The joint bidiagonalization process with partial reorthogonalization

Haibo Li

Abstract The joint bidiagonalization (JBD) process is a useful algorithm for the computation of the generalized singular value decomposition (GSVD) of a matrix pair \( \{A, L\} \). However, it always suffers from rounding errors, which causes the Lanczos vectors to lose their mutual orthogonality. In order to maintain some level of orthogonality, we present a semiorthogonalization strategy. Our rounding error analysis shows that the JBD process with semiorthogonalization strategy can ensure that the convergence of the computed quantities is not affected by rounding errors and the final accuracy is high enough. Based on the semiorthogonalization strategy, we develop the joint bidiagonalization process with partial reorthogonalization (JBDPRO). In the JBDPRO algorithm, reorthogonalizations occur only when necessary, which saves a big amount of reorthogonalization work compared with the full reorthogonalization method. Numerical experiments illustrate our theory and algorithm.

Keywords JBD · GSVD · Lanczos bidiagonalization · rounding error · orthogonal level · semiorthogonalization · partial reorthogonalization · JBDPRO

Mathematics Subject Classification (2010) 15A18 · 65F15 · 65F25 · 65F50 · 65G50

1 Introduction

The joint bidiagonalization (JBD) process is a useful algorithm for computing some extreme generalized singular values and vectors for a large sparse or structured matrix pair \( \{A, L\} \) [19, 25], as well as solving large-scale discrete ill-posed problems with general form Tikhonov regularization [6, 7]. It was first proposed by Zha [27] that iteratively reduces the matrix pair \( \{A, L\} \) to an upper or lower bidiagonal form, and later adapted by Kilmer [10] that jointly diagonalizes \( \{A, L\} \) to lower and upper bidiagonal forms.

Supposing \( A \in \mathbb{R}^{m \times n} \) and \( L \in \mathbb{R}^{p \times n} \), consider the compact QR factorization of the stacked matrix:

\[
\begin{pmatrix}
A \\
L
\end{pmatrix} = QR = 
\begin{pmatrix}
Q_A \\
Q_L
\end{pmatrix}R,
\]

(1.1)

This work is supported in part by the National Science Foundation of China (No. 11771249).

Haibo Li
Department of Mathematical Sciences, Tsinghua University, 100084 Beijing, China.
E-mail: li-hb15@mails.tsinghua.edu.cn
where \( Q \in \mathbb{R}^{(m+p)\times n} \) is column orthonormal and \( R \in \mathbb{R}^{n \times n} \). We partition \( Q \) such that \( Q_L \in \mathbb{R}^{m \times n} \) and \( Q_A \in \mathbb{R}^{p \times n} \), so we have \( A = Q_A R \) and \( L = Q_Q R \). Applying the BIDIAG-1 procedure and BIDIAG-2 procedure [20], which correspond to the lower and upper Lanczos bidiagonalization processes, to \( Q_A \) and \( Q_L \), respectively, we can reduce \( Q_A \) and \( Q_L \) to the following lower and upper bidiagonal matrices respectively:

\[
B_k = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}, \quad \hat{B}_k = \begin{pmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & \\ \hat{\beta}_2 & \hat{\alpha}_2 & & \\ & \ddots & \ddots & \\ & & \ddots & \hat{\alpha}_{k-1} \\ & & & \hat{\beta}_k \end{pmatrix} \in \mathbb{R}^{k \times k}.
\] (1.2)

This process has computed four column orthonormal matrices, that is

\[
U_{k+1} = (u_1, \cdots, u_{k+1}) \in \mathbb{R}^{m \times (k+1)}, \quad V_k = (v_1, \cdots, v_k) \in \mathbb{R}^{n \times k}
\] (1.3)

computed by the BIDIAG-1 procedure, and

\[
\hat{U}_k = (\hat{u}_1, \cdots, \hat{u}_k) \in \mathbb{R}^{p \times k}, \quad \hat{V}_k = (\hat{v}_1, \cdots, \hat{v}_k) \in \mathbb{R}^{n \times k}
\] (1.4)

computed by the BIDIAG-2 procedure.

In order to join the BIDIAG-1 and BIDIAG-2 procedures, the starting vector of BIDIAG-2 is chosen to be \( \hat{v}_1 = v_1 \) and the upper bidiagonalization process continues. The Lanczos vector \( \hat{v}_k \) and the element \( \hat{\beta}_k \) of \( \hat{B}_k \) can be computed by

\[
\hat{v}_i = (-1)^{i-1}v_i, \quad \hat{\beta}_i = (\alpha_{i+1} \beta_{i+1})/\alpha_i.
\] (1.5)

If \( A \) and \( L \) are large-scale matrices, the QR factorization [11] can be avoided by solving a least squares problem with \( (A^T, L^T)^T \) as the coefficient iteratively [12] at each iteration. By the above adjustments, we can obtain the JBD process, which will be described in detail in Section 2. The \( k \)-step JBD process explicitly computes three orthonormal matrices \( U_{k+1}, V_k, \hat{U}_k \), and two bidiagonal matrices \( B_k, \hat{B}_k \), while two orthonormal matrices \( V_k \) and \( \hat{V}_k \) can be obtained from \( \hat{V}_k \) implicitly.

The JBD process can be used to approximate a few largest or smallest generalized singular values and corresponding vectors of \( \{A, L\} \) iteratively by projecting the original large-scale problem to the reduced small-scale problem \( \{B_k, \hat{B}_k\} \). Furthermore, Kilmer et al. [10] presented an iterative method based on the JBD process to solve ill-posed problems with general form Tikhonov regularization for small-scale general form Tikhonov regularization problems which lies in low dimensional subspaces. Jia and Yang [19] analyzed this iterative regularization method and presented a new iterative regularized algorithm.

