe from Table 3 and Fig. 1 that, regarding the outer iterations, for \( \ell = 1, 5, \) and 9, CPF-JDGSVD with \( \tilde{\epsilon} = 10^{-3} \) and \( 10^{-4} \) behaves much like its exact counterpart. Furthermore, we have found that, compared with the iterative exact CPF-JDGSVD, i.e., \( \tilde{\epsilon} = 10^{-15} \), the inexact CPF-JDGSVD algorithm costs only no more than 53% of total inner iterations and no more than 61% of total CPU time to compute the desired GSVD components.

Clearly, a smaller \( \tilde{\epsilon} \) is unnecessary since it cannot reduce outer iterations but increases the total cost substantially. Therefore, we will adopt the default \( \tilde{\epsilon} = 10^{-3} \) in CPF-JDGSVD.

Experiment 5.3 Taking \( \tilde{\epsilon} = 10^{-3} \) and \( \text{fixtol} = 10^{-4} \), we show the good performance of Algorithm 1 on the other six problems with \( B = B_0 \) in Table 1. The desired GSVD components of \( \{A_a, B_a\} = \{\text{lp_bln}2^T, B_0\} \), \( \{A_b, B_b\} = \{\text{large}^T, B_0\} \) and \( \{A_c, B_c\} = \{\text{gemat1}^T, B_0\} \) with the targets \( \tau_a = 20, \tau_b = 5 \) and \( \tau_c = 12 \) are all clustered interior ones, and those of the
Table 4  Results of Algorithm 1 on the problems with $B = B_0$ in Table 1

| $A$          | $\ell = 1$ |        | $\ell = 5$ |        | $\ell = 9$ |        |
|--------------|------------|--------|------------|--------|------------|--------|
|              | $I_{out}$  | $I_{in}$ | $T_{cpu}$  | $I_{out}$ | $I_{in}$ | $T_{cpu}$ |
| lp_bnl2$^T$  | 12         | 497    | 0.09       | 43      | 1841      | 0.28   |
| large$^T$    | 8          | 9528   | 1.71       | 52      | 76190     | 13.7   |
| gemat1$^T$   | 9          | 6900   | 1.90       | 28      | 22257     | 6.71   |
| tmgpc1$^T$   | 9          | 403    | 1.82       | 32      | 1234      | 6.31   |
| wstn_1$^T$   | 14         | 35982  | 5.29e+2    | 37      | 106212    | 1.90e+3|
| degme$^T$    | 9          | 92     | 5.16       | 117     | 1399      | 83.3   |

Table 5  Results of JDGSVD on the problems with $B = B_0$ in Table 1

| $A$          | $\ell = 1$ |        | $\ell = 5$ |        | $\ell = 9$ |        |
|--------------|------------|--------|------------|--------|------------|--------|
|              | $I_{out}$  | $I_{in}$ | $T_{cpu}$  | $I_{out}$ | $I_{in}$ | $T_{cpu}$ | $Relres_\ell$ |
| lp_bnl2$^T$  | 11         | 918    | 0.20       | 42      | 3023      | 0.78   | 73       | 5640       | 1.51       | 2.19e−10 |
| large$^T$    | 8          | 9786   | 2.80       | 31      | 71127     | 24.4   | 52       | 137692     | 49.7       | 2.08e−10 |
| gemat1$^T$   | 14         | 10846  | 5.59       | 36      | 24986     | 13.6   | 57       | 36553      | 20.4       | 1.75e−10 |
| tmgpc1       | 9          | 807    | 6.29       | 29      | 1987      | 17.2   | 51       | 4282       | 40.4       | 1.25e−9  |
| wstn_1$^T$   | 11         | 51244  | 2.17e+3    | 28      | 131417    | 6.40e+3| 50       | 206423     | 1.11e+4    | 4.25e−10 |
| degme$^T$    | 11         | 582    | 40.4       | 358     | 20673     | 1.75e+3| 1007     | 64200      | 6.79e+3    | 1.22e−9  |

Table 4 reports the results. We have observed very similar phenomena to those in Experiments 5.1–5.2. The table indicates that CPF-JDGSVD worked efficiently for computing both the interior and the extreme GSVD components of the test matrix pairs. Remarkably, we have seen that the outer iterations for $\ell = 5$ and 9 are only slightly more than those for $\ell = 1$ rather than roughly four or eight times more. These confirm the effectiveness of the purgation described in Sect. 4.3, and justify that the reduced initial subspaces $X_{new}$’s of purging the converged right generalized singular vectors from the current subspaces indeed retain rich information on the next desired right generalized singular vectors.

