SIAM J. MATRIX ANAL. APPL.
Vol. 45, No. 1, pp. 232–259

THE JOINT BIDIAGONALIZATION OF A MATRIX PAIR WITH INACCURATE INNER ITERATIONS*

HAIBO LI†

Abstract. The joint bidiagonalization (JBD) process iteratively reduces a matrix pair $\{A, L\}$ to two bidiagonal forms simultaneously, which can be used for computing a partial generalized singular value decomposition (GSVD) of $\{A, L\}$. The process has a nested inner-outer iteration structure, where the inner iteration usually cannot be computed exactly. In this paper, we study the inaccurately computed inner iterations of JBD by first investigating the influence of computational error of the inner iteration on the outer iteration, and then proposing a reorthogonalized JBD (rJBD) process to keep orthogonality of a part of Lanczos vectors. An error analysis of the rJBD is carried out to build up connections with Lanczos bidiagonalizations. The results are then used to investigate convergence and accuracy of the rJBD based GSVD computation. It is shown that the accuracy of computed GSVD components depends on the computing accuracy of inner iterations and the condition number of $(A^T, L^T)^T$, while the convergence rate is not affected very much. For practical JBD based GSVD computations, our results can provide a guideline for choosing a proper computing accuracy of inner iterations in order to obtain approximate GSVD components with a desired accuracy. Numerical experiments are made to confirm our theoretical results.

Key words. joint bidiagonalization, GSVD, inner iteration, stopping tolerance, Lanczos bidiagonalization, convergence and accuracy

MSC codes. 65F25, 65F20, 15A18, 65G99

DOI. 10.1137/22M1541083

1. Introduction.

For many matrix computation algorithms, a basic routine is reducing a matrix to a structured one using a series of orthogonal transformations. For example, the first step of the QR algorithm for eigenvalue decomposition is reducing a matrix to a Hessenberg form [33], while for singular value decomposition (SVD) computation is reducing a matrix to a bidiagonal form [9]. For large-scale matrices, the reduction via direct orthogonal transformations is very expensive; thus a Lanczos-type iterative process is a common choice, e.g., the symmetric Lanczos process for symmetric eigenvalue problem, the Lanczos bidiagonalization for SVD, and so on [4, 7, 20, 21, 23]. When it comes to large-scale matrix pairs, one method is to implicitly transform the matrix pair problem to a stadand single matrix problem, such as the shift-and-invert Lanczos method for generalized symmetric eigenvalue decomposition [8, 13]. Another popular choice is the Jacobi–Davidson method for generalized eigenvalue/singular value decomposition, where an inner correction equation needs to be solved [2, 15, 28]. These algorithms often have the structure of nested inner-outer iterations, and the inner iteration contains a large-scale matrix computation problem that should be computed iteratively.

In this paper we focus on another inner-outer iterative algorithm that can reduce a matrix pair to two bidiagonal forms simultaneously, which is called the joint

*Received by the editors December 14, 2022; accepted for publication (in revised form) by A. K. Saibaba October 9, 2023; published electronically January 17, 2024.
https://doi.org/10.1137/22M1541083
Funding: This work was supported in part by the National Natural Science Foundation of China under grant 3192270206.
†Computing System Optimization Lab, Huawei Technologies, 100094 Beijing, China, and Institute of Computing Technology, Chinese Academy of Sciences, 100190 Beijing, China (haibolee1729@gmail.com).

bidiagonalization process. This algorithm was first proposed by Zha [35] for computing a partial generalized singular value decomposition (GSVD) of a large-scale matrix pair $\{A, L\}$ with $A \in \mathbb{R}^{m\times n}$ and $L \in \mathbb{R}^{p\times n}$. It was later adapted in [19] to solve large-scale linear ill-posed problems with general-form regularization [11, 12]. Let $C = (A^T, L^T)^T$. Consider the following compact QR factorization:

$$C = \begin{pmatrix} A \\ L \end{pmatrix} = QR = \begin{pmatrix} Q_A \\ Q_L \end{pmatrix} R, \tag{1.1}$$

where $Q \in \mathbb{R}^{(m+p)\times n}$ is column orthonormal with $Q_A \in \mathbb{R}^{m\times n}$, $Q_L \in \mathbb{R}^{p\times n}$, and $R \in \mathbb{R}^{n\times n}$ is upper triangular. Zha's method generates two upper bidiagonal matrices by implicitly applying the upper Lanczos bidiagonalization to both Q_A and Q_L. In contrast, the method proposed in [19] implicitly uses the lower and upper Lanczos bidiagonalizations to reduce Q_A and Q_L to lower and upper bidiagonal matrices, respectively. To keep the presentation short, we focus on Zha's joint bidiagonalization from now on, since it is more convenient for GSVD computation using upper bidiagonal matrices.

By choosing the same initial vectors $v_1 = \hat{v}_1$, the two Lanczos bidiagonalizations can reduce Q_A and Q_L to the following two bidiagonal matrices:

$$B_k = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \beta_{k-1} \\ & & & \alpha_k \end{pmatrix} \in \mathbb{R}^{k\times k}, \quad \widehat{B}_k = \begin{pmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & \\ & \hat{\alpha}_2 & \ddots & \\ & & \ddots & \hat{\beta}_{k-1} \\ & & & \hat{\alpha}_k \end{pmatrix} \in \mathbb{R}^{k\times k}.$$

Meanwhile they generate two groups of orthonormal vectors $\{u_1, \ldots, u_k\}$, $\{v_1, \ldots, v_k\}$ corresponding to Q_A and two groups of orthonormal vectors $\{\hat{u}_1, \ldots, \hat{u}_k\}$, $\{\hat{v}_1, \ldots, \hat{v}_k\}$ corresponding to Q_L. It is shown in [35] that the basic relation

$$\hat{v}_i = (-1)^{i-1} v_i, \quad \hat{\alpha}_i \hat{\beta}_i = \alpha_i \beta_i$$

holds. Based on this property, the two Lanczos bidiagonalizations can be jointed without an explicit QR factorization of C. Denote by $\mathcal{P}_Q$ the projection operator onto the subspace spanned by columns of Q, which can be written in matrix form as $\mathcal{P}_Q = QQ^T$ since Q has orthonormal columns. Then we have the following joint bidiagonalization algorithm:

Zha's joint bidiagonalization of $\{A, L\}$

Choose nonzero $s \in \mathbb{R}^n$, set $\tilde{v}_1 = Cs/\|Cs\|$

for $i = 1, 2, \ldots, k$

$\alpha_i u_i = \tilde{v}_i(1 : m) - \beta_{i-1} u_{i-1}$

$\beta_i \tilde{v}_{i+1} = \mathcal{P}_Q \begin{pmatrix} u_i \\ 0_p \end{pmatrix} - \alpha_i \tilde{v}_i$

$\hat{\alpha}_i \hat{u}_i = (-1)^{i-1} \tilde{v}_i(m + 1 : m + p) - \hat{\beta}_{i-1} \hat{u}_{i-1}$

$\hat{\beta}_i = (\alpha_i \beta_i)/\hat{\alpha}_i$

end

where $\beta_0 = \hat{\beta}_0 = 0$ and $\|u_i\| = \|\hat{u}_i\| = \|\tilde{v}_i\| = 1$. In this paper, $\|\cdot\|$ always means 2-norm of a matrix or vector. Let $\tilde{u}_i = (u_i^T, 0_p^T)^T$, where $0_p \in \mathbb{R}^p$ denotes the p-dimensional

zero vector. Note that $\mathcal{P}_Q\tilde{u}_i$ can be computed iteratively by the relation $\mathcal{P}_Q\tilde{u}_i = C\tilde{z}_i$ with

$$\tilde{z}_i = \arg\min_{\tilde{z}\in\mathbb{R}^n} \|C\tilde{z} - \tilde{u}_i\|, \tag{1.2}$$

where this least squares problem can be solved by an iterative method. The Lanczos vectors v_i and $\hat{v}_i$ can be implicitly obtained from $\tilde{v}_i$.

The joint bidiagonalization (JBD) of a matrix pair is a generalization of Lanczos bidiagonalization of a single matrix. In exact arithmetic, the k-step JBD reduces A and L to small-scale upper bidiagonal matrices B_k and $\widehat{B}_k$, and the reduction processes of A and L are equivalent to the upper Lanczos bidiagonalizations of Q_A and Q_L. Therefore, B_k and $\widehat{B}_k$ are the Ritz–Galerkin projections of Q_A and Q_L on proper Krylov subspaces. This makes the JBD process useful for designing efficient algorithms for large sparse matrix pair problems. For example, some extreme generalized singular values and vectors of $\{A, L\}$ can be approximated by using the SVD of B_k or $\widehat{B}_k$ [17, 35]; the linear ill-posed problems with general-form Tikhonov regularization $\min_x \left\{\|Ax-b\|^2 + \lambda^2\|Lx\|^2\right\}$ can be solved iteratively by solving small-scale problems containing B_k and $\widehat{B}_k$ at each iteration [18, 19].

For a practical implementation of JBD, there are some issues that must be addressed. For example, as was pointed out by Zha at the end of [35], the computed Lanczos vectors u_i, $\hat{u}_i$, and $\tilde{v}_i$ in finite precision arithmetic quickly lose orthogonality, which will cause a delay of convergence for approximating GSVD components and the appearance of spurious copies of approximations [17]. Therefore, a proper reorthogonalization strategy should be included in JBD to maintain some level of orthogonality to preserve regular convergence for GSVD computations. This issue has been studied by Jia and Li in [16, 17], where they investigate the semiorthogonalization strategy for JBD and propose an efficient partial reorthogonalization technique that can keep regular convergence behavior of computed quantities. Another issue is the increasing computation and storage cost due to the gradually expanding Krylov subspaces, especially when reorthogonalization is exploited, since all Lanczos vectors must be kept throughout the computation. Recently, Alvarruiz, Campos, and Roman [1] developed a thick restart technique for JBD to compute a partial GSVD, which can keep the size of the Krylov basis bounded, and thus the storage and computation cost can be further saved.

However, it should be pointed out that the above researchers do not take into consideration the inaccurate computation of $\mathcal{P}\tilde{u}_i$. The JBD process has the structure of nested inner-outer iterations, where the Lanczos bidiagonalization is the outer iteration while an iterative solver for (1.2) plays the role of inner iteration. The overall computational cost of the algorithm is proportional to the overall number of inner iterations, and thus a bottleneck is that iteratively solving a large-scale least squares problem at each outer iteration may be very costly, especially when the solution accuracy is high. Numerical experiments have shown that the inaccuracy in forming $\mathcal{P}_Q\tilde{u}_i$ does limit the final accuracy of computed GSVD components [35]; thus this issue is important for the JBD based GSVD computation. On the other hand, for the JBD based regularization algorithms for discrete ill-posed problems with general-form regularization, it is numerically shown that the inner least squares problem need not be solved with very high accuracy [18, 19]; e.g., for the LSQR solver for (1.2), the default stopping tolerance $\texttt{tol} = 10^{-6}$ for iteration is often enough to obtain a final regularized solution without loss of accuracy. This means that for ill-posed problems

the inner least squares problems may be solved with considerably relaxed accuracy, and thus the overall efficiency can be improved.

In this paper, we study the influence of inaccuracy of inner iterations on the behavior of the JBD algorithm, with an emphasis on the effect on the convergence and accuracy of computed GSVD components. The main contributions are the following:

- For a commonly used stopping criterion for iteratively solving (1.2), we investigate the influence of the computational error of the inner iteration on the outer iteration. This reveals that when the inner iteration is inaccurately computed, the outer iteration is not equivalent to the Lanczos bidiagonalization of Q_A and Q_L any longer but has a perturbation of order $\mathcal{O}(\kappa(C)\tau)$, where $\kappa(C)$ is the condition number of C and τ describes the solution accuracy of (1.2). A couple of recursive relations that describe the loss of orthogonality of the computed vectors are also established.
- We propose a reorthogonalized JBD (rJBD) process which maintains the orthogonality of $\tilde{v}_i$. We perform an error analysis on the k-step rJBD to establish connections between rJBD and the two Lanczos bidiagonalizations of Q_A, where $\kappa(C)\tau$ plays a crucial role for obtaining some useful upper bounds.
- The results of the above error analysis are used to investigate the convergence and accuracy of the computed GSVD components by rJBD. We show that the approximate generalized singular values can only reach an accuracy of order $\mathcal{O}(\kappa(C)\tau)$ by the SVD of B_k or $\widehat{B}_k$, and the accuracy of approximate right generalized singular vectors depends not only on the value of $\kappa(C)\tau$ but also on the gap between generalized singular values. In addition, it can be shown that the regular convergence rate of approximate generalized singular values and right vectors can be kept for the rJBD based GSVD computation.

Our results can theoretically explain some numerically observed phenomena of JBD in [35]. For example, we theoretically demonstrate that when $\mathcal{P}_Q\tilde{u}_i$ is computed with lower accuracy and C is more ill-conditioned, the orthogonality is lost at an earlier stage. We also give a theoretical explanation for the numerical conclusion in [35] that convergence rates for the approximate GSVD components are not affected very much, while the final accuracy does depend on $\kappa(C)$ and the computing accuracy of inner iterations. For practical JBD based GSVD computations, our results can provide a guideline for choosing the computing accuracy of inner iterations in order to obtain approximate GSVD components with a desired accuracy.

The paper is organized as follows. In section 2, we review some basic properties of the joint bidiagonalization and GSVD of a matrix pair. In section 3, we investigate the inaccurately computed inner iterations and propose a reorthogonalized JBD (rJBD) process to keep orthogonality of $\widetilde{V}_k$. An error analysis is carried out in section 4 to build up connections between the rJBD process and Lanczos bidiagonal reductions of Q_A and Q_L. The convergence and accuracy of approximate GSVD components computed by rJBD are investigated in section 5. Numerical experiment results for indicating our theory are given in section 6, and some concluding remarks follow in section 7.

Throughout the paper, we denote by 0_k the k-dimensional zero column vector, and by I_k and $0_{k\times l}$ the identity matrix of order k and zero matrix of order $k \times l$, respectively. The subscripts are omitted when there is no confusion. We use $e_i^{(k)}$ to denote the ith column of I_k and $\mathcal{R}(M)$ to denote the range space of matrix M.