In exact arithmetic, the \( k \)-step JBD algorithm is equivalent to the combination of the lower and upper Lanczos bidiagonalization processes. The lower Lanczos bidiagonalization computes two orthonormal matrices \( U_{k+1}, V_k \) and a lower bidiagonal matrix \( B_k \), while the upper Lanczos bidiagonalization computes two orthonormal matrices \( \hat{U}_k, \hat{V}_k \) and an upper bidiagonal matrix \( \hat{B}_k \). In finite precision arithmetic, however, this equivalence does not hold any longer. Supposing that the roundoff unit is \( \epsilon \), it was shown in [13] that the \( k \)-step process of computing \( U_{k+1}, V_k \) and \( B_k \) is equivalent to the upper Lanczos bidiagonalization of \( Q_A \) within error \( O(\epsilon) \), while the \( k \)-step process of computing \( \hat{U}_k, \hat{V}_k \) and \( \hat{B}_k \) is equivalent to the lower Lanczos bidiagonalization of \( Q_L \) within error \( O((m + p) \| B_k^{-1} \| \epsilon) \).
Due to the influence of rounding errors, the orthogonality among Lanczos vectors computed by the JBD process will be lost. This phenomenon was first observed in the symmetric Lanczos process [11], which was often used to compute some eigenvalues and eigenvectors of a symmetric matrix [15, 24]. The loss of orthogonality among Lanczos vectors leads to that some convergent Ritz values are additional copies of a real eigenvalue and a delay of the convergence [15, 14]. It was Paige, that has first analyzed the finite precision behavior of the symmetric Lanczos process [17]. He built a theory which links the convergence of the computed eigenvalue approximations to the loss of orthogonality, see [15, 16, 18].

Like the symmetric Lanczos process, when we use the Lanczos bidiagonalization process to approximate some singular values and vectors of a matrix [3], the loss of orthogonality among Lanczos vectors leads to the appearance of spurious singular values [12], which has been analyzed by Larsen in [12]. When we use the JBD process for the GSVD computation, the loss of orthogonality among Lanczos vectors will lead to the appearance of spurious generalized singular values and a delay of the convergence [27, 13]. It has been proved in [13] that the loss of orthogonality of Lanczos vectors \( \hat{u}_i \) and \( \hat{v}_i \) are similar as that of the Lanczos bidiagonalization process, while the orthogonal level of \( \tilde{u}_i \) is affected by the orthogonal levels of \( u_i \) and \( v_i \) and the growth of \( \| \hat{B}_k \| \).

In order to maintain some level of orthogonality of Lanczos vectors, full reorthogonalization and many other reorthogonalization strategies have been used. For the symmetric Lanczos process, Simon [22] built a recurrence formula to describe the loss of orthogonality among Lanczos vectors. His backward error analysis shows that semiorthogonality among Lanczos vectors is enough to guarantee the accuracy of the computed quantities up to machine precision and avoid the appearance of spurious eigenvalues. Based on the semiorthogonalization strategy, he introduced a new reorthogonalization method called partial reorthogonalization [23], which can save a big amount of reorthogonalization work compared with the full reorthogonalization. For the Lanczos bidiagonalization process, Larsen [12] proved that it is sufficient to maintain semiorthogonality among Lanczos vectors in order to obtain approximate singular values with accuracy up to machine precision and avoid spurious singular values from appearing. Based on the partial reorthogonalization method for the symmetric Lanczos process, he presented the Lanczos bidiagonalization with partial reorthogonalization algorithm. For the JBD process, it was shown in [13], that semiorthogonalties of the Lanczos vectors \( u_i, v_i \) and \( \hat{u}_i \) are enough to guarantee that the convergence of the computed quantities will not be affected by rounding errors and the final accuracy of approximate generalized singular values and vectors is high enough as long as \( \| \hat{B}_k \| \) does not grow too big. This result is a guidance for designing a practical semiorthogonalization strategy for the JBD process, which is the main topic of this paper.

In this paper, we propose a semiorthogonalization strategy for the \( k \)-step JBD process to keep the orthogonal levels of \( u_i, v_i \) and \( \hat{u}_i \) blow \( \sqrt{\varepsilon/(2k+1)} \). We make a rounding error analysis of the JBD process with semiorthogonalization strategy, which builds connections between the JBD process with semiorthogonalization strategy and the Lanczos bidiagonalization process in finite precision arithmetic. We prove that the computed \( B_k \) is up to round-off the Ritz-Galerkin projection of \( Q_k \) on the subspace \( \text{span}(U_{k+1}) \) and \( \text{span}(V_k) \), while \( \hat{B}_k \) is approximate the Ritz-Galerkin projection of \( Q_k \) on the subspace \( \text{span}(\hat{U}_k) \) and \( \text{span}(\hat{V}_k) \) within error \( O((m+p) \| \hat{B}_k \| \| \varepsilon \|) \), which implies that the accuracy of quantities computed from \( \hat{U}_k, \hat{V}_k \) and \( \hat{B}_k \) will not be affected, as long as \( \| \hat{B}_k \| \) does not grow too big.

Based on the semiorthogonalization strategy, we develop a practical algorithm called joint bidiagonalization process with partial reorthogonalization (JBDPRO). The central idea in partial reorthogonalization is that the levels of orthogonality among the Lanczos vectors satisfy a coupled recurrence relations, see [23, 12], which can be used as a practical tool for
computing estimates of the levels of orthogonality in an efficient way and to decide when to reorthogonalize, and which Lanczos vectors it is necessary to include in the reorthogonalization step. Numerical experiments shows that our JBDPRO algorithm is more efficient than the joint bidiagonalization with full reorthogonalization (JBDPRO) while can computes approximate generalized singular values and vectors with the same accuracy.