For these six problems, all the $B$ are well conditioned, and $\kappa(B) \approx 5$. We also apply the JDGSVD algorithm in [10] to these problems with all the parameters same as in the CPF-JDGSVD algorithm. Table 5 displays the results, where $Relres_\ell$ is the norm of the $\ell$-dimensional vector whose entries are the relative residual norms of the $\ell$ converged GSVD components defined by (5.3). As can be seen from Table 5, the relative residual norms of the converged approximate GSVD components computed by JDGSVD are very comparable to the stopping tolerance $10^{-10}$, as is expected since matrices $B$ are very well conditioned.

Comparing Table 4 with Table 5, we can see that CPF-JDGSVD uses very comparable outer iterations as JDGSVD for the first five matrix pairs but it is at least three times as fast as JDGSVD for the last problem with $\ell = 5$ and 9. Regarding the overall efficiency, CPF-JDGSVD uses fewer inner iterations or less than 52% of CPU time to compute the desired nine GSVD components of $\{A_b, B_b\}$, $\{A_c, B_c\}$ and $\{A_e, B_e\}$. It saves more than 33% of inner iterations and more than 63% of CPU time to compute all the desired GSVD components.
The accuracy of a converged matrix from \([5]\) and \(B\) to compute the GSVD components of the six problems with \(B = B_1\) in Table 2. We have also observed that CPF-JDGSVD is robust and computes the desired GSVD components in the correct order.

Experiment 5.4 Taking the defaults \(\bar{\epsilon} = 10^{-3} \) and fixtol = \(10^{-4}\), we use Algorithm 1 to compute the GSVD components of the six problems with \(B = B_1\) in Table 2. We write \(\{A_a, B_a\} = \{\text{slptsk}^T, B_1\}\), \(\{A_b, B_b\} = \{\text{rosen10}^T, B_1\}\), \(\{A_c, B_c\} = \{\text{flower54}^T, B_1\}\), \(\{A_d, B_d\} = \{\text{130}^T, B_1\}\), \(\{A_e, B_e\} = \{\text{cq5}^T, B_1\}\) and \(\{A_f, B_f\} = \{\text{stat96v5}^T, B_1\}\) with the targets \(\tau\) being \(\tau_a = 9\), \(\tau_b = 4\), \(\tau_c = 82\), \(\tau_d = 1\), \(\tau_e = 0.1\), and \(\tau_f = 4000\), respectively.

As is shown in Table 6, CPF-JDGSVD is efficient to compute all the desired GSVD components, i.e., the clustered interior ones of \(\{A_i, B_i\}, i = a, b, c, d\), the clustered smallest ones of \(\{A_e, B_e\}\), and the largest nontrivial ones of \(\{A_f, B_f\}\). We have also observed that CPF-JDGSVD is robust and computes the desired GSVD components in the correct order.

Experiment 5.5 We aim to compare the performance of CPF-JDGSVD and JDGSVD for computing the nine GSVD components of \(\{A, B\} = \{\text{blckhole}, B_1^T\}\) corresponding to the generalized singular values closest to \(\tau = 1000\), where blckhole is a 2132 \(\times\) 2132 sparse matrix from \([5]\) and \(B = B_1^T\) ensures that \(B\) has full column rank. The desired generalized singular values are the largest ones, and are well separated. We have also applied CPF-JDGSVD and JDGSVD to \(\{B, A\}\) with \(\tau = 10^{-3}\) and computed its nine GSVD components.