2. Joint bidiagonalization and GSVD of a matrix pair. Although not computing Q_A or Q_L explicitly, the joint bidiagonalization process proposed in [35] is based on the application of upper Lanczos bidiagonalization processes to Q_A and Q_L, respectively. In exact arithmetic, the k-step joint bidiagonalization reduces A and L to two upper bidiagonal matrices B_k and $\widehat{B}_k$, and it generates three groups of column orthonormal matrices $U_k = (u_1, \ldots, u_k)$, $\widehat{U}_k = (\hat{u}_1, \ldots, \hat{u}_k)$, and $\widetilde{V}_{k+1} = (\tilde{v}_1, \ldots, \tilde{v}_{k+1})$. Meanwhile, it follows that $\tilde{v}_i$ is in $\mathcal{R}(Q)$, and thus we can write it as $\tilde{v}_i = Qv_i$ with $v_i \in \mathbb{R}^n$. The k-step JBD process can be written in matrix form as

$$(I_m, 0_{m\times p})\widetilde{V}_k = U_k B_k, \tag{2.1}$$

$$QQ^T \begin{pmatrix} U_k \\ 0_{p\times k} \end{pmatrix} = \widetilde{V}_k B_k^T + \beta_k \tilde{v}_{k+1}(e_k^{(k)})^T, \tag{2.2}$$

$$(0_{p\times m}, I_p)\widetilde{V}_k P = \widehat{U}_k \widehat{B}_k, \tag{2.3}$$

where $P = \mathrm{diag}(1, -1, \ldots, (-1)^{k-1}) \in \mathbb{R}^{k\times k}$. Let $\hat{v}_i = (-1)^{i-1} v_i$. Using the relation

$$B_k^T B_k + \bar{B}_k^T \bar{B}_k = I_k \tag{2.4}$$

proved in [35], where $\bar{B}_k = \widehat{B}_k P$, one can deduce from (2.1)–(2.3) the following matrix-form relations:

$$Q_A V_k = U_k B_k, \quad Q_A^T U_k = V_k B_k^T + \beta_k v_{k+1}(e_k^{(k)})^T, \tag{2.5}$$

$$Q_L \widehat{V}_k = \widehat{U}_k \widehat{B}_k, \quad Q_L^T \widehat{U}_k = \widehat{V}_k \widehat{B}_k^T + \hat{\beta}_k \hat{v}_{k+1}(e_k^{(k)})^T, \tag{2.6}$$

where $V_k = (v_1, \ldots, v_k)$ and $\widehat{V}_k = (\hat{v}_1, \ldots, \hat{v}_k)$ are column orthonormal. Therefore, the process of computing U_k, V_{k+1}, and B_k is actually the upper Lanczos bidiagonalization of Q_A, while the process of computing $\widehat{U}_k$, $\widehat{V}_{k+1}$, and $\widehat{B}_k$ is the upper Lanczos bidiagonalization of Q_L.

The generalized singular value decomposition (GSVD) of a matrix pair was introduced by Van Loan [32], with subsequent additional developments by Paige and Saunders [26]. The following description of GSVD is based on the CS decomposition [10, section 2.5.4], where the compact QR factorization of C is defined as (1.1).

THEOREM 2.1 (CS decomposition). *Suppose* $\mathrm{Rank}(C) = r$. *For the column orthonormal matrix Q, the CS decomposition of $\{Q_A, Q_L\}$ is*

$$\begin{pmatrix} Q_A \\ Q_L \end{pmatrix} = \begin{pmatrix} P_A & \\ & P_L \end{pmatrix} \begin{pmatrix} C_A \\ S_L \end{pmatrix} W^T,$$

where

$$C_A = \begin{pmatrix} \Sigma_A, & 0 \\ r & n-r \end{pmatrix} \begin{matrix} m \end{matrix}, \quad S_L = \begin{pmatrix} \Sigma_L, & 0 \\ r & n-r \end{pmatrix} p$$

with

$$\Sigma_A = \begin{pmatrix} I_q & & \\ & C_l & \\ & & O \\ q & l & r-q-l \end{pmatrix} \begin{matrix} q \\ l \\ m-q-l \end{matrix}, \quad \Sigma_B = \begin{pmatrix} O & & \\ & S_l & \\ & & I_t \\ q & l & r-q-l \end{pmatrix} \begin{matrix} p-r+q \\ l \\ r-q-l \end{matrix}$$

satisfying $C_A^T C_A + S_L^T S_L = I_n$, *and* $P_A \in \mathbb{R}^{m\times m}$, $P_L \in \mathbb{R}^{p\times p}$, *and* $W \in \mathbb{R}^{n\times n}$ *are orthogonal matrices.*

If we write $C_l = \mathrm{diag}(c_{q+1}, \ldots, c_{q+l})$ with $c_{q+1} \geq \cdots \geq c_{q+l} > 0$ and $S_l = \mathrm{diag}(s_{q+1}, \ldots, s_{q+l})$ with $0 < s_{q+1} \leq \cdots \leq s_{q+l}$, then $c_i^2 + s_i^2 = 1$, $i = q+1, \ldots, q+l$, and the generalized singular values of $\{A, L\}$ are

$$\underbrace{\infty, \ldots, \infty}_{q}, \quad \underbrace{c_{q+1}/s_{q+1}, \ldots, c_{q+l}/s_{q+l}}_{l}, \quad \underbrace{0, \ldots, 0}_{t},$$

where $t = r - q - l$. To ease the presentation, we always assume that $\{A, L\}$ is regular, i.e., $\mathrm{Rank}(\mathrm{C}) = n$, which is called a Grassmann matrix pair [22, 30]. Discussions about the GSVD of a nonregular matrix pair can be found in [24, 26, 31]. For the regular $\{A, L\}$, it follows that R is nonsingular and the GSVD is

$$A = P_A C_A X^{-1}, \quad L = P_L S_L X^{-1} \tag{2.7}$$

with $X = R^{-1}W \in \mathbb{R}^{n\times n}$. The columns of P_A, P_L, and X are called generalized singular vectors.

By (2.1) and (2.3), the k-step JBD process satisfies

$$AZ_k = U_k B_k, \quad LZ_k = \widehat{U}_k \bar{B}_k, \tag{2.8}$$

where $Z_k = R^{-1}V_k = (z_1, \ldots, z_k)$. Therefore, B_k is the Ritz–Galerkin projection of A on Krylov subspaces $\mathrm{span}(U_k)$ and $\mathrm{span}(Z_k)$, while $\bar{B}_k$ is the Ritz–Galerkin projection of L on Krylov subspaces $\mathrm{span}(\widehat{U}_k)$ and $\mathrm{span}(Z_k)$. This makes it convenient to approximate some extreme (largest or smallest) generalized singular values and corresponding vectors of $\{A, L\}$ by the SVD of B_k and $\bar{B}_k$, where the singular values of B_k and $\bar{B}_k$ can be used to approximate c_i and s_i, respectively. A more detailed investigation on the JBD method for GSVD computation is found in section 5.

3. Inaccurate inner iterations and the reorthogonalized JBD process. The JBD process has the structure of nested inner-outer iterations. The overall computational cost of the algorithm is proportional to the overall number of inner iterations. In some cases, using a sparse QR factorization for the inner least squares problem is a good choice. For most large-scale sparse problems, however, the LSQR solver is often much faster, and thus should be exploited for computing inner iterations of the JBD process.

Suppose that (1.2) is solved iteratively with the following stopping criterion:

$$\frac{\|C^T \bar{r}_i\|}{\|C\|\|\bar{r}_i\|} \leq \tau, \tag{3.1}$$

where $\bar{r}_i = \tilde{u}_i - C\bar{z}_i$ is the residual with $\bar{z}_i$ the approximate solution to (1.2). In (3.1), τ is called the stopping tolerance, which describes the accuracy of the computed approximation. This stopping criterion is commonly used in iterative methods for solving least squares problems such as the LSQR; see, e.g., [27]. In our analysis, rounding errors in finite precision arithmetic are not taken into account. Denote by $\kappa(M) = \|M\|\|M^\dagger\|$ the condition number of a matrix M. The following result describes the influence of the computational error of the inner iteration on the outer iteration.

THEOREM 3.1. *For the k-step JBD, suppose the inner least squares problem* (1.2) *is solved iteratively with stopping criterion* (3.1). *Then there exist vectors* $\tilde{g}_i \in \mathcal{R}(Q)$ *such that*

$$\beta_i \tilde{v}_{i+1} = QQ^T \begin{pmatrix} u_i \\ 0_p \end{pmatrix} - \alpha_i \tilde{v}_i - \tilde{g}_i \tag{3.2}$$

for $i = 1, \ldots, k$. *If* $\kappa(C)\tau < 1$, *then* $\tilde{g}_i$ *satisfies*

$$\|\tilde{g}_i\| \le 3\kappa(C)\tau + \mathcal{O}(\kappa(C)^2\tau^2). \tag{3.3}$$

Proof. Suppose that the exact solution of (1.2) is $\tilde{z}_i$ with residual $\tilde{r}_i = \tilde{u}_i - C\tilde{z}_i$. Then $QQ^T\tilde{u}_i = C\tilde{z}_i$. Since the computed approximation to $QQ^T\tilde{u}_i$ is $C\bar{z}_i$, at the ith iteration we have

$$\begin{aligned} \beta_i \tilde{v}_{i+1} &= C\bar{z}_i - \alpha_i \tilde{v}_i \\ &= C\tilde{z}_i - \alpha_i \tilde{v}_i - (C\tilde{z}_i - C\bar{z}_i) \\ &= QQ^T\tilde{u}_i - \alpha_i \tilde{v}_i - \tilde{g}_i, \end{aligned}$$

where $\tilde{g}_i := C\tilde{z}_i - C\bar{z}_i \in \mathcal{R}(Q)$.

Now we give the upper bound on $\tilde{g}_i$. It is known from [27] that $\bar{z}_i$ is the exact solution to the perturbed problem

$$\min_{\tilde{z}} \|\tilde{u}_i - (C + \widetilde{E}_i)\tilde{z}\| \tag{3.4}$$

with

$$\widetilde{E}_i = -\frac{\bar{r}_i^T \bar{r}_i C}{\|\bar{r}_i\|^2}, \quad \frac{\|\widetilde{E}_i\|}{\|C\|} = \frac{\|C^T\bar{r}_i\|}{\|C\|\|\bar{r}_i\|} \le \tau.$$

Suppose the residual of (3.4) is $\bar{r}_i = \tilde{u}_i - (C + \widetilde{E}_i)\bar{z}_i$. By the perturbation theory of least squares problems [14, Theorem 20.1], we have

$$\frac{\|\tilde{z}_i - \bar{z}_i\|}{\|\tilde{z}_i\|} \le \frac{\kappa(C)\tau}{1 - \kappa(C)\tau}\left(1 + \frac{\kappa(C)\|\tilde{r}_i\|}{\|C\|\|\tilde{z}_i\|}\right), \tag{3.5}$$

$$\|\tilde{r}_i - \bar{r}_i\| \le 2\kappa(C)\|\tilde{u}_i\|\tau = 2\kappa(C)\tau. \tag{3.6}$$

We mention that the right-hand terms of (3.5) and (3.6) are slightly different from those in [14, Theorem 20.1], since $\tilde{u}_i$ in (3.4) is not perturbed.[1] By the expressions of $\tilde{r}_i$ and $\bar{r}_i$, we have

$$\begin{aligned} \|C\tilde{z}_i - C\bar{z}_i\| &= \|\widetilde{E}_i\bar{z}_i - (\tilde{r}_i - \bar{r}_i)\| \le \|\widetilde{E}_i\|\|\bar{z}_i\| + \|\tilde{r}_i - \bar{r}_i\| \\ &\le (\|C\|\|\bar{z}_i\| + 2\kappa(C))\tau. \end{aligned}$$

Note that $\|\tilde{z}_i\| = \|C^\dagger\tilde{u}_i\| \le \|C^\dagger\|$ and $\|\tilde{r}_i\| \le \|\tilde{u}_i\| = 1$. By (3.5) we have

$$\begin{aligned} \|C\|\|\bar{z}_i\|\tau &\le \|C\|\tau(\|\tilde{z}_i\| + \|\tilde{z}_i - \bar{z}_i\|) \\ &\le \|C\|\|\tilde{z}_i\|\tau + \frac{\|C\|\|\tilde{z}_i\|\tau\kappa(C)\tau}{1 - \kappa(C)\tau} + \frac{\kappa(C)^2\tau^2\|\tilde{r}_i\|}{1 - \kappa(C)\tau} \\ &\le \kappa(C)\tau + \frac{\kappa(C)^2\tau^2}{1 - \kappa(C)\tau} + \frac{\kappa(C)^2\tau^2}{1 - \kappa(C)\tau} \\ &= \kappa(C)\tau + \mathcal{O}(\kappa(C)^2\tau^2), \end{aligned}$$

[1] One can check the proof of Theorem 20.1 in [14, chapter 20.10] to verify the correctness of (3.5) and (3.6).

which leads to

$$\|C\tilde{z}_i - C\bar{z}_i\| \le 3\kappa(C)\tau + \mathcal{O}(\kappa(C)^2\tau^2).$$

Thus the upper bound on $\|\tilde{g}_i\|$ is obtained. ∎

Since $\tilde{v}_1$ and $\tilde{g}_i$ are in $\mathcal{R}(Q)$, by (3.2) we get $\tilde{v}_i \in \mathcal{R}(Q)$. However, due to the appearance of nonzero $\tilde{g}_i$, the computed matrices $\widetilde{V}_{k+1}$, U_k, and $\widehat{U}_k$ do not have orthonormal columns any longer. For example, by letting $\tilde{v}_i = Qv_i$ we have from (3.2)

$$\begin{aligned}\beta_1\tilde{v}_1^T\tilde{v}_2 &= \tilde{v}_1^T(QQ^T\tilde{u}_1 - \alpha_1\tilde{v}_1 - \tilde{g}_1)\\ &= v_1^TQ_A^Tu_1 - \alpha_1 - \tilde{v}_1^T\tilde{g}_1\\ &= [(I_m, 0_{m\times p})\tilde{v}_1]^Tu_1 - \alpha_1 - \tilde{v}_1^T\tilde{g}_1\\ &= -\tilde{v}_1^T\tilde{g}_1,\end{aligned}$$

where we use $\alpha_1u_1 = (I_m, 0_{m\times p})\tilde{v}_1$. Therefore, $\tilde{v}_1$ and $\tilde{v}_2$ are not orthogonal to each other. The following theorem describes the loss of orthogonality of u_i and $\tilde{v}_i$.