This paper is organized as follows. In Section 2 we review the JBD process and its properties in both exact and finite precision arithmetic. In Section 3 we propose a semiorthogonalization strategy, and make a detailed analysis of the JBD process with semiorthogonalization strategy in finite precision. Base on the semiorthogonalization strategy, in Section 4, we develop the JBDPRO algorithm. In Section 5, we use some numerical examples to illustrate our theory and algorithm. Finally, we conclude the paper in Section 6.

In this paper, the norm notation \( \| \cdot \| \) always refers to \( \| \cdot \|_2 \), which denotes the Euclidean norm for vectors or the associated matrix norm. The machine precision, which is also called roundoff unit, is denoted by the Greek letter \( \epsilon \).

2 Joint bidiagonalization process and GSVD computation

In this section, we describe the JBD process and its basic properties in both exact and finite precision arithmetic. We also describe the GSVD computation of \( \{A, L\} \) based on the JBD process.

Supposing that the compact QR factorization of \( (A^T, L^T)^T \) is as (1.1), the initial idea of the JBD process is to reduce matrices \( A \) and \( L \) to lower and upper bidiagonal forms simultaneously, by applying the BIDIAG-1 procedure and BIDIAG-2 procedure [20], which correspond to the lower and upper Lanczos bidiagonalizations of \( Q_A \) and \( Q_L \) respectively. For large-scale matrices \( A \) and \( L \), the explicitly QR factorization (1.1), which is impractical due to efficiency and storage, can be avoided by solving a least squares problem with \( (A^T, L^T)^T \) as the coefficient matrix iteratively at each iteration. This is summarized in Algorithm 1.

Algorithm 1 The \( k \)-step joint bidiagonalization (JBD) process

1: Choosing a starting vector \( b \in \mathbb{R}^m, \beta_1 u_1 = b, \beta_1 = \| b \| \)
2: \( \alpha_1 \tilde{v}_1 = QQ^T \begin{pmatrix} u_1 \\ 0_p \end{pmatrix} \)
3: \( \hat{\alpha}_1 \hat{u}_1 = \tilde{v}_1(m+1:m+p) \)
4: for \( i = 1, 2, \ldots, k \) do
5: \( \beta_{i+1} u_{i+1} = \tilde{v}_i(1:m) - \alpha_i u_i \)
6: \( \alpha_{i+1} \tilde{v}_{i+1} = QQ^T \begin{pmatrix} u_{i+1} \\ 0_p \end{pmatrix} - \beta_{i+1} \tilde{v}_i \)
7: \( \hat{\beta}_i = (\alpha_{i+1} \beta_{i+1}) / \hat{\alpha}_i \)
8: \( \hat{\alpha}_{i+1} \hat{u}_{i+1} = (-1)^i \tilde{v}_{i+1}(m+1:m+p) - \hat{\beta}_i \hat{u}_i \)
9: end for

In Algorithm 1 we need to compute \( QQ^T \begin{pmatrix} u_i \\ 0_p \end{pmatrix} \) at each iteration \( i = 1, 2, \ldots, k \), which is not directly accessible since \( Q \) is not available. Notice that \( QQ^T \begin{pmatrix} u_i \\ 0_p \end{pmatrix} \) is just the orthogonal
projection of \( \begin{pmatrix} u_i \\ 0_p \end{pmatrix} \) onto the column space of \( \begin{pmatrix} A \\ L \end{pmatrix} \), so \( QQ^T \begin{pmatrix} u_i \\ 0_p \end{pmatrix} = \begin{pmatrix} A \\ L \end{pmatrix} \tilde{x}_i \), where
\[
\tilde{x}_i = \arg \min_{\tilde{x} \in \mathbb{R}^n} \left\| \begin{pmatrix} A \\ L \end{pmatrix} \tilde{x} - \begin{pmatrix} u_i \\ 0_p \end{pmatrix} \right\|.
\]

The large-scale least squares problem (2.2) can be solved using the most commonly iterative solver LSQR algorithm [20].

In exact arithmetic, the \( k \)-step JBD process computes the lower bidiagonal matrix \( B_k \) and upper bidiagonal matrix \( \hat{B}_k \), as well as three column orthonormal matrices
\[
U_{k+1} = (u_1, \cdots, u_{k+1}) \in \mathbb{R}^{m \times (k+1)},
\]
\[
\hat{U}_k = (\hat{u}_1, \cdots, \hat{u}_k) \in \mathbb{R}^{h \times k},
\]
\[
\hat{V}_k = (\hat{v}_1, \cdots, \hat{v}_k) \in \mathbb{R}^{a \times k}.
\]