For this problem, both \(A\) and \(B\) are well conditioned with \(\kappa(A) = 4.17 \times 10^3\), \(\kappa(B) = 1.36 \times 10^3\), and \(\kappa([A, B]) = 26.6\). Table 7 displays the results, and Fig. 2a depicts the accuracy of the converged generalized singular vectors obtained by CPF-JDGSVD and JDGSVD, respectively. The accuracy of a converged \((\tilde{u}, \tilde{v}, \tilde{x})\) by CPF-JDGSVD is

\[
\epsilon(\tilde{u}, \tilde{v}, \tilde{x}) = \sqrt{\sin^2 \angle(\tilde{u}, u) + \sin^2 \angle(\tilde{v}, v) + \sin^2 \angle(\tilde{x}, x)} \tag{5.4}
\]
with the “exact” generalized singular vectors \((u, v, x)\) computed by \texttt{gsvd}; the accuracy of \((\hat{u}, \hat{v}, \hat{x})\) obtained by JDGSVD is defined similarly.

We see from Table 7 that CPF-JDGSVD is much more efficient than JDGSVD, and it uses less than 10% of outer and inner iterations and less than 4% of CPU time than the latter. We have observed that all the generalized singular values computed by the two algorithms are accurate with the relative errors lying in \([10^{-15}, 10^{-11}]\). However, Fig. 2a shows that all the desired generalized singular vectors computed by CPF-JDGSVD are very accurate by recalling that we used the stopping tolerance \(tol = 10^{-10}\), and they are two to nearly five orders more accurate than those computed by JDGSVD. Indeed, we see from Table 7 that the relative residual norms \(\text{Relres}_t\) of the converged GSVD components obtained by JDGSVD are approximately three and eight orders larger than those by CPF-JDGSVD for \(\{A, B\}\) and \(\{B, A\}\), respectively. This shows that transforming the GSVD problem into the generalized eigenvalue problem of \(\begin{bmatrix} A^T \quad B^T \end{bmatrix}, \begin{bmatrix} I \quad B^T B \end{bmatrix}\) or \(\begin{bmatrix} B^T \quad I \end{bmatrix}, \begin{bmatrix} I \quad A^T A \end{bmatrix}\) is not a general-purpose formulation since a numerically backward stable algorithm for the generalized eigenvalue problem cannot produce numerically backward stable approximate GSVD components of the GSVD problem when \(\kappa(B)\) or \(\kappa(A)\) is not small, as has been proved in [13].

Moreover, we have observed that CPF-JDGSVD computed the desired GSVD components of \(\{A, B\}\) one by one correctly, while JDGSVD only computed the first six desired GSVD components and then repeatedly computed the first one after the six ones had converged. This is because \(B^T B\) is quite ill conditioned with \(\kappa(B^T B) = 1.85 \times 10^6\) and the right searching subspace used in JDGSVD lost the numerical \(B^T B\)-orthogonality to the converged right generalized singular subspace, which causes that the information on the converged GSVD component reappeared and the same GSVD component was computed repeatedly.

For \(\{B, A\}\), we have found that JDGSVD successfully computes its first eight desired GSVD components. However, the desired largest and well separated generalized singular values of \(\{A, B\}\) become the smallest clustered ones of \(\{B, A\}\). It may be this reason that made JDGSVD fail to compute the ninth desired GSVD component when total \(n\) correction equations have been solved approximately. We have found that the relative residual norm of the computed approximate GSVD component could not drop below \(10^{-9}\). In contrast, as we see from Table 7, CPF-JDGSVD succeeds to compute all the desired GSVD components accurately.

---

\(\text{Springer}\)
**Experiment 5.6** We use CPF-JDGSVD and JDGSVD to compute the nine GSVD components of the matrix pair \( \{A, B\} = \{plddb^T, B_2^T\} \) corresponding to the generalized singular values closest to \( \tau = 70 \), where \( B_2 \) is the \( n \times (n + 2) \) scaled discrete approximation of the second order derivative operator of dimension one:

\[
B_2 = \begin{bmatrix}
-1 & 2 & -1 \\
\vdots & \ddots & \vdots \\
-1 & 2 & -1 \\
\end{bmatrix} \in \mathbb{R}^{n \times (n+2)}.
\]

The desired generalized singular values are the interior ones of \( \{A, B\} \) and highly clustered.