THEOREM 3.2. *Define $\mu_{ji} := u_j^Tu_i$ and $\nu_{ji} := \tilde{v}_j^T\tilde{v}_i$. Let $\beta_0\mu_{0i} = 0$. Then μ_{ji} and ν_{ji} satisfy the following coupled recursive relations:*

$$\alpha_i\mu_{ji} = \beta_j\nu_{j+1,i} + \alpha_j\nu_{ji} - \beta_{i-1}\mu_{j,i-1} + \tilde{v}_i^T\tilde{g}_j, \quad 1 \le j \le i-1, \tag{3.7}$$

$$\beta_i\nu_{j,i+1} = \alpha_i\mu_{ji} + \beta_{j-1}\mu_{j-1,i} - \alpha_i\nu_{ji} - \tilde{v}_j^T\tilde{g}_i, \quad 1 \le j \le i. \tag{3.8}$$

Proof. Using relation (3.2) and $\alpha_iu_i = \tilde{v}_i(1:m) - \beta_{i-1}u_{i-1}$, we have

$$\alpha_iu_i = Q_Av_i - \beta_{i-1}u_{i-1}, \tag{3.9}$$

$$\beta_iv_{i+1} = Q_A^Tu_i - \alpha_iv_i - Q^T\tilde{g}_i, \tag{3.10}$$

where we have used $\tilde{v}_i = Qv_i$. Premultiplying (3.9) by u_j^T, we have

$$\begin{aligned}\alpha_i\mu_{ji} &= u_j^TQ_Av_i - \beta_{i-1}\mu_{j,i-1}\\ &= v_i^T(\alpha_jv_j + \beta_jv_{j+1} + Q^T\tilde{g}_j) - \beta_{i-1}\mu_{j,i-1}\\ &= \alpha_j\nu_{ji} + \beta_j\nu_{j+1,i} - \beta_{i-1}\mu_{j,i-1} + \tilde{v}_i^T\tilde{g}_j,\end{aligned}$$

where we have used $v_j^Tv_i = \tilde{v}_j^T\tilde{v}_i$. The relation (3.8) can be proved similarly. ∎

This result is a corresponding version of [21, Theorem 6] that describes the loss of orthogonality of computed Lanczos vectors when rounding errors are considered. Here, the loss of orthogonality occurs because a perturbation term $Q^T\tilde{g}_i$ is added to the exact recursive relations of the Lanczos bidiagonalization at each iteration, which is caused by the inaccurate inner iteration; this can be observed from the coupled recursive relations (3.9) and (3.10). A similar recursive relation about the loss of orthogonality of $\hat{u}_i$ could also be established, but the expression is more complicated; we omit it since it is not the main aim of this paper. Theorem 3.2 indicates that as the inner iteration becomes more inaccurate and C becomes more ill-conditioned, the loss of orthogonality of u_i and $\tilde{v}_i$ occurs more rapidly. This phenomenon was numerically observed and pointed out when the JBD was first proposed in [35].

For Lanczos-type methods, the loss of orthogonality leads to a delay of convergence of Ritz values, making the JBD method for GSVD have an irregular convergence

Algorithm 3.1 The reorthogonalized JBD (rJBD) process.

Input: $A \in \mathbb{R}^{m\times n}$, $L \in \mathbb{R}^{p\times n}$, nonzero $s \in \mathbb{R}^n$

1: Let $\tilde{v}_1 = Cs/\|Cs\|$ ▷ $C = (A^T, L^T)^T$
2: **for** $i = 1, 2, \ldots, k$ **do**
3: $\alpha_i u_i = \tilde{v}_i(1:m) - \beta_{i-1}u_{i-1}$
4: Solve $\min_{\tilde{z}\in\mathbb{R}^n} \|C\tilde{z} - \tilde{u}_i\|$ by LSQR with the stopping criterion (3.1), where the approximated solution is denoted by $\bar{z}_i$ ▷ $\tilde{u}_i = (u_i^T, 0_p^T)^T$
5: $s_i = C\bar{z}_i - \alpha_i\tilde{v}_i$
6: $\beta_i\tilde{v}_{i+1} = s_i - \sum_{j=1}^{i}(s_i^T\tilde{v}_j)\tilde{v}_j$ ▷ Reorthogonalize $\tilde{v}_{i+1}$
7: $\hat{\alpha}_i\hat{u}_i = (-1)^{i-1}\tilde{v}_i(m+1:m+p) - \hat{\beta}_{i-1}\hat{u}_{i-1}$
8: $\hat{\beta}_i = (\alpha_i\beta_i)/\hat{\alpha}_i$
9: **end for**

Output: $\{u_i, \hat{u}_i\}_{i=1}^k$, $\{\tilde{v}_i\}_{i=1}^{k+1}$, $\{\alpha_i, \beta_i, \hat{\alpha}_i, \hat{\beta}_i\}_{i=1}^k$

behavior. To avoid this problem, we propose the following rJBD process. At each step, we use the Gram–Schmidt orthogonalization to reorthogonalize $\tilde{v}_i$ such that $\widetilde{V}_{k+1}$ is column orthonormal, while vectors u_i and $\hat{u}_i$ do not need to be reorthogonalized. This can save storage and computation costs compared to full reorthogonalization of all $\tilde{v}_i$, u_i, and $\hat{u}_i$. This modified algorithm is described in Algorithm 3.1.

We mention that for the rJBD process, the computed quantities u_i, $\tilde{v}_i$, $\hat{u}_i$, α_i, $\hat{\alpha}_i$, etc. are different from those obtained by the JBD process. Here we use the same notations to avoid introducing too many tedious notations, and from now on, these notations always denote quantities computed by rJBD. For the reorthogonalization of $\tilde{v}_{i+1}$, by steps 5 and 6, we can write it in a general form:

$$\beta_i\tilde{v}_{i+1} = C\bar{z}_i - \alpha_i\tilde{v}_i - \sum_{j=1}^{i}\xi_{ji}\tilde{v}_j,$$

where $\xi_{ji} = s_i^T\tilde{v}_j$ for the classical Gram–Schmidt reorthogonalization as is presented in step 6. In practical computations, using the modified Gram–Schmidt reorthogonalization is usually a better choice. By the above relation, we have

$$\begin{aligned}\beta_i\tilde{v}_{i+1} &= C\tilde{z}_i - \alpha_i\tilde{v}_i - \sum_{j=1}^{i}\xi_{ji}\tilde{v}_j - (C\tilde{z}_i - C\bar{z}_i)\\ &= QQ^T\tilde{u}_i - \alpha_i\tilde{v}_i - \sum_{j=1}^{i}\xi_{ji}\tilde{v}_j - (C\tilde{z}_i - C\bar{z}_i).\end{aligned}$$

Similarly to Theorem 3.1 and its proof, if we define $\tilde{g}_i := C\tilde{z}_i - C\bar{z}_i$, then we have

$$\beta_i\tilde{v}_{i+1} = QQ^T\begin{pmatrix}u_i\\0_p\end{pmatrix} - \alpha_i\tilde{v}_i - \sum_{j=1}^{i}\xi_{ji}\tilde{v}_j - \tilde{g}_i. \tag{3.11}$$

In particular, the property $\tilde{g}_i \in \mathcal{R}(Q)$ and the upper bound in (3.3) still hold. Note that (3.11) is applied to rJBD, and $\tilde{g}_i$ is different from that in (3.2) for JBD.

Let $\widetilde{G}_k = (\tilde{g}_1, \ldots, \tilde{g}_k)$. The k-step rJBD can be written in matrix form:

$$(I_m, 0_{m\times p})\widetilde{V}_k = U_k B_k, \tag{3.12}$$

$$QQ^T\begin{pmatrix} U_k \\ 0_{p\times k}\end{pmatrix} = \widetilde{V}_k(B_k^T + D_k) + \beta_k \tilde{v}_{k+1}\left(e_k^{(k)}\right)^T + \widetilde{G}_k, \tag{3.13}$$

$$(0_{p\times m}, I_p)\widetilde{V}_k P = \widehat{U}_k \widehat{B}_k, \tag{3.14}$$

where

$$D_k = \begin{pmatrix} \xi_{11} & \cdots & \cdots & \xi_{1k} \\ & \xi_{22} & \cdots & \xi_{2k} \\ & & \ddots & \vdots \\ & & & \xi_{kk}\end{pmatrix} \in \mathbb{R}^{k\times k}.$$

For the rJBD, the matrix $\widetilde{V}_{k+1}$ is column orthonormal, while U_k and $\widehat{U}_k$ are not column orthonormal.

4. Error analysis of the rJBD process. For the k-step rJBD, if one of α_i, β_i, $\hat{\alpha}_i$, and $\hat{\beta}_i$ becomes zero, then the procedure terminates. It is usually called a "lucky terminate" [9], since the procedure has found an invariant singular subspace. In our analysis, we assume that α_i, β_i, $\hat{\alpha}_i$, and $\hat{\beta}_i$ never become zero or numerical negligible after k steps.

Since $\tilde{v}_1$ and $\tilde{g}_i$ are in $\mathcal{R}(Q)$, by (3.11) we have $\tilde{v}_i \in \mathcal{R}(Q)$ for $i = 1, 2, \ldots$. Suppose $\tilde{v}_i = Qv_i$, and let $\hat{v}_i = (-1)^{i-1}v_i$. Then we obtain from (3.12)–(3.14) that

$$Q_A V_k = U_k B_k, \tag{4.1}$$

$$Q_A^T U_k = V_k(B_k^T + D_k) + \beta_k v_{k+1}\left(e_k^{(k)}\right)^T + G_k, \tag{4.2}$$

$$Q_L \widehat{V}_k = \widehat{U}_k \widehat{B}_k, \tag{4.3}$$

where $V_k = (v_1, \ldots, v_k)$, $\widehat{V}_k = (\hat{v}_1, \ldots, \hat{v}_k)$, and $G_k = Q^T\widetilde{G}_k = (g_1, \ldots, g_k)$ with $g_i = Q^T\tilde{g}_i$. By $\tilde{g}_i \in \mathcal{R}(Q)$ we have $\|g_i\| = \|\tilde{g}_i\|$. Based on these matrix-form relations, we make an error analysis of the rJBD process, which builds up connections with the bidiagonal reductions of Q_A and Q_L.

4.1. Bidiagonal reduction of Q_A. Note that (4.1) and (4.2) imply that the process of generating B_k is closely related to the Lanczos bidiagonalization of Q_A, where the differences include the reorthogonalizations of v_i and perturbation errors g_i. We develop methods inspired by [3] to establish a backward error bound about the k-step bidiagonal reduction of Q_A. For the sake of simplicity, we only discuss the case for $m \geq n$.

First we give a relation describing the generation of each column of B_k.

THEOREM 4.1. *For the k-step rJBD process, define*

$$\widehat{P}_{l+1} = P_1 \cdots P_{l+1}, \quad P_i = I_{m+n} - p_i p_i^T, \quad p_i = \begin{pmatrix} -e_i^{(n)} \\ u_i\end{pmatrix} \in \mathbb{R}^{m+n} \tag{4.4}$$

for $1 \leq l \leq k-1$. There exist $f_{l+1} \in \mathbb{R}^{m+n}$ such that

$$\widehat{P}_{l+1}\begin{pmatrix} \beta_l e_l^{(l)} \\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} = \begin{pmatrix} 0_n \\ Q_A v_{l+1}\end{pmatrix} + f_{l+1}, \quad \|f_{l+1}\| = \mathcal{O}(l\kappa(C)\tau), \tag{4.5}$$

with $s = m + n - l$.

Proof. By (4.1) and the expression of P_i, we have

$$\begin{aligned} P_{l+1}\begin{pmatrix}\beta_l e_l^{(l)}\\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} &= \begin{pmatrix}\beta_l e_l^{(l)}\\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} - p_{l+1}^T\begin{pmatrix}\beta_l e_l^{(l)}\\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} p_{l+1}\\ &= \begin{pmatrix}\beta_l e_l^{(l)}\\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} + \alpha_{l+1}\begin{pmatrix}-e_{l+1}^{(n)}\\ u_{l+1}\end{pmatrix}\\ &= \begin{pmatrix}\beta_l e_l^{(n)}\\ \alpha_{l+1}u_{l+1}\end{pmatrix} = \begin{pmatrix}\beta_l e_l^{(n)}\\ Q_A v_{l+1} - \beta_l u_l\end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} P_l\begin{pmatrix}\beta_l e_l^{(n)}\\ Q_A v_{l+1} - \beta_l u_l\end{pmatrix} &= P_l\begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - \beta_l P_l\begin{pmatrix}-e_l^{(n)}\\ u_l\end{pmatrix}\\ &= \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - (u_l^T Q_A v_{l+1})p_l + \beta_l p_l. \end{aligned}$$

By (4.2), we have

$$Q_A^T u_i = \alpha_i v_i + \beta_i v_{i+1} + \sum_{j=1}^{i}\xi_{ji}v_j + g_i \tag{4.6}$$

for $i = 1, 2, \ldots, l$. By (4.6) and using the column orthogonality of V_l, we have

$$u_l^T Q_A v_{l+1} = v_{l+1}^T(Q_A^T u_l) = v_{l+1}^T\left(\alpha_l v_l + \beta_l v_{l+1} + \sum_{j=1}^{l}\xi_{jl}v_j + g_l\right) = \beta_l + v_{l+1}^T g_l,$$

which leads to

$$P_l\begin{pmatrix}\beta_l e_l^{(n)}\\ Q_A v_{l+1} - \beta_l u_l\end{pmatrix} = \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - (v_{l+1}^T g_l)p_l.$$

Using the same method as above, we have

$$\begin{aligned} P_i\begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} &= \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - (u_i^T Q_A v_{l+1})p_i\\ &= \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - v_{l+1}^T\left(\alpha_i v_i + \beta_i v_{i+1} + \sum_{j=1}^{i}\xi_{ji}v_j + g_i\right)p_i\\ &= \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - (v_{l+1}^T g_i)p_i \end{aligned}$$

for $i = 1, 2, \ldots, l-1$. Combining the above two equalities leads to

$$\widehat{P}_{l+1}\begin{pmatrix}\beta_l e_l^{(l)}\\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} = P_1\cdots P_{l-1}\left(\begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} - (v_{l+1}^T g_l)p_l\right) = \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} + f_{l+1},$$

with $f_{l+1} = -\sum_{i=1}^{l}(P_1\cdots P_{i-1})(v_{l+1}^T g_i)p_i$, where $P_0 = I_{m+n}$. Note that $\|p_i\| = \sqrt{2}$ and P_i are Householder matrices. By using the upper bound on $\|g_i\| = \|\tilde{g}_i\|$ and neglecting high order terms of τ, we get $\|f_{l+1}\| = \mathcal{O}(l\kappa(C)\tau)$. ∎

This result will play an important role in the following analysis. Now we give a backward error bound about the k-step bidiagonal reduction of Q_A, which is the main result in this subsection.