If we write all the \( k \) recurrences in matrix form, we have
\[
(l_m, 0_{m \times p}) \hat{V}_k = U_{k+1}B_k,
\]
\[
QQ^T \begin{pmatrix} U_{k+1} \\ 0_{p \times (k+1)} \end{pmatrix} = \hat{V}_k \hat{B}_k^T + \alpha_{k+1} \hat{v}_{k+1} \hat{e}_k^T,
\]
\[
(0_{p \times m}, I_p) \hat{V}_k = \hat{U}_k \hat{B}_k,
\]
where \( P = \text{diag}(1, -1, 1, \ldots, (-1)^{k-1})_{k \times k} \). Noticing that \( \hat{v}_i \) is in the subspace spanned by the columns of \( \hat{Q} \) for \( i = 1, \ldots, k \), supposing \( \tilde{v}_i = Q \hat{v}_i \) and let \( \tilde{v}_i = (-1)^{i-1} v_i \), then the following two matrices
\[
V_k = (v_1, \cdots, v_k) \in \mathbb{R}^{a \times k}, \quad \tilde{V}_k = (\tilde{v}_1, \cdots, \tilde{v}_k) \in \mathbb{R}^{a \times k}
\]
are column orthonormal and \( \hat{V}_k = V_k P \). In exact algorithm, one can obtain from relations (2.3)–(2.5) that
\[
\begin{align*}
Q_A V_k &= U_{k+1} B_k, \\
Q_A^T U_{k+1} &= V_k B_k^{\tilde{T}} + \alpha_{k+1} \tilde{v}_{k+1} \tilde{e}_k, \\
Q_A \hat{V}_k &= \hat{U}_k \hat{B}_k, \\
Q_A^T \hat{U}_k &= \hat{V}_k \hat{B}_k^{\tilde{T}} + \tilde{\alpha}_{k+1} \tilde{e}_k.
\end{align*}
\]

Relations (2.7) and (2.3) indicate that the process of computing \( U_{k+1}, V_k \) and \( B_k \) is equivalent to the lower Lanczos bidiagonalization of matrix \( Q_A \), while relations (2.9) and (2.10) indicate that the process of computing \( \hat{U}_{k+1}, \hat{V}_k \) and \( \hat{B}_k \) is equivalent to the upper Lanczos bidiagonalization of matrix \( Q_L \). Therefore, the JBD process is a naturally generalization of the Lanczos bidiagonalization process.

The JBD process can be used to approximate some extreme generalized singular values and vectors of a large sparse or structured matrix pair \( \{A, L\} \) [27][13]. We first describe the GSVD of \( \{A, L\} \). Let
\[
Q_A = P_A C_A W^T, \quad Q_L = P_L S_L W^T
\]
be the CS decomposition of the matrix pair \( \{Q_A, Q_L\} \) [4][26], where \( P_A = (p_1^A, \ldots, p_m^A) \in \mathbb{R}^{m \times m}, P_L = (p_1^L, \ldots, p_m^L) \in \mathbb{R}^{m \times p} \) and \( W = (w_1, \ldots, w_p) \in \mathbb{R}^{p \times n} \) are orthogonal matrices, and \( C_A \in \mathbb{R}^{m \times n} \) and \( S_L \in \mathbb{R}^{a \times n} \) are diagonal matrices(not necessarily square) satisfying \( C_A^T C_A + \)
which is almost always true in practice, and the rare cases where $B$ can be achieved by a direct method since

$$A = P_S C_S G^{-1}, \quad L = P_S S_L G^{-1} \quad (2.12)$$

with $G = R^{-1} W \in \mathbb{R}^{n \times n}$, where the invertibility of $G$ follows from the assumption that $(A^T, L^T)^T$ has full rank. The $i$-th generalized singular value of $(A, L)$ is $c_i/s_i$, while the $i$-th corresponding generalized right singular vector is $g_i = R^{-1} w_i$ and left vectors of $A$ and $L$ are $p_i$ and $t_i$ respectively. We point out that if $s_i = 0$, the generalized singular value $c_i/s_i$ is $\infty$.

After the $k$-step JBD process of $(A, L)$, we have computed $B_k$ and $\hat{B}_k$. Suppose that we have computed the SVD of $B_k$:

$$B_k = P_k \Theta_k W_k^T, \quad \Theta_k = \text{diag}(c_1^{(k)}, \ldots, c_k^{(k)}), \quad 1 \geq c_1^{(k)} > \cdots > c_k^{(k)} \geq 0, \quad (2.13)$$

where $P_k = (p_1^{(k)}, \ldots, p_k^{(k)})$ and $W_k = (w_1^{(k)}, \ldots, w_k^{(k)})$ are orthogonal. The decomposition $(2.13)$ can be achieved by a direct method since $B_k$ is a matrix of small scale. The approximate generalized singular value of $(A, L)$ is $c_i^{(k)}/\sqrt{1 -(c_i^{(k)})^2}$, while the approximate right generalized singular vector $x_i^{(k)} = R^{-1} V_i w_i^{(k)}$ and the approximate left generalized singular vector of $A$ is $y_i^{(k)} = U_{k+1} p_i^{(k)}$.

For large-scale matrices $A$ and $L$, the explicit computation of $R^{-1}$ can be avoided. Notice that

$$(A, L) x_i^{(k)} = QRR^{-1} V_i w_i^{(k)} = V_i w_i^{(k)},$$

hence by solving a least squares problem, we obtain $x_i^{(k)}$ from $V_i w_i^{(k)}$.

If we also want to compute the approximations of the left generalized singular vectors of $L$, we need to compute the SVD of $\hat{B}_k$. The approximate generalized singular values and corresponding right vectors can also be computed from the SVD of $\hat{B}_k$. The procedure is similar as the above and we omit it.

In finite precision arithmetic, however, the orthogonality among Lanczos vectors computed by the JBD process will be lost, and relations $2.9 \sim 2.10$ will not hold any longer. The rounding error analysis of the JBD process in finite precision arithmetic is based on a set of assumptions and properties of the behavior of the rounding errors occurring, which constitutes a rational model for the actual computation. We state them here following $[13]$.