For this problem, \( \kappa(A) = 1.23 \times 10^4 \), \( \kappa(B) = 1.69 \times 10^6 \), and \( \kappa([A]_{\mathbb{R}^2}) = 1.40 \times 10^2 \). Both \( A \) and \( B \) are not well conditioned, and it is thus expected that JDGSVD cannot compute generalized singular vectors accurately and may not be numerically backward stable but CPF-JDGSVD works well. We have found that CPF-JDGSVD used 105 outer iterations and 119874 inner iterations to achieve the convergence, about twice of 53 outer and 67330 inner iterations used by JDGSVD. All the generalized singular values computed by JDGSVD and CPF-JDGSVD are very accurate with the relative errors lying in \([10^{-15}, 10^{-13}]\). Unfortunately, as can be seen from Fig. 2b, the generalized singular vectors computed by CPF-JDGSVD are very accurate, and are one and half to four orders more accurate than those computed by JDGSVD. We have also applied CPF-JDGSVD and JDGSVD to \( \{B, A\} \) with \( \tau = \frac{1}{70} \) and observed similar accuracy advantage of CPF-JDGSVD over JDGSVD.

Finally, some words are on the inner iterations. The correction equation (4.12) or (4.13) is symmetric indefinite, and may be ill conditioned. When a Krylov iterative solver such as MINRES is used to solve them, preconditioning is naturally appealing. Unfortunately, it is generally hard to effectively precondition such correction equations. We have used the MATLAB built-in function ilu with setup.droptol = 0.1 and 0.01 to compute sparse ILU factorizations of \( A^T A - \theta^2 B^T B \) and \( A^T A - \tau^2 B^T B \) as preconditioners, and solved the resulting preconditioned nonsymmetric correction equations using the BiCGStab algorithm [24]. We have found that such preconditioners are quite ineffective and, for many of the test problems, the preconditioned BiCGStab uses more inner iterations than the unpreconditioned MINRES. Therefore, we do not present the results on the ILU preconditioned BiCGStab. Similar is for the correction equations in JDGSVD [10].

**6 Conclusions**

We have proposed a CPF-JDGSVD method for computing a partial GSVD of the large regular matrix pair \( \{A, B\} \). At extraction phase, the method is a standard Rayleigh–Ritz projection that implicitly solves the mathematically equivalent generalized eigenvalue problem of \( \{A^T A, B^T B\} \) without explicitly forming the cross-product matrices \( A^T A \) and \( B^T B \), and thus avoids the possible accuracy loss of the computed GSVD components. In the inner iterations, the algorithm iteratively solves the correction equations approximately. We have established a convergence result on the approximate generalized singular values, derived the compact bounds for the condition numbers of the correction equations, and established an important relationship between the accuracy of expansion vectors and the solution accuracy of inner iterations. Based on these results and a detailed analysis on them, we have proposed reliable stopping criteria for the inner iterations. For a practical purpose, we have focused...
on several issues and have developed a thick-restart CPF-JDGSVD algorithm with deflation and purgation for computing more than one GSVD components of \( \{A, B\} \) corresponding to the generalized singular values closest to \( \tau \).

Numerical experiments have confirmed the efficiency, reliability and accuracy of our CPF-JDGSVD algorithm for computing both interior and extreme GSVD components of various matrix pairs. In the meantime, we have numerically compared CPF-JDGSVD with JDGSVD, and justified that the former is a more general-purpose algorithm.

We point out that CPF-JDGSVD and its thick-restart with deflation and purgation are directly applicable to the GSVD computation of a complex matrix pair \( (A, B) \) by simply replacing the transpose with the conjugate transpose. All the results and analysis trivially hold in the complex case.