THEOREM 4.2. *For the k-step rJBD process, there exist a column orthonormal matrix $\bar{U}_k \in \mathbb{R}^{m\times k}$ and a matrix $E_k \in \mathbb{R}^{m\times n}$ such that*

$$(Q_A + E_k)V_k = \bar{U}_k B_k, \tag{4.7}$$

$$(Q_A + E_k)^T \bar{U}_k = V_k B_k^T + \beta_k v_{k+1}(e_k^{(k)})^T, \tag{4.8}$$

and

$$\|E_k\| = \mathcal{O}(\sqrt{n}k\kappa(C)\tau). \tag{4.9}$$

Before proving Theorem 4.2, we first give some remarks. Notice that the relations (4.7) and (4.8) are matrix-form recurrences of the k-step (upper) Lanczos bidiagonalization of $\bar{Q}_A = Q_A + E_k$. Thus the subspace $\mathrm{span}(V_k)$ is the Krylov subspace $\mathcal{K}_k(\bar{Q}_A^T\bar{Q}_A, v_1)$; see, e.g., [10, chapter 10.4.1]. Therefore, the singular values of B_k will approximate the singular values of $\bar{Q}_A$ instead of those of Q_A. Clearly, the accuracy of approximations to c_i by the SVD of B_k is limited by the value of $\kappa(C)\tau$.

The proof of Theorem 4.2 depends on the following two lemmas. The first lemma give a relation similar to that in Theorem 4.1, where some quantities are constructed only for aiding subsequent proofs.

LEMMA 4.3. *For the k-step rJBD with $k < n$, there exist vectors $\check{u}_{k+1}, \ldots, \check{u}_n$ and $\check{v}_{k+2}, \ldots, \check{v}_n$ and nonnegative numbers $\check{\alpha}_{k+1}, \ldots, \check{\alpha}_n$ and $\check{\beta}_{k+1}, \ldots, \check{\beta}_{n-1}$ (define $\check{\beta}_n := 0$ and $\check{v}_{n+1} := 0$), such that*

(I) *$\check{u}_i$ and $\check{v}_i$ are of unit-norm, and $\check{V} = (V_{k+1}, \check{v}_{k+2}, \ldots, \check{v}_n)$ is orthogonal;*

(II) *if we define for $k \le l \le n-1$*

$$\widetilde{P}_{l+1} = \widehat{P}_k \check{P}_{k+1} \cdots \check{P}_{l+1}, \quad \check{P}_i = I_{m+n} - \check{p}_i \check{p}_i^T, \quad \check{p}_i = \begin{pmatrix} -e_i^{(n)} \\ \check{u}_i \end{pmatrix},$$

then there exist $f_{k+1} \in \mathbb{R}^{m+n}$ such that

$$\widetilde{P}_{l+1} \begin{pmatrix} \check{\beta}_l e_l^{(l)} \\ \check{\alpha}_{l+1} e_1^{(s)} \end{pmatrix} = \begin{pmatrix} 0_n \\ Q_A \check{v}_{l+1} \end{pmatrix} + f_{k+1}, \quad \|f_{k+1}\| = \mathcal{O}(k\kappa(C)\tau), \tag{4.10}$$

where $\check{v}_{k+1} := v_{k+1}$ and $\check{\beta}_k := \beta_k$.

Proof. First we construct vectors $\check{u}_{k+1}, \ldots, \check{u}_n$ and $\check{v}_{k+2}, \ldots, \check{v}_n$. For $i \ge k+1$, vectors $\check{u}_i$ and $\check{v}_{i+1}$ are generated as

$$\check{\alpha}_i \check{u}_i = Q_A \check{v}_i - \check{\beta}_{i-1}\check{u}_{i-1},$$

$$r_i = Q_A^T \check{u}_i - \check{\alpha}_i v_i \quad \check{\beta}_i \check{v}_{i+1} = r_i - \sum_{j=1}^{k+1} (v_j^T r_i) v_j - \sum_{j=k+2}^{i} (\check{v}_j^T r_i)\check{v}_j,$$

such that $\|\check{u}_i\| = \|\check{v}_{i+1}\| = 1$, where for $i = k+1$ we let $\check{u}_k = u_k$, $\check{v}_{k+1} = v_{k+1}$, and $\check{\beta}_k = \beta_k$. If the procedure terminates at some step, it can be continued by choosing a new starting vector. Note that $\check{v}_{i+1}$ are generated with full reorthogonalization. Thus $\check{\beta}_n \check{v}_{n+1} = 0_n$ and $\check{V}$ is orthogonal.

For $l \geq k+1$, by using similar calculations as in the proof of Theorem 4.1, we have

$$\begin{aligned}
\check{P}_l\check{P}_{l+1}\begin{pmatrix}\check{\beta}_l e_l^{(l)}\\ \check{\alpha}_{l+1}e_1^{(s)}\end{pmatrix} &= \begin{pmatrix}0_n\\ Q_A\check{v}_{l+1}\end{pmatrix},\\
\check{P}_i\begin{pmatrix}0_n\\ Q_A\check{v}_{l+1}\end{pmatrix} &= \begin{pmatrix}0_n\\ Q_A\check{v}_{l+1}\end{pmatrix}, \quad i=k+1,\dots,l-1,\\
P_i\begin{pmatrix}0_n\\ Q_A\check{v}_{l+1}\end{pmatrix} &= \begin{pmatrix}0_n\\ Q_A\check{v}_{l+1}\end{pmatrix} - (\check{v}_{l+1}^T g_i)p_i, \quad i=1,\dots,k.
\end{aligned}$$

Thus we obtain

$$\widehat{P}_k\check{P}_{k+1}\cdots\check{P}_{l+1}\begin{pmatrix}\beta_l e_l^{(l)}\\ \alpha_{l+1}e_1^{(s)}\end{pmatrix} = P_1\cdots P_k\begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} = \begin{pmatrix}0_n\\ Q_A v_{l+1}\end{pmatrix} + f_{k+1}$$

with $f_{k+1} = -\sum_{i=1}^{k}(P_1\cdots P_{i-1})(\check{v}_{l+1}^T g_i)p_i$ and $\|f_{k+1}\| = \mathcal{O}(k\kappa(C)\tau)$.

For $l = k$, it can also be verified that (4.10) holds and we omit the similar calculations. ∎

Theorem 4.1 and Lemma 4.3 lead to the following result.

LEMMA 4.4. *For the k-step rJBD process, define the upper bidiagonal matrix as*

$$\check{B} = \begin{pmatrix}
\alpha_1 & \beta_1 & & & & & & \\
 & \alpha_2 & \ddots & & & & & \\
 & & \ddots & \beta_{k-1} & & & & \\
 & & & \alpha_k & \beta_k & & & \\
 & & & & \check{\alpha}_{k+1} & \ddots & & \\
 & & & & & \ddots & \check{\beta}_{n-1} & \\
 & & & & & & \check{\alpha}_n &
\end{pmatrix} \in \mathbb{R}^{n\times n}.$$

If follows that

$$\begin{pmatrix}0_{n\times n}\\ Q_A\check{V}_n\end{pmatrix} + F_k = \widetilde{P}_n\begin{pmatrix}\check{B}\\ 0_{m\times n}\end{pmatrix} \tag{4.11}$$

with $F_k = (\underbrace{0_{m+n}, f_2,\dots,f_k}_{k}, \underbrace{f_{k+1},\dots,f_{k+1}}_{n-k}) \in \mathbb{R}^{(m+n)\times n}$, *where* $f_{n+1} := 0$ *for* $k = n$.

Proof. The proof can be completed by comparing each column of the right- and left-hand sides of (4.11).

For the first column, we have

$$\begin{aligned}
\widetilde{P}_n\begin{pmatrix}\alpha_1\\ 0_{m+n-1}\end{pmatrix} &= P_1\left[P_2\cdots P_k\check{P}_{k+1}\cdots\check{P}_n\begin{pmatrix}\alpha_1\\ 0_{m+n-1}\end{pmatrix}\right]\\
&= P_1\begin{pmatrix}\alpha_1\\ 0_{m+n-1}\end{pmatrix} = \begin{pmatrix}\alpha_1\\ 0_{m+n-1}\end{pmatrix} + \alpha_1\begin{pmatrix}-e_1^{(n)}\\ u_1\end{pmatrix}\\
&= \begin{pmatrix}0_n\\ \alpha_1 u_1\end{pmatrix} = \begin{pmatrix}0_n\\ Q_A v_1\end{pmatrix}.
\end{aligned}$$

For the $(l+1)$th column with $1 \le l \le k-1$, by Theorem 4.1 we have

$$\begin{aligned}\widetilde{P}_n \begin{pmatrix} \beta_l e_l^{(l)} \\ \alpha_{l+1} e_1^{(s)} \end{pmatrix} &= \widehat{P}_{l+1} \left[P_{l+2} \cdots \check{P}_n \begin{pmatrix} \beta_l e_l^{(l)} \\ \alpha_{l+1} e_1^{(s)} \end{pmatrix} \right] \\ &= \widehat{P}_{l+1} \begin{pmatrix} \beta_l e_l^{(l)} \\ \alpha_{l+1} e_1^{(s)} \end{pmatrix} = \begin{pmatrix} 0_n \\ Q_A v_{l+1} \end{pmatrix} + f_{l+1}.\end{aligned}$$

Similar calculations can be carried out for the $(l+1)$th column with $k \le l \le n-1$ by using Lemma 4.3. Therefore, the equality (4.11) holds. ∎

With the aid of the above two lemmas, we can now give the proof of Theorem 4.2.

Proof of Theorem 4.2. By Lemma 4.4, we have

$$\begin{pmatrix} 0_{n\times n} \\ Q_A \check{V}_n \end{pmatrix} + F_k = \widetilde{P}_n \begin{pmatrix} \check{B} \\ 0_{m\times n} \end{pmatrix} = \begin{pmatrix} \widehat{P}_{11} & \widehat{P}_{12} \\ \widehat{P}_{21} & \widehat{P}_{22} \end{pmatrix} \begin{pmatrix} \check{B} \\ 0_{m\times n} \end{pmatrix} = \begin{pmatrix} \widehat{P}_{11} \\ \widehat{P}_{21} \end{pmatrix} \check{B},$$

where $\widetilde{P}_n$ is partitioned as

$$\widetilde{P}_n = \begin{array}{c} \begin{matrix} n & m \end{matrix} \\ \begin{pmatrix} \widehat{P}_{11} & \widehat{P}_{12} \\ \widehat{P}_{21} & \widehat{P}_{22} \end{pmatrix} \end{array} \begin{array}{c} n \\ m \end{array}.$$

By [25, Theorem 4.1], there exist a column orthonormal matrix $\bar{U}_n \in \mathbb{R}^{m\times n}$ and a matrix $M \in \mathbb{R}^{m\times n}$ satisfying $0.5 \le \|M\| \le 1$ such that

$$Q_A \check{V}_n + \widehat{E}_k = \bar{U}_n \check{B}_n,$$

where $\widehat{E}_k = (M\widehat{P}_{11}^T, \quad I_m) F_k$. Therefore, we have

$$(Q_A + E_k)\check{V}_n = \bar{U}_n \check{B}_n$$

with $E_k = (M\widehat{P}_{11}^T, \quad I_m) F_k \check{V}_n^T$, which can also be written as

$$(Q_A + E_k)^T \bar{U}_n = \check{V}_n \check{B}_n^T.$$

By equating the first k columns of the above two equalities, respectively, we obtain (4.7) and (4.8). Finally, we have the upper bound

$$\|E_k\| \le \left\| \left(M\widehat{P}_{11}^T, \quad I_m \right) \right\| \|F_k\| \le \sqrt{2}\|F_k\| = \mathcal{O}(\sqrt{n} k \kappa(C)\tau),$$

where we have used $\|F_k\| \le \|F_k\|_F \le \sqrt{n} \max_{1\le l\le k} \|f_{l+1}\|$. ∎

4.2. Bidiagonal reduction of Q_L. For the rJBD process, a relation that is similar to (2.4) holds, and the process of generating $\widehat{B}_k$ is closely related to the Lanczos bidiagonalization of Q_L.

THEOREM 4.5. *For the k-step rJBD process, we have*

$$B_k^T B_k + \bar{B}_k^T \bar{B}_k = I_k + H_k, \tag{4.12}$$

where H_k is a diagonal matrix, and the ith diagonal of H_k is of order $\mathcal{O}(\theta_{i-1}\kappa(C)\tau)$ with $\theta_i = \sum_{j=0}^{i-1} (\hat{\beta}_i \cdots \hat{\beta}_{i-j})/(\hat{\alpha}_i \cdots \hat{\alpha}_{i-j})$ for $i \ge 1$ and $\theta_0 = 0$.

TABLE 1
Properties of the test matrices.