The first property is that

$$\| B_k \| \leq 1 + O(\epsilon), \quad (2.14)$$

Second, the property of local orthogonality of $u_i$ and $\hat{u}_i$ holds, that is,

$$\beta_{k+1} u_{k+1}^T u_i = O(m \| Q_k \| \epsilon), \quad \hat{\alpha}_{k+1} \hat{u}_{k+1}^T \hat{u}_i = O(p \| Q_k \| \epsilon), \quad (2.15)$$

Third, it is assumed that

$$no \ \alpha_i, \ \beta_{k+1}, \ \hat{\alpha}_i \ or \ \hat{\beta}_i \ ever \ become \ negligible,$$

which is almost always true in practice, and the rare cases where $\alpha_i, \beta_{k+1}, \hat{\alpha}_i$ or $\hat{\beta}_i$ do become smaller are actually the lucky ones, since then the algorithm should be terminated, having found an invariant singular subspace.
We also assume that the inner least squares problem (2.1) is always solved accurately. It was shown in [13] that the following four relations hold:

\begin{align}  
Q_k V_k &= U_{k+1} B_k + F_k, \quad (2.18) \\
Q_k^T U_{k+1} &= V_k B_k^T + \alpha_{k+1} V_{k+1} e_{k+1}^T + G_{k+1}, \quad (2.19) \\
Q_k V_k &= \tilde{U}_k \tilde{B}_k + \tilde{F}_k, \quad (2.20) \\
Q_k^T \tilde{U}_k &= \tilde{V}_k \tilde{B}_k^T + \beta_{k+1} e_{k+1}^T + \tilde{G}_k, \quad (2.21) 
\end{align}

where

\[
F_k = (f_1, \ldots, f_k), \quad G_{k+1} = (g_1, \ldots, g_{k+1}), \\
\tilde{F}_k = (\tilde{f}_1, \ldots, \tilde{f}_k), \quad \tilde{G}_k = (\tilde{g}_1, \ldots, \tilde{g}_k),
\]

and

\[
f_i, g_i = O(\|Q_k\| \|\varepsilon\|) = O(\varepsilon), \quad \tilde{f}_i = O(\|Q_k\| \|\varepsilon\|) = O(\varepsilon), \\
\tilde{g}_i = O((m + p) \|\tilde{B}_k^{-1}\| \|\varepsilon\|)
\]

for \( i = 1, \ldots, k \) and \( g_{k+1} \). Relations (2.18) and (2.19) indicate that the process of computing \( U_{k+1}, V_k \) and \( B_k \) is equivalent to the lower Lanczos bidiagonalization of \( Q_k \) within error \( O(\varepsilon) \), while (2.20) and (2.21) indicate that the process of computing \( \tilde{U}_k, \tilde{V}_k \) and \( \tilde{B}_k \) is equivalent to the upper Lanczos bidiagonalization of \( Q_k \) within error \( O((m + p) \|\tilde{B}_k^{-1}\| \|\varepsilon\|) \).

In finite precision arithmetic, if we use the JBD process to approximate some extreme generalized singular values and vectors of \( \{A, L\} \), the loss of orthogonality of Lanczos vectors will lead to the appearance of additional copies of approximations of the real generalized singular values, which are called "ghosts", see [13].

Following [12, 13], the orthogonal level of Lanczos vectors are defined as follows.

**Definition 2.1.** Let \( \mu_j = u_j^T u_j \), define the orthogonal level of Lanczos vectors \( \{u_1, \ldots, u_k\} \) be

\[
\omega_k = \max_{1 \leq i < j \leq k} |\mu_{ij}|.
\]

The above definition is also used for \( \{\tilde{v}_1, \ldots, \tilde{v}_k\} \) and \( \{\tilde{u}_1, \ldots, \tilde{u}_k\} \), and we use notations \( \tilde{\omega}_k \) and \( \hat{\omega}_k \) to denote the above two quantities.

In the following analysis, we often use terminology “the orthogonal level of \( u_i \)” for simplicity, which means the orthogonal level of \( \{u_1, \ldots, u_k\} \).

To avoid “ghosts” from appearing, one can use the full reorthogonalization for \( u_i, \tilde{u}_i \) and \( \tilde{v}_i \) at each iteration, to make sure that the orthogonal levels of \( u_i, \tilde{u}_i \) and \( \tilde{v}_i \) are about \( O(\varepsilon) \). The disadvantage of full reorthogonalization strategy is that it will cause too much additional computation. It has been shown in [13] that semiorthogonalties of \( u_i, \tilde{u}_i \) and \( \tilde{v}_i \) are enough to guarantee the accuracy of the approximate generalized singular values and avoid “ghosts” from appearing.

**Theorem 2.1** Given the \( k \)-step JBD process of \( \{A, L\} \) on a computer with roundoff unit \( \varepsilon \), supposing that the inner least squares problem (2.1) is solved accurately, let \( V_k = Q^T \tilde{V}_k, \tilde{V}_k = V_k P, \) where \( P = \text{diag}(1, -1, 1, \ldots, (-1)^{k-1})_{k \times k} \), let the compact QR factorizations of \( U_{k+1}, V_k \), \( \tilde{U}_k, \tilde{V}_k \) are \( U_{k+1} = M_{k+1} R_{k+1}, \quad V_k = N_k S_k, \quad \tilde{U}_k = \tilde{M}_k \tilde{R}_k, \quad \tilde{V}_k = \tilde{N}_k \tilde{S}_k \), where the diagonals of upper triangular matrices are nonnegative, let \( \delta = O((m + p) \|\tilde{B}_k^{-1}\| \|\varepsilon\|) \). If

\[
\omega_{k+1}, \tilde{\omega}_k \leq \sqrt{\frac{\varepsilon}{2k+1}}, \quad \hat{\omega}_k \leq \sqrt{\frac{\delta}{2k+1}},
\]

...
then
\[ M_k^T Q_k N_k = B_k + E_k, \quad E_k = O(\varepsilon), \quad (2.26) \]
and
\[ M_k^T Q_k \tilde{N}_k = \tilde{B}_k + \tilde{E}_k, \quad \tilde{E}_k = O(\delta), \quad (2.27) \]
where the notation \( M = O(\sigma) \) means that the elements of matrix \( M \) are of \( O(\sigma) \).