Nevertheless, CPF-JDGSVD may face two problems when computing the GSVD components corresponding to some truly interior or clustered generalized singular values of a large matrix pair \( \{A, B\} \). As we have seen, CPF-JDGSVD is mathematically the standard Rayleigh–Ritz method applied to the generalized eigenvalue problem of \( \{A^T A, B^T B\} \), and it uses the standard extraction approach to obtain approximate GSVD components. It is well known from, e.g., [26, 28] that the standard extraction approach suits better for the computation of exterior eigenvalues but may not work very well and convergence behavior may be irregular for the computation of interior ones. Adapted these facts to the GSVD context, CPF-JDGSVD may encounter serious difficulties to select the correct Ritz values even though there is a good one among them, and the approximate generalized singular vectors may converge irregularly and may even fail to converge even thought the approximate generalized singular value converges. We have indeed observed irregular convergence behavior of the outer iterations of CPF-JDGSVD from some of the experiments when computing truly interior generalized singular values. To overcome this potential drawback, harmonic and refined harmonic extraction based JDGSVD type methods are appealing, which are expected to suit better for the computation of interior GSVD components of a large matrix pair, just as done by the harmonic and refined harmonic JD type methods proposed in [12] for computing several interior SVD components of a single matrix. In [14], we have proposed two harmonic extraction based JDGSVD algorithms, which have been numerically confirmed to be superior to the CPF-JDGSVD algorithm for the truly interior GSVD components. Though they are generally better than CPF-JDGSVD for the interior GSVD, we have observed irregular and slow convergence. Proper refined and refined harmonic extractions are expected to have smoother convergence behavior and higher overall efficiency. This will constitute our forthcoming work.

The second problem is the solutions of correction equations. For \( \tau \) inside the generalized singular values, the correction equations (4.9) and (4.10) are highly indefinite; if a desired generalized singular value is clustered, the correction equation (4.9) is ill-conditioned. In either case, MINRES generally converges very slowly and may consume many inner iterations even for a low or modest accuracy. As a result, the overall efficiency of Algorithm 1 is low. To accelerate the convergence of inner iterations, preconditioning is definitely appealing. Unfortunately, for highly indefinite linear systems, those commonly used ILU preconditioners are generally ineffective, so are factorized approximate sparse inverse preconditioners [31]. We have confirmed that ILU preconditioners work poorly for our correction equations. However, notice that the coefficient matrices of correction equations have special forms and structures, it is possible to seek for some specific effective preconditioning techniques. As a first step towards preconditioning, it may be necessary and helpful to investigate the eigenvalue distributions of the coefficient matrices in correction equations, consider their condition numbers, and find out the convergence rates of MINRES applied to them. Such kind of results
may provide theoretical background for proposing and designing specific preconditioners for the correction equations involved.

Acknowledgements
We would like to thank the two referees for their suggestions and comments, which made us improve the presentation of the paper.

Funding
The work of the first and second authors was supported by the Youth Program of the Natural Science Foundation of Jiangsu Province (No. BK20220482) and the National Science Foundation of China (No. 12171273), respectively.

Data Availability
Enquires about data availability should be directed to the authors.

Declarations

Conflict of interest
The two authors declare that they have no financial interests, and they read and approved the final manuscript. The algorithmic Matlab code is available upon reasonable request from the corresponding author.