Matrix pair	m, p, n	$\kappa(C)$	Description
{illc1850, well1850}	1850, 1850, 712	38.6	least squares problem
{dw2048, rdb2048}	2048, 2048, 2048	261.0	electromagnetics problem
{swang1, $0.1L_{1d}$}	3169, 3168, 3169	45.0	semiconductor device problem
$\{A_1, L_1\}$	10000, 10000, 10000	100000.0	self-constructed

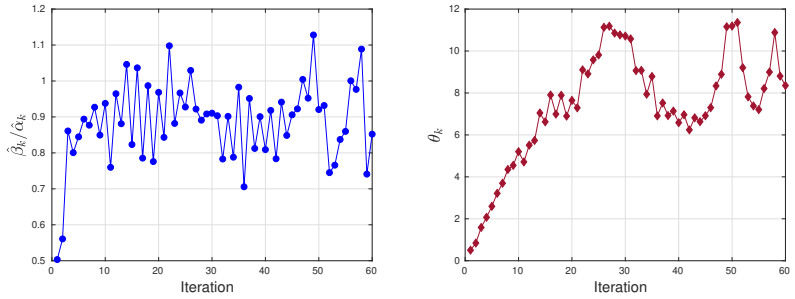


FIG. 1. *Variation of values* $\hat{\beta}_k/\hat{\alpha}_k$ *and* θ_k *for rJBD of* {illc1850, well1850}, $\tau = 10^{-12}$.

The growth speed of θ_i is moderate under the assumption that $\hat{\beta}_i/\hat{\alpha}_i$ is a moderate value for $i = 1, \ldots, k$, which is reasonable since if the rJBD process does not numerically terminate, then both $\hat{\beta}_i$ and $\hat{\alpha}_i$ are moderate values. We use the example as shown in Figure 1 to illustrate the variation of values $\hat{\beta}_k/\hat{\alpha}_k$ and θ_k with respect to k. The property of the test matrix pair {illc1850, well1850} is shown in Table 1.

Note that the only difference between relations (4.12) and (2.4) is the perturbation term H_k, which comes from the inaccurate inner iteration. From Theorem 4.5, we know that the singular values of $\bar{B}_k$ are determined by those of B_k within errors of order $\mathcal{O}(\kappa(C)\tau)$, where we omit the moderate value $\max_{1\le i\le k-1}\theta_i$. This ensures that $\bar{B}_k$ can also be used to approximate generalized singular values of $\{A, L\}$.

To prove this result, we need the following lemma.

LEMMA 4.6. *For any* $i \ge 1$, *we have*

$$\hat{\beta}_i Q_L^T \hat{u}_i = \hat{\beta}_i^2 \hat{v}_{i+1} + \widehat{V}_i d_i + q_i$$

with a vector $d_i \in \mathbb{R}^i$, *and* $\|q_i\| = \mathcal{O}(\theta_i \kappa(C)\tau)$.

Proof. The proof can be completed by mathematical induction. For $i = 1$, we obtain from $\hat{\alpha}_1 \hat{u}_1 = Q_L \hat{v}_1$ that

$$\begin{aligned}\hat{\alpha}_1 Q_L^T \hat{u}_1 &= (I_n - Q_A^T Q_A)\hat{v}_1 = \hat{v}_1 - Q_A^T(\alpha_1 u_1)\\ &= \hat{v}_1 - \alpha_1(\beta_1 v_2 + \alpha_1 v_1 + \xi_{11} v_1 + g_1)\\ &= \hat{\alpha}_1\hat{\beta}_1\hat{v}_2 + (1 - \alpha_1^2 - \alpha_1\xi_{11})\hat{v}_1 - \alpha_1 g_1.\end{aligned}$$

Then we get

$$\hat{\beta}_1 Q_L^T \hat{u}_1 = \hat{\beta}_1^2 \hat{v}_2 + \widehat{V}_1 d_1 - \hat{\beta}_1/\hat{\alpha}_1 \cdot \alpha_1 g_1 = \hat{\beta}_1^2 \hat{v}_1 + \widehat{V}_1 d_1 + q_1$$

with $d_1 = \hat{\beta}_1/\hat{\alpha}_1 \cdot (1 - \alpha_1^2 - \alpha_1\xi_{11})$ and $q_1 = -\hat{\beta}_1/\hat{\alpha}_1 \cdot \alpha_1 g_1$, and $\|q_1\| = \mathcal{O}(\theta_1\kappa(C)\tau)$ since $\alpha_1 = \|\tilde{v}_1(1:m)\| \le 1$.

Suppose the relation is true for indices up to $i-1 \geq 1$. For index i, we have

$$\begin{aligned}\hat{\alpha}_i Q_L^T \hat{u}_i &= Q_L^T Q_L \hat{v}_i - \hat{\beta}_{i-1} Q_L^T \hat{u}_{i-1} \\ &= (I_n - Q_A^T Q_A)\hat{v}_i - \hat{\beta}_{i-1} Q_L^T \hat{u}_{i-1} \\ &= \hat{v}_i + (-1)^i Q_A^T (\alpha_i u_i + \beta_{i-1} u_{i-1}) - \hat{\beta}_{i-1} Q_L^T \hat{u}_{i-1}.\end{aligned}$$

By (4.6) we have

$$(-1)^i \alpha_i Q_A^T u_i = \hat{\alpha}_i \hat{\beta}_i \hat{v}_{i+1} + (-1)^i \alpha_i \left(\alpha_i v_i + \sum_{j=1}^{i} \xi_{ji} v_j + g_i \right)$$

and

$$Q_A^T u_{i-1} \in \operatorname{span}\{\hat{v}_1, \ldots, \hat{v}_i\} + g_{i-1}.$$

Combining the above two relations with the induction hypothesis

$$\hat{\beta}_{i-1} Q_L^T \hat{u}_{i-1} = \hat{\beta}_{i-1}^2 \hat{v}_i + \widehat{V}_{i-1} d_{i-1} + q_{i-1}$$

and $\|q_{i-1}\| = \mathcal{O}(\theta_{i-1}\kappa(C)\tau)$, we get

$$\hat{\alpha}_i Q_L^T \hat{u}_i = \hat{\alpha}_i \hat{\beta}_i \hat{v}_{i+1} + \widehat{V}_i \tilde{d}_i + (-1)^i (\alpha_i g_i + \beta_{i-1} g_{i-1}) + q_{i-1} \tag{4.13}$$

with a $\tilde{d}_i \in \mathbb{R}^i$. By Theorem 4.1 we have

$$(\alpha_i^2 + \beta_{i-1}^2)^{1/2} = \left\| \begin{pmatrix} \beta_{i-1} e_i^{(i)} \\ \alpha_i e_1^{(s)} \end{pmatrix} \right\| \leq \left\| \begin{pmatrix} 0_n \\ Q_A v_i \end{pmatrix} \right\| + \|f_i\| \leq 1 + \mathcal{O}(i\kappa(C)\tau). \tag{4.14}$$

Thus we get

$$\|\alpha_i g_i + \beta_{i-1} g_{i-1}\| \leq \sqrt{2}(\alpha_i^2 + \beta_{i-1}^2)^{1/2} \max\{\|g_i\|, \|g_{i-1}\|\} = \mathcal{O}(\kappa(C)\tau)$$

by neglecting higher orders of τ. We finally obtain from (4.13)

$$\hat{\beta}_i Q_L^T \hat{u}_i = \hat{\beta}_i^2 \hat{v}_{i+1} + \widehat{V}_i d_i + q_i$$

with $q_i = \hat{\beta}_i/\hat{\alpha}_i[(-1)^i(\alpha_i g_i + \beta_{i-1} g_{i-1}) + q_{i-1}]$, and

$$\|q_i\| \leq \hat{\beta}_i/\hat{\alpha}_i \mathcal{O}(\kappa(C)\tau) + \hat{\beta}_i/\hat{\alpha}_i q_{i-1} = \mathcal{O}(\theta_i \kappa(C)\tau)$$

since $\hat{\beta}_i/\hat{\alpha}_i + \hat{\beta}_i/\hat{\alpha}_i \cdot \theta_{i-1} = \theta_i$. This completes the proof of the induction step. ∎

Now we can give the proof of Theorem 4.5.

Proof of Theorem 4.5. Using the bidiagonal structure of B_k and $\bar{B}_k$, we know that H_k is symmetric tridiagonal. Note that the subdiagonals of $B_k^T B_k$ and $\bar{B}_k^T \bar{B}_k$ are $\alpha_i \beta_i$ and $-\hat{\alpha}_i \hat{\beta}_i$, respectively. Thus, the subdiagonals of H_k are zero and H_k is a diagonal matrix. For the ith diagonal element, that is, $\alpha_i^2 + \beta_{i-1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_{i-1}^2$, we use the relations

$$\begin{aligned}\alpha_i u_i + \beta_{i-1} u_{i-1} &= \tilde{v}_i(1:m), \\ \hat{\alpha}_i \hat{u}_i + \hat{\beta}_{i-1} \hat{u}_{i-1} &= \tilde{v}_i(m+1:m+p).\end{aligned}$$

Adding the squares of norms of the above two equalities leads to

$$\alpha_1^2+\beta_{i-1}^2+\hat{\alpha}_i^2+\hat{\beta}_{i-1}^2+2(\alpha_i\beta_{i-1}u_{i-1}^Tu_i+\hat{\alpha}_i\hat{\beta}_{i-1}\hat{u}_{i-1}^T\hat{u}_i)=1.$$

For $i=1$, we have $\alpha_1^2+\hat{\alpha}_i^2=1$ due to $\beta_0=\hat{\beta}_0=0$. For $i>1$, since

$$\begin{aligned}\alpha_i u_{i-1}^Tu_i &= u_{i-1}^T(Q_Av_i-\beta_{i-1}u_{i-1})=u_{i-1}^TQ_Av_i-\beta_{i-1}\\ &= v_i^T\left(\beta_{i-1}v_i+\alpha_{i-1}v_{i-1}+\sum_{j=1}^{i-1}\xi_{ji}v_j+g_{i-1}\right)-\beta_{i-1}\\ &= v_i^Tg_{i-1},\end{aligned}$$

by (4.14) we have

$$|\alpha_i\beta_{i-1}u_{i-1}^Tu_i|\le\beta_{i-1}\|g_{i-1}\|\le[1+\mathcal{O}(i\kappa(C)\tau)]\|g_{i-1}\|=\mathcal{O}(\kappa(C)\tau).$$

By Lemma 4.6 we have

$$\begin{aligned}|\hat{\alpha}_i\hat{\beta}_{i-1}\hat{u}_{i-1}^T\hat{u}_i| &= |\hat{\beta}_{i-1}\hat{u}_{i-1}^T(Q_L\hat{v}_i-\hat{\beta}_{i-1}\hat{u}_{i-1})|\\ &= |\hat{v}_i^T(\hat{\beta}_{i-1}Q_L^T\hat{u}_{i-1})-\hat{\beta}_{i-1}^2|\\ &= |\hat{v}_i^T(\hat{\beta}_{i-1}^2\hat{v}_i+\widehat{V}_{i-1}l_{i-1}+q_{i-1})-\hat{\beta}_{i-1}^2|\\ &= \mathcal{O}(\theta_{i-1}\kappa(C)\tau).\end{aligned}$$

Therefore we obtain

$$-2(\alpha_i\beta_{i-1}u_{i-1}^Tu_i+\hat{\alpha}_i\hat{\beta}_{i-1}\hat{u}_{i-1}^T\hat{u}_i)=\mathcal{O}(\theta_{i-1}\kappa(C)\tau),$$

which is the ith diagonal of H_k. ∎

Similarly to relations (4.1) and (4.2), there are a couple of recursive relations describing the reduction process from Q_L to $\widehat{B}_k$.

THEOREM 4.7. *The following relations hold for the k-step rJBD process:*

$$Q_L\widehat{V}_k=\widehat{U}_k\widehat{B}_k, \tag{4.15}$$

$$Q_L^T\widehat{U}_k=\widehat{V}_k\left(\widehat{B}_k^T+\widehat{D}_k\right)+\hat{\beta}_k\hat{v}_{k+1}\left(e_k^{(k)}\right)^T+\widehat{G}_k, \tag{4.16}$$

where $\widehat{D}_k$ is upper triangular, and

$$\|\widehat{G}_k\|=\mathcal{O}(\|\widehat{B}_k^{-1}\|\sqrt{n}\kappa(C)\tau). \tag{4.17}$$

Proof. Relation (4.15) is just (4.3). Combining (4.1) and (4.2), we have

$$\begin{aligned}Q_A^TQ_AV_k=Q_A^TU_k(B_k) &= \left[V_k(B_k^T+D_k)+\beta_kv_{k+1}\left(e_k^{(k)}\right)^T+G_k\right]B_k\\ &= V_kB_k^TB_k+\alpha_k\beta_kv_{k+1}\left(e_k^{(k)}\right)^T+V_kD_kB_k+G_kB_k.\end{aligned}$$

Premultiplying (4.15) by Q_L^T, we have

$$Q_L^TQ_LV_k=Q_L^T\widehat{U}_k\widehat{B}_kP.$$

Adding the above two equalities and using Theorem 4.5, we obtain

$$\begin{aligned} V_k &= (Q_A^T Q_A + Q_L^T Q_L) V_k \\ &= V_k[I_k - P\widehat{B}_k^T \widehat{B}_k P + H_k] + Q_L \widehat{U}_k \widehat{B}_k P + \alpha_k \beta_k v_{k+1} \left(e_k^{(k)}\right)^T + V_k D_k B_k + G_k B_k. \end{aligned}$$

Using $\alpha_k \beta_k = \hat{\alpha}_k \hat{\beta}_k$ and $\hat{v}_{k+1} = (-1)^k v_{k+1}$, after some rearrangements we obtain

$$\widehat{V}_k \widehat{B}_k^T \widehat{B}_k = Q_L^T \widehat{U}_k \widehat{B}_k - \hat{\alpha}_k \hat{\beta}_k \hat{v}_{k+1} \left(e_k^{(k)}\right)^T + \widehat{V}_k P(D_k B_k + H_k) P + G_k B_k P.$$

Therefore, we have

$$Q_L^T \widehat{U}_k = \widehat{V}_k(\widehat{B}_k^T + \widehat{D}_k) + \hat{\beta}_k \hat{v}_{k+1} \left(e_k^{(k)}\right)^T - G_k B_k P \widehat{B}_k^{-1},$$

where $\widehat{D}_k = -P(D_k B_k + H_k) P \widehat{B}_k^{-1}$ is upper triangular. Relation (4.16) is obtained by letting $\widehat{G}_k = -G_k B_k P \widehat{B}_k^{-1}$. By Theorem 4.2 we get

$$\|B_k\| = \|\bar{U}_k^T (Q_A + E_k) V_k\| \le \|Q_A\| + \|E_k\| \le 1 + \mathcal{O}(\sqrt{n} k \kappa(C) \tau).$$

By neglecting high order terms of τ in $\|G_k B_k P \widehat{B}_k^{-1}\|$, we finally obtain the upper bound on $\|\widehat{G}_k\|$. ∎

Since $\widehat{D}_k$ is upper triangular, if we write the matrix $\widehat{D}_k$ as

$$\widehat{D}_k = \begin{pmatrix} \hat{\xi}_{11} & \cdots & \cdots & \hat{\xi}_{1k} \\ & \hat{\xi}_{22} & \cdots & \hat{\xi}_{2k} \\ & & \ddots & \vdots \\ & & & \hat{\xi}_{kk} \end{pmatrix} \in \mathbb{R}^{k \times k},$$

then Theorem 4.7 implies for each $i = 1, \ldots, k$ that

$$\hat{\beta}_i \hat{v}_{i+1} = Q_L^T \hat{u}_i - \hat{\alpha}_i \hat{v}_i - \sum_{j=1}^{i} \hat{\xi}_{ji} \hat{v}_j - \hat{g}_i$$

with $\|\hat{g}_i\| = \mathcal{O}(\|\widehat{B}_k^{-1}\| \sqrt{n} \kappa(C) \tau)$, which corresponds to the reorthogonalization of $\hat{v}_i$ with error term $\hat{g}_i$, where $\hat{\xi}_{ji}$ are coefficients appearing in the reorthogonalization. Based on (4.15) and (4.16), a result similar to that in Theorem 4.2 about the bidiagonal reduction of Q_L can also be obtained.