The theorem shows that if the Lanczos vectors are kept semiorthogonal, the computed \( B_k \) is up to roundoff the Ritz-Galerkin projection of \( Q \) on the subspaces \( \text{span}(U_{k+1}) \) and \( \text{span}(V_k) \), while the computed \( \tilde{B}_k \) is the Ritz-Galerkin projection of \( Q \) on the subspaces \( \text{span}(\tilde{U}_k) \) and \( \text{span}(\tilde{V}_k) \) within error \( O(\delta) \). Therefore, we can use the SVD of \( B_k \) or \( \tilde{B}_k \) to approximate the generalized singular values and vectors of \( \{A, L\} \) and avoid “ghosts” from appearing. The final accuracy of approximated quantities computed from \( \tilde{B}_k \) will be high enough as long as \( \| \tilde{B}_k^{-1} \| \) does not grow too big.

In the next section, we will propose a semiorthogonalization strategy as well as make a detailed analysis of the JBD process equipped with the semiorthogonalization strategy.

### 3 A Semiorthogonalization strategy for the JBD process

For the \( k \)-step JBD process, in order to keep the orthogonal level of \( \{u_1, \ldots, u_k\} \) below
\[ \omega_0 = \sqrt{\varepsilon / (2k + 1)} , \]
the \( i \)-th step we need to use a reorthogonalization strategy for \( u_{i+1} \), see [22, 12, 24]. At the \( i \)-th step, supposing that
\[ \beta_{i+1}^j u_{i+1} = \tilde{v}_i (1 : m) - \alpha_i u_i - f_i' \]
if \( |u_{i+1}^T u_j| > \omega_0 \) for some \( j < i \), then we choose \( i - 1 \) real numbers \( \xi_1, \ldots, \xi_{i-1} \), and form
\[ \beta_{i+1} u_{i+1} = \beta_{i+1}^j u_{i+1} - \sum_{j=1}^{i-1} \xi_j u_j - f_i'' \]
where \( f_i' \) and \( f_i'' \) are rounding errors appeared in the computation. The algorithm will be continued with \( u_{i+1} \) instead of \( u_{i+1}' \).

**Definition 3.1.** The above modification of the JBD process will be called a semiorthogonalization strategy for \( u_{i+1} \) if the following conditions are satisfied:

1. The numbers \( \xi_1, \ldots, \xi_{i-1} \) are chosen such that
\[ u_{i+1}^T u_j \leq \omega_0, \quad j = 1, \ldots, i. \]
\[ (3.3) \]

2. The computation of \( \xi_j, \quad j = 1, \ldots, i - 1, \) and the formulation of \( u_{i+1} \) causes at most rounding errors of \( O(\varepsilon) \), i.e. we have
\[ \beta_{i+1} u_{i+1} = \tilde{v}_i (1 : m) - \alpha_i u_i - \sum_{j=1}^{i-1} \xi_j u_j - f_i \]
where \( f_i = O(\varepsilon) \).
The semiorthogonalization strategy for $\tilde{v}_{i+1}$ and $\tilde{u}_{i+1}$ are similar, and the corresponding $i$-th step recurrences are
\begin{equation}
\alpha_{i+1} \tilde{v}_{i+1} = QQ^T \left( \begin{array}{c} u_{i+1} \\ 0_p \end{array} \right) - \beta_{i+1} \tilde{v}_i - \sum_{j=1}^{i-1} \eta_{j+1} \tilde{v}_j - g_{i+1} , \end{equation}
(3.5)
\begin{equation}
\hat{\alpha}_{i+1} \hat{u}_{i+1} = (-1)^i \tilde{v}_{i+1}(m+1 : m+p) - \hat{\beta}_i \hat{u}_i - \sum_{j=1}^{i-1} \hat{\eta}_{j+1} \hat{u}_j - \hat{f}_{i+1} , \end{equation}
(3.6)
where $g_{i+1}, \hat{f}_{i+1} = O(\varepsilon)$. We point out that the rounding error terms $f_i, g_{i+1}$ and $\hat{f}_{i+1}$ here are different from that appeared in relations $\ref{2.18}$–$\ref{2.23}$, we use the same notations just for simplicity.

Notice that the reorthogonalization of $u_{i+1}$ does not use the vector $u_i$, due to the property of local orthogonality among $u_i$ and $u_{i+1}$. The reasons are similar for the reorthogonalizations of $\tilde{v}_{i+1}$ and $\tilde{u}_{i+1}$. After the semiorthogonalization step, relations $\ref{2.15}$ and $\ref{2.16}$ will still hold.

**Remark 3.1** Since $\hat{\delta} = O((m+p) \parallel \hat{B}_k^{-1} \parallel \varepsilon)$ is dependent on $\parallel \hat{B}_k^{-1} \parallel$, it is difficult to keep orthogonal level of $\{\hat{u}_1, \ldots, \hat{u}_k\}$ just below $\sqrt{\hat{\delta}/(2k+1)}$. We simply use $\omega_0$ instead of $\sqrt{\hat{\delta}/(2k+1)}$, which is also fruitful.