References

1. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., Van der Vorst, H.A.: Templates for the solution of algebraic eigenvalue problems: a practical guide. SIAM, Philadelphia, PA (2000)
2. Betcke, T.: The generalized singular value decomposition and the method of particular solutions. SIAM J. Sci. Comput. 30(3), 1278–1295 (2008)
3. Björck, Å.: Numerical methods for least squares problems. SIAM, Philadelphia, PA (1996)
4. Chu, K.W.E.: Singular value and generalized singular value decompositions and the solution of linear matrix equations. Linear Algebra Appl. 88, 83–98 (1987)
5. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Software 38, 1–25 (2011). Data available online at http://www.cise.ufl.edu/research/sparse/matrices/
6. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. John Hopkins University Press, Baltimore (2012)
7. Greenbaum, A.: Iterative methods for solving linear systems. SIAM, Philadelphia, PA (1997)
8. Hansen, P.C.: Regularization, GSVD and truncated GSVD. BIT 29(3), 491–504 (1989)
9. Hansen, P.C.: Rank-deficient and discrete Ill-posed problems: numerical aspects of linear inversion. SIAM, Philadelphia, PA (1998)
10. Hochstenbach, M.E.: A Jacobi-Davidson type method for the generalized singular value problem. Linear Algebra Appl. 431(3–4), 471–487 (2009)
11. Howland, P., Jeon, M., Park, H.: Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. SIAM J. Matrix Anal. Appl. 25(1), 165–179 (2003)
12. Huang, J., Jia, Z.: On inner iterations of Jacobi-Davidson type methods for large SVD computations. SIAM J. Sci. Comput. 41(3), A1574–A1603 (2019)
13. Huang, J., Jia, Z.: On choices of formulations of computing the generalized singular value decomposition of a matrix pair. Numer. Algor. 87, 689–718 (2021)
14. Huang, J., Jia, Z.: Two harmonic Jacobi-Davidson methods for computing a partial generalized singular value decomposition of a large matrix pair. J. Sci. Comput. 93(2), 41 (2022). https://doi.org/10.1007/s10915-022-01993-07
15. Jia, Z., Li, C.: Inner iterations in the shift-invert residual Arnoldi method and the Jacobi-Davidson method. Sci. China Math. 57, 1733–1752 (2014)
16. Jia, Z., Li, C.: Harmonic and refined harmonic shift-invert residual Arnoldi and Jacobi-Davidson methods for interior eigenvalue problems. J. Comput. Appl. Math. 282, 83–97 (2015)
17. Jia, Z., Li, H.: The joint bidiagonalization process with partial reorthogonalization. Numer. Algor. 88, 965–992 (2021)
18. Jia, Z., Stewart, G.: An analysis of the Rayleigh-Ritz method for approximating eigenspaces. Math. Comput. 70(234), 637–647 (2001)
19. Jia, Z., Yang, Y.: A joint bidiagonalization based iterative algorithm for large scale general-form tikhonov regularization. Appl. Numer. Math. 157, 159–177 (2020)
20. Kågström, B.: The generalized singular value decomposition and the general (A−λB)-problem. BIT 24(4), 568–583 (1984)
21. Paige, C.C., Saunders, M.A.: Towards a generalized singular value decomposition. SIAM J. Numer. Anal. 18(3), 398–405 (1981)
22. Park, C.H., Park, H.: A relationship between linear discriminant analysis and the generalized minimum squared error solution. SIAM J. Matrix Anal. Appl. 27(2), 474–492 (2005)
23. Refahi Sheikhani, A.H., Kordrostami, S.: New iterative methods for generalized singular value problems. Math. Sci. 11, 247–265 (2017)
24. Saad, Y.: Iterative methods for sparse linear systems, 2nd edn. SIAM, Philadelphia, PA (2003)
25. Stathopoulos, A., Saad, Y., Wu, K.: Dynamic thick restarting of the Davidson and the implicitly restarted Arnoldi methods. SIAM J. Sci. Comput. 19, 227–245 (1998)
26. Stewart, G.W.: Matrix algorithms eigen systems, vol. II. SIAM, Philadelphia, PA (2001)
27. Stewart, G.W., Sun, J.G.: Matrix perturbation theory. Academic Press Inc, Boston (1990)
28. van der Vorst, H.A.: Computational Methods for Large Eigenvalue Problems. Handbook of Numerical Analysis, Vol. VIII, Ciarlet, P.G., Lions, J.L. (eds), Elsevier (2002)
29. Van Huffel, S., Lemmerling, P.: Total least squares and errors-in-variables modeling. Kluwer Academic Publishers (2002)
30. Van Loan, C.F.: Generalizing the singular value decomposition. SIAM J. Numer. Anal. 13(1), 76–83 (1976)
31. Wathen, A.J.: Preconditioning. Acta Numer. 24, 329–376 (2015)
32. Zha, H.: Computing the generalized singular values/vectors of large sparse or structured matrix pairs. Numer. Math. 72(3), 391–417 (1996)
33. Zwaan, I.N.: Cross product-free matrix pencils for computing generalized singular values (2019). arXiv:1912.08518 [math.NA]
34. Zwaan, I.N., Hochstenbach, M.E.: Generalized Davidson and multidirectional-type methods for the generalized singular value decomposition (2017). arXiv:1705.06120 [math.NA]

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.