5. Convergence and accuracy of the approximate GSVD components. The results of section 4 can be used to investigate the convergence and accuracy of GSVD components computed by rJBD. First we give a brief review on the JBD based GSVD computation. For the GSVD (2.7) of $\{A, L\}$, let $X = (x_1, \ldots, x_n)$, $P_A = (p_{A,1}, \ldots, p_{A,m})$, and $P_L = (p_{L,1}, \ldots, p_{L,p})$. Then it can be written in the vector form

$$\begin{cases} Ax_i = c_i p_{A,i}, \\ Lx_i = s_i p_{L,i}, \\ s_i A^T p_{A,i} = c_i L^T p_{L,i} \end{cases}$$

for $i = 1, \ldots, n$, where the ith largest generalized singular value is c_i/s_i, and the corresponding generalized singular vectors are x_i, $p_{A,i}$, and $p_{L,i}$, respectively. We

also use pair $\{c_i, s_i\}$ to denote a generalized singular value. Note that each x_i satisfies the normalization condition

$$x_i^T(A^TA + L^TL)x_i = 1. \tag{5.1}$$

In this paper, we only consider approximations to $\{c_i, s_i\}$ and the corresponding right generalized singular vector x_i. In order to approximate left generalized singular vectors $p_{A,i}$ and $p_{L,i}$, a strategy extracting information from $\mathrm{span}(U_k)$ and $\mathrm{span}(\widehat{U}_k)$ is needed since U_k and $\widehat{U}_k$ are not column orthonormal. For a possibly worked method, one can refer to [3, section 6].

Assume the compact SVD of B_k is computed as

$$B_k = P_k\Theta_k Y_k^T, \quad \Theta_k = \mathrm{diag}\left(c_1^{(k)}, \ldots, c_k^{(k)}\right), \quad 1 \geq c_1^{(k)} > \cdots > c_k^{(k)} \geq 0, \tag{5.2}$$

where $P_k = (p_1^{(k)}, \ldots, p_k^{(k)})$ and $Y_k = (y_1^{(k)}, \ldots, y_k^{(k)})$ are $k \times k$ orthogonal matrices. Since $c_i^2 + s_i^2 = 1$, we only need to compute c_i, and the approximate generalized singular values are $\{c_i^{(k)}, s_i^{(k)}\}$ with $s_i^{(k)} = (1 - (c_i^{(k)})^2)^{1/2}$. The approximate right generalized singular vectors are $x_i^{(k)} = R^{-1}V_k y_i^{(k)}$ for $i = 1, \ldots, k$. Recall from section 2 that R is invertible under the assumption that $\{A, L\}$ is a regular matrix pair. It is shown in [35] that the explicit computation of R^{-1} can be avoided to compute $x_i^{(k)}$ by solving

$$\begin{pmatrix} A \\ L \end{pmatrix} x_i^{(k)} = QRR^{-1}V_k y_i^{(k)} = \widetilde{V}_k y_i^{(k)} \tag{5.3}$$

iteratively. The above approximations can also be obtained by the SVD of $\widehat{B}_k$, which is connected to that of B_k by (4.12). Detailed discussions about the SVD of B_k and $\widehat{B}_k$ when $H_k \neq 0$ can be found in [17, section 4]. Here we do not discuss it any longer.

Now we investigate the final accuracy of computed GSVD components by the SVD of B_k. Suppose that the algorithm is stopped at the k_0th step, and the singular values and right singular vectors of B_{k_0} are $c_i^{(k_0)}$ and $w_i^{(k_0)}$ for $1 \leq i \leq k_0$. By Theorem 4.2, $c_i^{(k_0)}$ and $V_{k_0}w_i^{(k_0)}$ will approximate the SVD components of $\bar{Q}_A = Q_A + E_{k_0}$ since B_{k_0} is the Ritz–Galerkin projection of $\bar{Q}_A$ on subspaces $\mathrm{span}(\bar{U}_{k_0})$ and $\mathrm{span}(V_{k_0})$. In order to analyze the final accuracy, we use the following assumption.

Assumption 1. Denote the ith largest singular value of $\bar{Q}_A$ by $\bar{c}_i$ with corresponding right singular vector $\bar{w}_i$. We assume at the k_0th step that

$$|\bar{c}_i - c_i^{(k_0)}| \ll \tau, \quad \|\bar{w}_i - V_{k_0}y_i^{(k_0)}\| \ll \tau. \tag{5.4}$$

This assumption can always be satisfied for a sufficiently large $k_0 \leq n$, since B_{k_0} can be used to approximate the SVD components of the $m \times n$ matrix $\bar{Q}_A$.

THEOREM 5.1. *For the rJBD based GSVD computation by the SVD of B_k which stops at k_0 such that Assumption* 1 *is satisfied, it follows for any $1 \leq i \leq k_0$ that*

$$|c_i - c_i^{(k_0)}| = \mathcal{O}(\sqrt{n}k_0\kappa(C)\tau). \tag{5.5}$$

Suppose the multiplicity of $\{c_i, s_i\}$ is 1 *and* (5.3) *is solved exactly. Let $\gamma_i = \min\{c_{i-1} - c_i, c_i - c_{i+1}\}$ for $1 < i < k_0$ and $\gamma_1 = c_1 - c_2$, $\gamma_{k_0} = c_{k_0-1} - c_{k_0}$. For any $1 \leq i \leq k_0$, if $\|E_{k_0}\| < \gamma_i$, then*

$$\|x_i - x_i^{(k_0)}\|/\|R^{-1}\| = \mathcal{O}\left(\frac{\sqrt{n}k_0\kappa(C)\tau}{\gamma_i - \|E_{k_0}\|}\right). \tag{5.6}$$

Proof. Notice that the SVD of Q_A is $Q_A = P_A C_A W^T$. By the perturbation theory of singular values (see, e.g., [10, Corollary 8.6.2]), we have

$$|c_i - \bar{c}_i| \le \|Q_A - \bar{Q}_A\| = \|E_{k_0}\| = \mathcal{O}(\sqrt{n}k_0\kappa(C)\tau).$$

Therefore, under Assumption 1 we have

$$|c_i - c_i^{(k_0)}| \le |c_i - \bar{c}_i| + |\bar{c}_i - c_i^{(k_0)}| = \mathcal{O}(\sqrt{n}k_0\kappa(C)\tau).$$

For c_i that is a singular value of Q_A with multiplicity 1, by the perturbation theorem of singular vectors [5, Theorem 1.2.8], we have the perturbation bound

$$|\sin\theta(w_i, \bar{w}_i)| \le \frac{\|E_{k_0}\|}{\gamma_i - \|E_{k_0}\|},$$

which leads to $\|w_i - \bar{w}_i\| = \mathcal{O}\Big(\frac{\sqrt{n}k_0\kappa(C)\tau}{\gamma_i - \|E_{k_0}\|}\Big)$ by neglecting high order terms of τ. Note that $x_i^{(k_0)} = R^{-1}V_{k_0}y_i^{(k_0)}$. Under Assumption 1 we have

$$\begin{aligned}\|x_i - x_i^{(k_0)}\| &\le \|R^{-1}w_i - R^{-1}\bar{w}_i\| + \|R^{-1}\bar{w}_i - R^{-1}V_{k_0}y_i^{(k_0)}\| \\ &\le \|R^{-1}\|\|w_i - \bar{w}_i\| + \|R^{-1}\|\|\bar{w}_i - V_{k_0}y_i^{(k_0)}\| \\ &= \mathcal{O}\left(\frac{\|R^{-1}\|\sqrt{n}k_0\kappa(C)\tau}{\gamma_i - \|E_{k_0}\|}\right).\end{aligned}$$

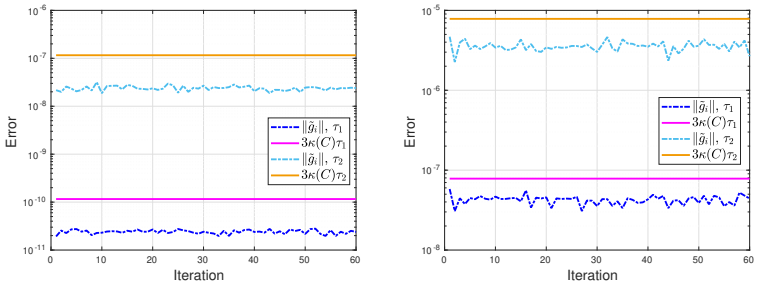
Dividing both sides by $\|R^{-1}\|$, the upper bound is obtained. ∎

We remark that the matrix-size/iteration-step dependent constant $\sqrt{n}k_0$ in $\mathcal{O}(\cdot)$ is nonessential, because it is introduced only for the end to estimate an upper bound for $\|E_{k_0}\|$. Note that the convergence rate of $c_i^{(k)}$ and $w_i^{(k)}$ as k increases from 1 to k_0 mainly depends on the convergence rate of approximating the SVD of B_k. Thus Theorem 5.1 implies that the final accuracy of approximate GSVD components is limited by $\kappa(C)\tau$ while the convergence rate is not affected too much. Combining (5.2) with (4.12) we have $\bar{B}_k^T\bar{B}_k - H_k = Y_k(I_k - \Theta_k^2)Y_k^T$, which implies that the singular values of $\bar{B}_k$ are determined by those of B_k within errors of order $\mathcal{O}(\kappa(C)\tau)$. Thus if we want to use the SVD of $\bar{B}_k$ to approximate s_i, the final accuracy is also limited by the value of $\kappa(C)\tau$; this will be illustrated by a numerical example in section 6.

Note that the normalization condition (5.1) is $x_i^T R^T R x_i = 1$. The expression (5.6) can be regarded as another form of relative error. It indicates that the final accuracy of approximate right generalized singular vectors depends not only on the value of $\kappa(C)\tau$ but also on the gap between generalized singular values. For singular values with multiplicity bigger than 1, the computation of invariant singular subspaces instead of single singular vectors is usually considered. Although in this case the mathematical expression is a bit more complicated, the spirit is similar to the approach of obtaining (5.6); interested readers can refer to [34] or [29, chapter 5.4].

Finally, we investigate the solution accuracy of (5.3) for getting the final $x_i^{(k_0)}$. Suppose (5.3) is solved iteratively using stopping criterion (3.1) with tolerance $\bar{\tau}$ and the corresponding solution is $\bar{x}_i^{(k_0)}$. Using the same approach as that for establishing (3.5), we have

$$\frac{\|x_i^{(k_0)} - \bar{x}_i^{(k_0)}\|}{\|x_i^{(k_0)}\|} \le \frac{\kappa(C)\bar{\tau}}{1 - \kappa(C)\bar{\tau}}\left(1 + \frac{\kappa(C)\|\tilde{s}_i\|}{\|C\|\|x_i^{(k_0)}\|}\right) = \frac{\kappa(C)\bar{\tau}}{1 - \kappa(C)\bar{\tau}}, \tag{5.7}$$



(a) {illc1850, well1850} (b) {dw2048, rdb2048}

(c) {swang1, $0.1L_{1d}$} (d) $\{A_1, L_1\}$

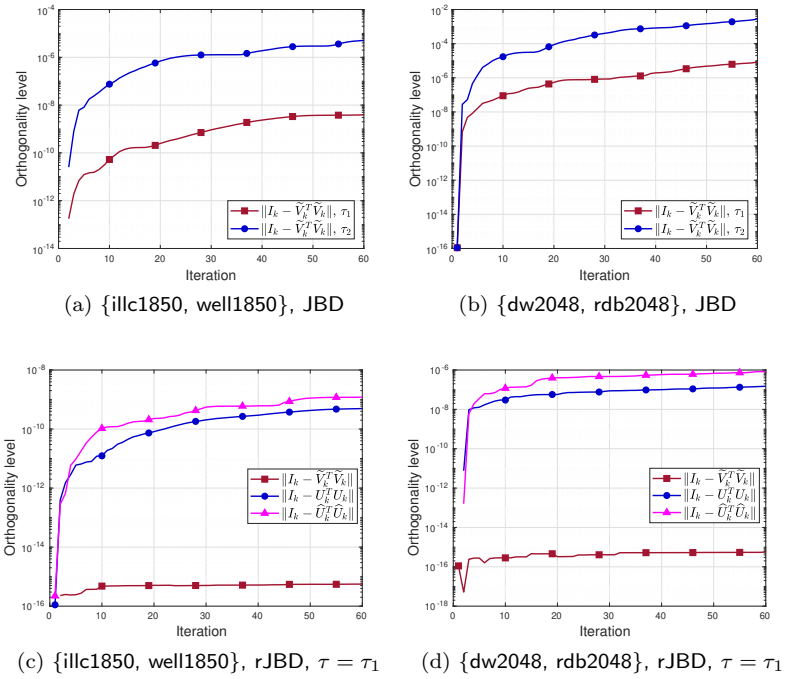
FIG. 2. *Computational error $\tilde{g}_i$ and its upper bound for JBD. For* {illc1850,well1850} and {swang1,$0.1L_{1d}$}, $\tau_1 = 10^{-12}$, $\tau_2 = 10^{-9}$; for {dw2048,rdb2048} and $\{A_1,L_1\}$, $\tau_1 = 10^{-10}$, $\tau_2 = 10^{-8}$.

where the residual $\tilde{s}_i = \widetilde{V}_k y_i^{(k_0)} - C x_i^{(k_0)} = 0$ since (5.3) is consistent. Comparing (5.6) and (5.7), the relative error $\|x_i^{(k_0)} - \bar{x}_i^{(k_0)}\|/\|x_i^{(k_0)}\|$ need not be much smaller than $\kappa(C)\tau/\gamma_i$. For a well-conditioned C, values $\bar{\tau} \in [0.1\tau, 10\tau]$ are often feasible as illustrated by experimental results.