### 3.1 Properties of the semiorthogonalization strategy

After $k$ steps, we have computed three groups of Lanczos vectors $\{u_1, \ldots, u_{k+1}\}$, $\{\tilde{v}_1, \ldots, \tilde{v}_k\}$, and $\{\hat{u}_1, \ldots, \hat{u}_k\}$, of which orthogonal levels are below $\omega_0$. Further more, from $\ref{3.3}$–$\ref{3.6}$ we have the following relations:
\begin{align}
(I_m, 0_{m \times p}) \tilde{V}_k &= U_{k+1} (B_k + C_k) + F_k ,
\end{align}
(3.7)
\begin{align}
QQ^T \left( \begin{array}{c} U_{k+1} \\ 0_{p \times (k+1)} \end{array} \right) &= \tilde{V}_k (B_k^T + D_k) + \alpha_{k+1} \tilde{v}_{k+1} e_{k+1}^T + G_{k+1} ,
\end{align}
(3.8)
\begin{align}
(0_{p \times m}, I_p) \tilde{V}_k P &= \hat{U}_k (\hat{B}_k + \hat{C}_k) + \hat{F}_k ,
\end{align}
(3.9)
where $F_k = (f_1, \ldots, f_k)$, $G_{k+1} = (g_1, \ldots, g_{k+1})$, $\tilde{F}_k = (\tilde{f}_1, \ldots, \tilde{f}_k)$, and
\begin{align}
C_k &= \begin{pmatrix} 0 & \tilde{g}_{12} & \cdots & \tilde{g}_{1k} \\ 0 & 0 & \cdots & \tilde{g}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{(k+1) \times k} , \quad \hat{C}_k &= \begin{pmatrix} 0 & \hat{g}_{13} & \cdots & \hat{g}_{1k} \\ 0 & 0 & \cdots & \hat{g}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{k \times k} ,
\end{align}
(3.10)
\begin{align}
D_k &= \begin{pmatrix} 0 & 0 & \cdots & \eta_{1k+1} \\ 0 & 0 & \cdots & \eta_{2k+1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{k \times (k+1)} ,
\end{align}
(3.11)
For the JBD process with semiorthogonalization strategy, noticing that \( \hat{v}_i \) is approximately in the subspace spanned by the columns of \( Q \) within error \( O(\epsilon) \) for \( i = 1, \ldots, k \), so let \( v_i = Q^T \hat{v}_i \) and \( \hat{v}_i = (1)^{i-1}v_i \), then \( \{v_1, \ldots, v_k\} \) and \( \{\hat{v}_1, \ldots, \hat{v}_k\} \) are kept semiorthogonal. Moreover, from relations (3.7)–(3.9) we can obtain that

\[
Q^T \hat{u}_i \in \text{span}\{\hat{v}_i, \ldots, \hat{v}_{i+1}\} + O(\epsilon) .
\]  

**Proof.** We prove (3.14) by mathematical induction. Consider the base case \( i = 1 \), we have

\[
\begin{align*}
\hat{a}_i Q^T \hat{u}_i &= Q^T \sigma_i \hat{v}_1 - Q^T \hat{f}_1 \\
&= (I - Q^T \sigma_i Q_i) \hat{v}_1 - Q^T \hat{f}_1 \\
&= \hat{v}_1 - \sum_{j=1}^{i-1} \beta_{ji} Q^T \hat{u}_j - Q^T \hat{f}_1 .
\end{align*}
\]

Next, suppose (3.14) is true for indices up to \( i \). For \( i+1 \), we have

\[
\begin{align*}
\hat{a}_{i+1} Q^T \hat{u}_{i+1} &= Q^T \sigma_{i+1} \hat{v}_{i+1} - \hat{\beta}_i Q^T \hat{u}_i - \sum_{j=1}^{i-1} \hat{\xi}_{ji+1} Q^T \hat{u}_j - Q^T \hat{f}_{i+1} .
\end{align*}
\]

Since \( \langle \hat{\beta}_i Q^T \hat{u}_i - \sum_{j=1}^{i-1} \hat{\xi}_{ji+1} Q^T \hat{u}_j, \hat{v}_{i+1} \rangle \in \text{span}\{\hat{v}_i, \ldots, \hat{v}_{i+1}\} + O(\epsilon) \), we only need to prove that \( Q^T \sigma_{i+1} \hat{v}_{i+1} \in \text{span}\{\hat{v}_i, \ldots, \hat{v}_{i+2}\} + O(\epsilon) \). Notice that

\[
\begin{align*}
Q^T \sigma_{i+1} \hat{v}_{i+1} &= (I - Q^T \sigma_i Q_i) \hat{v}_{i+1} \\
&= \hat{v}_{i+1} + (-1)^{i+1} \sum_{j=1}^{i} \hat{\beta}_{i+1} Q^T \hat{u}_{i+1} + \sum_{j=1}^{i} \hat{\xi}_{ji+1} Q^T \hat{u}_j + f_{i+1} ,
\end{align*}
\]

from (3.12), we have

\[
(\alpha_{i+1} Q^T \hat{u}_{i+1} + \beta_{i+1} Q^T \hat{u}_{i+1} + \sum_{j=1}^{i} \xi_{ji+1} Q^T \hat{u}_j) \in \text{span}\{\hat{v}_i, \ldots, \hat{v}_{i+2}\} + O(\epsilon) ,
\]
which completes the proof of the induction step.

By mathematical induction principle, (3.14) holds for all \( i = 1, 2, \ldots \).

Now we give the bounds of the nonzero elements of matrices \( C_k, D_k \) and \( \hat{C}_k \).