6. Experimental results. We report some experimental results to justify the theoretical results obtained. All numerical experiments are performed in MATLAB R2019b, where all computations are carried out using double precision with roundoff unit $2^{-53} \approx 1.11 \times 10^{-16}$. The codes are available at https://github.com/Machealb/gsvd_iter.

The tested matrices are mainly taken from the SuiteSparse matrix collection [6] with the same names. The matrix $L_{1d} \in \mathbb{R}^{(n-1)\times n}$ is a bidiagonal matrix with one more row than columns and values -1 and 1 on the subdiagonal and diagonal parts, respectively. The matrix pair $\{A_1, L_1\}$ is constructed as follows. Set $m = n = p = 10000$. Let $C_A = \mathrm{diag}(\{c_i\}_{i=1}^n)$ with $c_i = (n-i+1)/2n$ and $S_L = \mathrm{diag}(\{s_i\}_{i=1}^n)$ with $s_i = (1-c_i^2)^{1/2}$. Then let D be a diagonal matrix generated by the MATLAB built-in function $D =$ `diag(linspace(1,1e5,n))`. Finally let $A_1 = C_A D$ and $L_1 = S_L D$. By the construction, the QR factorization of $C = \binom{A_1}{L_1}$ is $\binom{A_1}{L_1} = \binom{C_A}{S_L} D$ with $Q = \binom{C_A}{S_L}$, and $\kappa(C) = \kappa(D) = 10^5$. The properties of the matrices are described in Table 1. Note the scaling factor 0.1 before L_{1d}, which ensures a faster convergence of LSQR for inner iterations. For each matrix pair, we use the random vector $s =$ `randn(n,1)` with random seed `rng(2022)` as the starting vector for JBD and rJBD. The inner least squares problem is iteratively solved by the LSQR with stopping criterion (3.1).

Experiments for JBD and rJBD. We first use some numerical experiments to confirm the theoretical results about the JBD and rJBD processes. Figure 2 shows



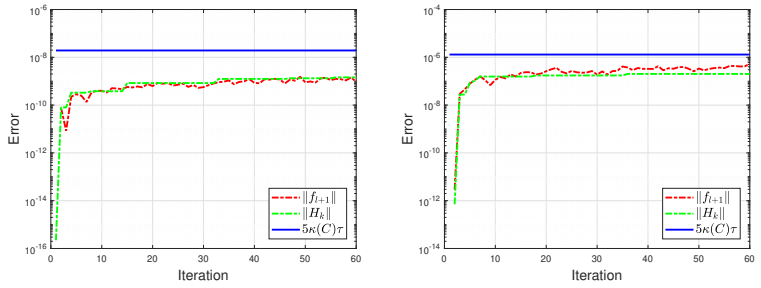
(a) {illc1850, well1850}, JBD (b) {dw2048, rdb2048}, JBD

(c) {illc1850, well1850}, rJBD, $\tau = \tau_1$ (d) {dw2048, rdb2048}, rJBD, $\tau = \tau_1$

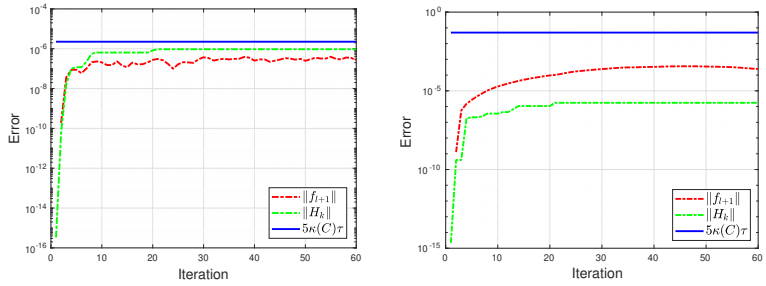
FIG. 3. *Loss of orthogonality of $\tilde{v}_i$ for JBD and orthogonality level of $\tilde{v}_i$, u_i, and $\hat{u}_i$ for rJBD.*

the numerical behavior of JBD where inner iterations are computed using LSQR with stopping tolerance τ. The computational error $\tilde{g}_i$ and its upper bound (3.3) of the four test matrix pairs are drawn. For each test matrix pair and each $\tau = \tau_1$ or $\tau = \tau_2$, we find that $\|\tilde{g}_i\|$ varies slightly and $3\kappa(C)\tau$ is an upper bound, which confirms Theorem 3.1. Note that for $\{A_1, L_1\}$ the upper bound is more overestimated than others; this is because the upper bound in (3.6) is more likely to be overestimated when the condition number is very large. In addition, in the experiments we have found that for (1.2) it takes extremely many iterations (even more than n) of LSQR to achieve the desired accuracy described by (3.1). In this case, a proper preconditioner or a scaling factor transforming $\{A, L\}$ to $\{A, \gamma L\}$ can be very useful in accelerating convergence. Figure 3 depicts the orthogonality level of $\tilde{v}_i$, u_i, and $\hat{u}_i$ measured by $\|I_k - \widetilde{V}_k^T \widetilde{V}_k\|$ and so on when inner iterations are computed inexactly. We can find that a large τ leads more quickly to a loss of orthogonality of $\tilde{v}_i$. This phenomenon has already been observed in [35], the reason for which is revealed by Theorem 3.2. For rJBD that applies full reorthogonalization to $\tilde{v}_i$, the orthogonality level of $\tilde{v}_i$ is kept around a value close to the roundoff unit, but the orthogonality of u_i and $\hat{u}_i$ still decreases gradually since reorthogonalization is not applied to them.

Figures 4 and 5 illustrate our error analysis results of rJBD. For the four test examples, the norms of both f_{l+1} and H_k grow slightly with respect to iteration number l or k, and they can be controlled by $\kappa(C)\tau$ times a moderate constant for the rJBD with not too many iterations. This confirms Theorems 4.1 and 4.5. The relation (4.12) indicates that the singular values of $\widehat{B}_k$ are determined by those of B_k with a perturbation of order $\|H_k\| = \mathcal{O}(\kappa(C)\tau)$. Therefore, together with Theorem 4.2, we can expect that the absolute errors of approximate generalized singular values computed by the SVD of B_k or $\widehat{B}_k$ are both of order $\mathcal{O}(\kappa(C)\tau)$ and the convergence

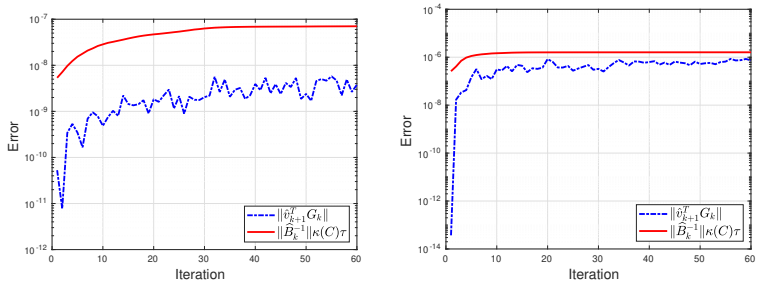


(a) {illc1850, well1850}, $\tau = 10^{-10}$ (b) {dw2048, rdb2048}, $\tau = 10^{-9}$

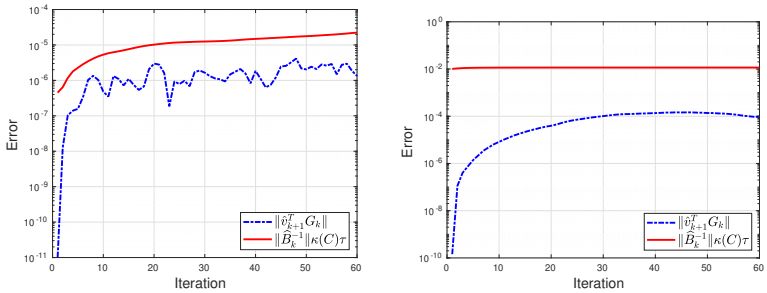


(c) {swang1, $0.1L_1$}, $\tau = 10^{-8}$ (d) $\{A_1,\ B_1\}$, $\tau = 10^{-7}$

FIG. 4. *The norms of error terms f_{l+1}, H_k and their upper bounds for rJBD.*

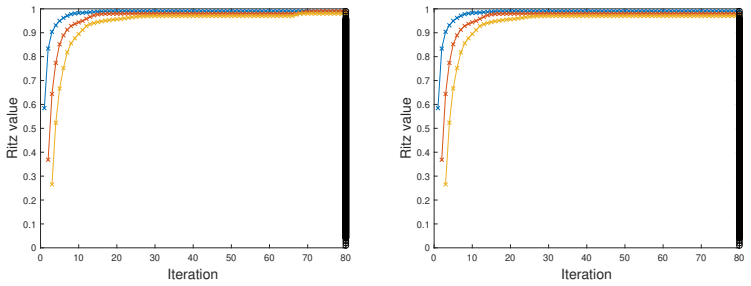


(a) {illc1850, well1850}, $\tau = 10^{-10}$ (b) {dw2048, rdb2048}, $\tau = 10^{-9}$

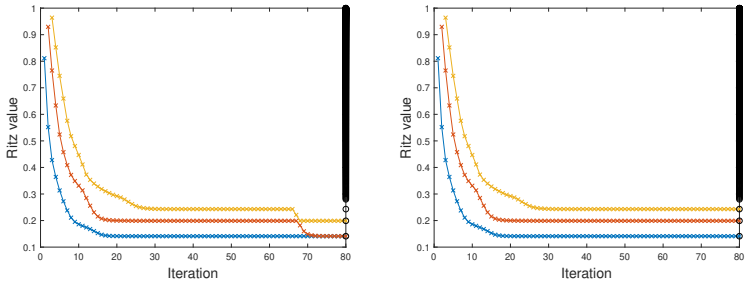


(c) {swang1, $0.1L_1$}, $\tau = 10^{-8}$ (d) $\{A_1,\ B_1\}$, $\tau = 10^{-7}$

FIG. 5. *The norms of error terms $\tilde{v}_{k+1}^T \widehat{G}_k$ and their upper bound for rJBD.*



(a) JBD (b) rJBD



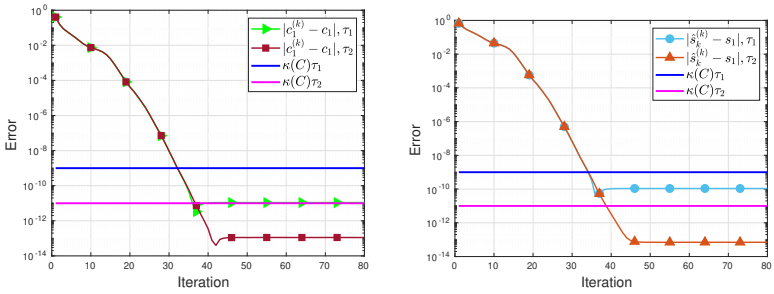
(c) JBD (d) rJBD

FIG. 6. *Convergence of Ritz values. The top two use the SVD of B_k to approximate c_1, c_2, and c_3; the bottom two use the SVD of $\widehat{B}_k$ to approximate s_1, s_2, and s_3. $\tau = 10^{-10}$.*

behaviors of the two are very similar as long as the value of $\kappa(C)\tau$ is small to a certain extent, as will be shown in Figure 6. Since $\widehat{G}_k$ cannot be computed explicitly, we depict $\|\tilde{v}_{k+1}^T \widehat{G}_k\|$ with $\tilde{v}_{k+1}^T \widehat{G}_k = \tilde{v}_{k+1}^T Q_L^T \widehat{U}_k - \hat{\beta}_k e_k^{(k)}$ by (4.16). We observe that $\|\tilde{v}_{k+1}^T \widehat{G}_k\|$ grows slightly with $\|\widehat{B}_k^{-1}\kappa(C)\tau\|$ an upper bound, which confirms Theorem 4.7. For $\{A_1, L_1\}$ the upper bounds are more overestimated than others, the reason for which is the same as in the case of $\tilde{g}_i$.

Experiments for GSVD computations. Then we illustrate the numerical behavior of the rJBD method for partial GSVD computations. The matrix pair $\{A_2, L_2\}$ used in the experiments is constructed as follows. Set $m = n = p = 800$. Let $C_A = \mathrm{diag}(\{c_i\}_{i=1}^n)$ with $c(1) = 0.99, c(2) = 0.98, c(3) = 0.97$, $c(4 : n-3) =$ `linspace(0.96,0.04,n-6)` and $c(n-2) = 0.03, c(n-1) = 0.02, c(n) = 0.01$, and $S_B = \mathrm{diag}(\{s_i\}_{i=1}^n)$ with $s_i = (1 - c_i^2)^{1/2}$. Then let W be an orthogonal matrix by letting $W =$ `gallery('orthog',n,2)` and $D =$ `diag(linspace(1,10,n))`. Finally let $A_2 = C_A W^T D$ and $L_2 = S_L W^T D$. By the construction, the generalized singular values of $\{A_2, L_2\}$ are c_i/s_i with right generalized singular vectors the ith column of $D^{-1}W$, and $\kappa(C) = 10$.

Figure 6 depicts the convergence of Ritz values, which are the first three largest singular values of B_k or the first three smallest singular values of $\widehat{B}_k$. Both of them are used to compute the first three largest generalized singular values of $\{A_2, B_2\}$ by approximating c_1, c_2, c_3 or s_1, s_2, s_3. The right vertical line indicates the values of c_i or s_i for $i = 1, \ldots, n$, and the left and right panels exhibit the convergence behaviors of JBD and rJBD for partial GSVD computations, respectively. For JBD we observe from Figure 6(a) that the second largest Ritz value suddenly jumps up at some



(a) $\tau_1 = 10^{-10}$, $\tau_2 = 10^{-12}$ (b) $\tau_1 = 10^{-10}$, $\tau_2 = 10^{-12}$

FIG. 7. *The approximate accuracy of* $c_1^{(k)}$ *by the SVD of* B_k *and of* $\hat{s}_k^{(k)}$ *by the SVD of* $\widehat{B}_k$.

iteration to continue converging to c_1, which is the so called "ghost phenomenon" due to the loss of orthogonality of Lanczos vectors. This phenomenon can be observed more clearly in Figure 6(c), which depicts the same convergence behavior as Figure 6(a), since they approximate the same generalized singular values. For the rJBD method, the convergence of Ritz values becomes regular, which is in accordance with the fact that in exact arithmetic a simple generalized singular value is approximated by Ritz values without ghosts. This property can be explained by Theorem 4.2. The convergence behaviors of approximations to the smallest generalized singular values are similar and we do not show them.

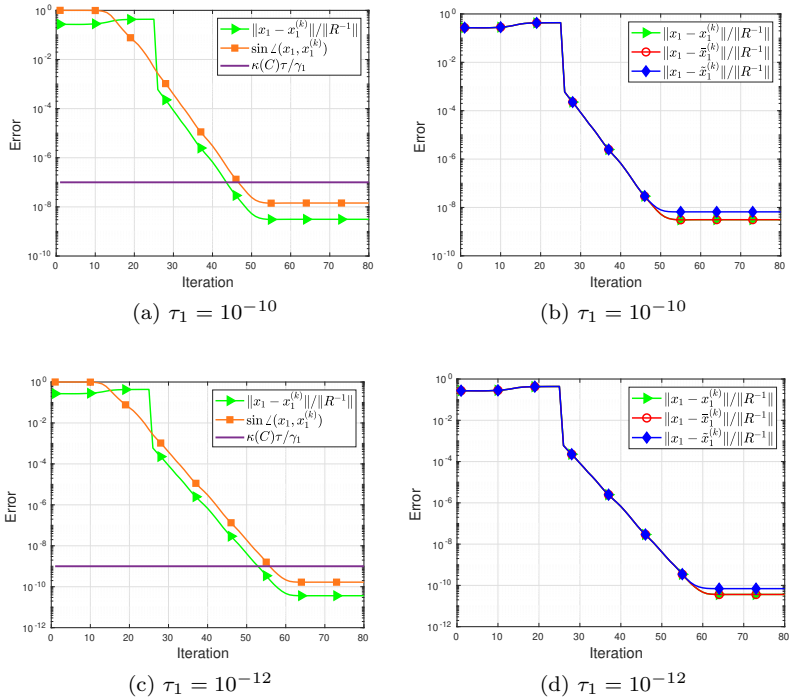
Figure 7 shows the final accuracy of $c_1^{(k)}$ for approximating c_1 and of $\hat{s}_k^{(k)}$ for approximating s_1, where the stopping tolerances for inner iterations are 10^{-10} and 10^{-12}. By Theorem 4.2, $c_1^{(k)}$ will converge to c_1 with absolute error $|c_1^{(k)} - c_1|$ of order $\mathcal{O}(\kappa(C)\tau)$, which can be clearly observed in Figure 7(a). Figure 7(b) shows the convergence and absolute error $|\hat{s}_k^{(k)} - s_1|$ of $\hat{s}_k^{(k)}$ by the SVD of $\widehat{B}_k$, which is very similar to that of $c_1^{(k)}$ but with a slight difference, due to the relation (4.12).

Finally we show the convergence and final accuracy of approximate generalized singular vectors. The stopping tolerances for inner iterations are 10^{-10} and 10^{-12}, and the approximations to the right generalized singular vector x_1 corresponding to c_1/s_1 are obtained by the SVD of B_k, where $x_1^{(k)}$ is computed by explicitly using QR factorization of C to solve (5.3), while $\bar{x}_1^{(k)}$ and $\tilde{x}_1^{(k)}$ are computed by solving (5.3) using LSQR with stopping tolerances $\bar{\tau}_1 = \tau$ and $\bar{\tau}_2 = 100\tau$. The approximation errors are also measured using

$$\sin\angle\big(x_1, x_1^{(k)}\big), \quad \sin\angle\big(x_1, \bar{x}_1^{(k)}\big), \quad \sin\angle\big(x_1, \tilde{x}_1^{(k)}\big).$$

Since $\gamma_1 \gg k\kappa(C)\tau$ for $k = 1, \ldots, 80$, in this case (5.6) becomes $\|x_1 - x_1^{(k)}\|/\|R^{-1}\| = \mathcal{O}(\sqrt{n}k\kappa(C)\tau/\gamma_1)$, and we use $\kappa(C)\tau/\gamma_1$ as an upper bound on final accuracy of approximate vectors.

From Figure 8 we can find that $x_1^{(k)}$ can eventually approximate x_1 with relative errors bounded by $\kappa(C)\tau/\gamma_1$, and the convergence rate is not affected too much by different values of τ. The computed $\bar{x}_1^{(k)}$ with $\bar{\tau}_1 = \tau$ has the same accuracy as $x_1^{(k)}$, while the accuracy of $\tilde{x}_1^{(k)}$ computed with $\bar{\tau}_1 = 100\tau$ is slightly worse. Although we do not show it here, using $\bar{\tau} = 10\tau$ to solve (5.3) can also get a vector with the same accuracy as $x_1^{(k)}$. All values of $\bar{\tau} \in [0.1\tau, 10\tau]$ are feasible for computing x_1.



(a) $\tau_1 = 10^{-10}$ (b) $\tau_1 = 10^{-10}$

(c) $\tau_1 = 10^{-12}$ (d) $\tau_1 = 10^{-12}$

FIG. 8. *The accuracy of approximations to* x_1, *where* $\bar{x}_1^{(k)}$ *and* $\tilde{x}_1^{(k)}$ *are computed from* (5.3) *by LSQR with stopping tolerance* $\bar{\tau}_1 = \tau$ *and* $\bar{\tau}_2 = 100\tau$.

7. Conclusion and outlook. For the joint bidiagonalization of a matrix pair $\{A, L\}$, we have studied the influence of inaccuracy of inner iterations on the behavior of the algorithm. For a commonly used stopping criterion with tolerance τ to describe solution accuracy of inner least squares problems, we have shown that the orthogonality of Lanczos vectors will be lost where the loss rate depends on τ and the condition number of $C = (A^T, L^T)^T$. A reorthogonalized JBD process called rJBD is proposed to keep orthogonality of $\widetilde{V}_k$, and an error analysis has been carried out to build up connections between the rJBD process and Lanczos bidiagonalizations of Q_A and Q_L, where a backward error bound about the bidiagonal reduction of Q_A is established. The results of error analysis are used to investigate the convergence and accuracy of the computed GSVD components of $\{A, L\}$ by rJBD, which shows that the approximate generalized singular values can only reach an accuracy of order $\mathcal{O}(\kappa(C)\tau)$ and the accuracy of approximate right generalized singular vectors depends not only on the value of $\kappa(C)\tau$ but also on the gap between generalized singular values, while the convergence rate is not affected very much. Some numerical experiments are made to confirm the theoretical results.

For practical JBD based GSVD computations, our results can provide a guideline for choosing a proper computing accuracy of inner iterations in order to obtain approximate GSVD components with a desired accuracy. In addition, there are still some issues that need to be considered to make the JBD method a practical GSVD algorithm. For example, an efficient procedure is needed to extract information from $\mathrm{span}(U_k)$ and $\mathrm{span}(\widehat{U}_k)$ generated by rJBD to compute left generalized singular vectors. Another issue is how to accelerate convergence of the inner iteration. A proper

preconditioner may be very useful for iteratively solving inner least squares problems. Numerical experiments show that an appropriate scaling factor transforming $\{A, L\}$ to $\{A, \gamma L\}$ has a positive effect on both the number of iterations needed by LSQR for inner iterations and the number of outer Lanczos iterations; thus scaling strategies are worth investigating. These issues constitute the subject of future research.

Acknowledgments. I thank Dr. Long Wang, Prof. Guangming Tan, and Prof. Weile Jia for their consistent support during this research. I am grateful to the anonymous referees and the editor, Prof. Arvind Saibaba, for their detailed reading of the manuscript and insightful comments that helped to improve the paper.

REFERENCES

[1] F. Alvarruiz, C. Campos, and J. E. Roman, *Thick-restarted joint Lanczos bidiagonalization for the GSVD*, J. Comput. Appl. Math., 440 (2023), 115506, https://doi.org/10.1016/j.cam.2023.115506.

[2] P. Arbenz and M. E. Hochstenbach, *A Jacobi–Davidson method for solving complex symmetric eigenvalue problems*, SIAM J. Sci. Comput., 25 (2004), pp. 1655–1673, https://doi.org/10.1137/S1064827502410992.

[3] J. L. Barlow, *Reorthogonalization for the Golub–Kahan–Lanczos bidiagonal reduction*, Numer. Math., 124 (2013), pp. 237–278.

[4] Å. Björck, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT, 28 (1988), pp. 659–670.

[5] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996, https://doi.org/10.1137/1.9781611971484.

[6] T. A. Davis and Y. Hu, *The university of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), 1, https://doi.org/10.1145/2049662.2049663, http://www.cise.ufl.edu/research/sparse/matrices.

[7] L. Elden, *Partial least-squares vs. Lanczos bidiagonalization—*I*: Analysis of a projection method for multiple regression*, Comput Statist. Data. Anal., 46 (2004), pp. 11–31.

[8] T. Ericsson, *A generalised eigenvalue problem and the Lánczos algorithm*, in Large Scale Eigenvalue Problems (Oberlech, 1985), North-Holland Math. Stud. 127, North-Holland, Amsterdam, 1986, pp. 95–119.

[9] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224, https://doi.org/10.1137/0702016.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., The Johns Hopkins University Press, Baltimore, 2013.

[11] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998, https://doi.org/10.1137/1.9780898719697.

[12] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010, https://doi.org/10.1137/1.9780898718836.

[13] U. Hetmaniuk and R. B. Lehoucq, *Uniform accuracy of eigenpairs from a shift-invert Lanczos method*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 927–948, https://doi.org/10.1137/050629288.

[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2002, https://doi.org/10.1137/1.9780898718027.

[15] M. E. Hochstenbach and Y. Notay, *Controlling inner iterations in the Jacobi–Davidson method*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 460–477, https://doi.org/10.1137/080732110.

[16] Z. Jia and H. Li, *The joint bidiagonalization process with partial reorthogonalization*, Numer. Algorithms, 88 (2021), pp. 965–992, https://doi.org/10.1007/s11075-020-01064-8.

[17] Z. Jia and H. Li, *The joint bidiagonalization method for large GSVD computations in finite precision*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 382–407, https://doi.org/10.1137/22M1483608.

[18] Z. Jia and Y. Yang, *A joint bidiagonalization based iterative algorithm for large scale general-form Tikhonov regularization*, Appl. Numer. Math., 157 (2020), pp. 159–177.

[19] M. E. Kilmer, P. C. Hansen, and M. I. Español, *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330, https://doi.org/10.1137/050645592.

[20] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.
[21] R. M. Larsen, *Lanczos Bidiagonalization with Partial Reorthogonalization*, DAIMI Report Ser., 27 (1998), no. 537.
[22] R.-C. Li, *Bounds on perturbations of generalized singular values and of associated subspaces*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 195–234, https://doi.org/10.1137/0614017.
[23] C. C. Paige, *Computational variants of the Lanczos method for the eigenproblem*, IMA J. Appl. Math., 10 (1972), pp. 373–381.
[24] C. C. Paige, *A note on a result of Sun Ji-Guang: Sensitivity of the CS and GSV decompositions*, SIAM J. Numer. Anal., 21 (1984), pp. 186–191, https://doi.org/10.1137/0721013.
[25] C. C. Paige, *A useful form of unitary matrix obtained from any sequence of unit* 2*-norm n-vectors*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 565–583, https://doi.org/10.1137/080725167.
[26] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405, https://doi.org/10.1137/0718026.
[27] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[28] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM Rev., 42 (2000), pp. 267–293, https://doi.org/10.1137/S0036144599363084.
[29] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.
[30] J.-G. Sun, *Perturbation analysis for the generalized singular value problem*, SIAM J. Numer. Anal., 20 (1983), pp. 611–625, https://doi.org/10.1137/0720041.
[31] J.-g. Sun, *Perturbation theorems for generalized singular values*, J. Comput. Math., 1 (1983), pp. 233–242.
[32] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83, https://doi.org/10.1137/0713009.
[33] D. S. Watkins, *The QR algorithm revisited*, SIAM Rev., 50 (2008), pp. 133–145, https://doi.org/10.1137/060659454.
[34] P.-Å. Wedin, *Perturbation bounds in connection with singular value decomposition*, BIT Numer. Math., 12 (1972), pp. 99–111.
[35] H. Zha, *Computing the generalized singular values/vectors of large sparse or structured matrix pairs*, Numer. Math., 72 (1996), pp. 391–417.