**Proposition 3.1.** For the k-step JBD process with semiorthogonalization strategy, we have

\[
C_k = O(\sqrt{\kappa}) , \quad D_k = O(\sqrt{\kappa}) .
\]

**Proof.** At the \( i \)-th step, following (3.1) and (3.2), we have

\[
\beta_{i+1} u_{i+1} = \beta_{i+1} u_{i+1}^* - \sum_{j=1}^{i-1} \xi_{ij} u_j - f_i^* ,
\]

where \( f_i^*, f_i^* = O(\varepsilon) \).

We first prove

\[
\beta_{i+1} u_{i+1} = O(\varepsilon) , \quad i = 1, \ldots, i - 1.
\]

Premultiply (3.16) by \( U_i^T \), we have

\[
\beta_{i+1} u_{i+1} = U_i^T Q \alpha u_i - \alpha U_i^T u_i - U_i^T f_i
\]

so

\[\beta_{i+1} \| U_i^T u_{i+1} \| \leq [1 + O(\varepsilon)] + O(1) + O(\varepsilon) = O(\varepsilon) .\]

Now we give the bounds of the nonzero elements of matrices \( C_k, D_k \) and \( \hat{C}_k \).

**Proposition 3.1.** For the k-step JBD process with semiorthogonalization strategy, we have

\[
C_k = O(\sqrt{\kappa}) , \quad D_k = O(\sqrt{\kappa}) .
\]

**Proof.** At the \( i \)-th step, following (3.1) and (3.2), we have

\[
\beta_{i+1} u_{i+1} = \beta_{i+1} u_{i+1}^* - \sum_{j=1}^{i-1} \xi_{ij} u_j - f_i^* ,
\]

where \( f_i^*, f_i^* = O(\varepsilon) \).

We first prove

\[
\beta_{i+1} u_{i+1} = O(\varepsilon) , \quad i = 1, \ldots, i - 1.
\]

Premultiply (3.16) by \( U_i^T \), we have

\[
\beta_{i+1} u_{i+1} = U_i^T Q \alpha u_i - \alpha U_i^T u_i - U_i^T f_i
\]

so

\[\beta_{i+1} \| U_i^T u_{i+1} \| \leq [1 + O(\varepsilon)] + O(1) + O(\varepsilon) = O(\varepsilon) .\]

Then we prove \( M = \max_{1 \leq j \leq i-1} |\xi_{ij}| = O(\varepsilon) \). From (3.17), after being premultiplied by \( u_i^T \) and some rearranging, we obtain

\[
\xi_{ii} = \beta_{i+1} u_{i+1} - \beta_{i+1} u_{i+1} - \sum_{j=1}^{i-1} (u_i^T u_j) \xi_{ij} - u_i^T f_i .
\]

Noticing that \( \beta_{i+1} u_{i+1} = O(\varepsilon) \) for \( i = 1, \ldots, i - 1 \), we obtain

\[|\xi_{ii}| \leq O(\varepsilon) + O(\varepsilon) + O(\varepsilon) = O(\varepsilon) .\]

Now the right side does not depend on \( i \) anymore, and we finally obtain by taking the maximum on the left side that

\[(1 - i \sqrt{\varepsilon}) M \leq O(\varepsilon) + O(\varepsilon) ,
\]

so \( M = O(\sqrt{\varepsilon}) \).

The bound that \( \max_{1 \leq j \leq i-2} |\eta_{ij}| = O(\sqrt{\varepsilon}) \) can be obtained using the same method. \(\blacksquare\)

**Proposition 3.2.** For the k-step JBD process with semiorthogonalization strategy, we have

\[
\hat{C}_k = O(\sqrt{\varepsilon}) .
\]
Proof. At the \((i - 1)\)-th step, from (3.6) and (3.13), we can write the reorthogonalization of \(\hat{u}_i\) as

\[
\hat{\alpha}^i_0 \hat{u}_i = \hat{Q}_L \hat{\nu}_i - \hat{\beta}_{i-1} \hat{u}_{i-1} - \hat{f}_i^o ,
\]

(3.19)

\[
\hat{\alpha}^i_0 \hat{u}_i = \hat{\alpha}^i_0 \hat{u}_i - i \sum_{j=1}^{i-2} \hat{\xi}_ji \hat{a}_j - \hat{f}_i^o ,
\]

(3.20)

where \(\hat{f}_i^o, \hat{f}_i^u = O(\varepsilon)\). Thus, for \(l = 1, \ldots, i - 2\), we have

\[
\hat{\alpha}^i_0 \hat{u}_i^T \hat{\alpha}_i^T \hat{u}_i = \hat{\alpha}^i_0 \hat{u}_i^T \hat{Q}_L \hat{\nu}_i - \hat{\beta}_{i-1} \hat{u}_i^T \hat{u}_{i-1} - \hat{a}_i^T \hat{f}_i^o .
\]

From (3.14) and its proof, we know that

\[
\hat{Q}_L^T \hat{u}_i = \sum_{j=1}^{i-1} \lambda_j \hat{\nu}_j + O(\varepsilon) ,
\]

where \(\lambda_j = O(1)\) for \(j = 1, \ldots, l + 1\). Hence, noticing that \(\hat{\alpha}^i_0 \hat{u}_i^T \hat{u}_{i-1}, \hat{\nu}_j^T \hat{\nu}_j = O(\sqrt{\varepsilon})\) for \(l = 1, \ldots, i - 2\) and \(j = 1, \ldots, l + 1\), we obtain

\[
\hat{\alpha}^i_0 \hat{u}_i^T \hat{\alpha}_i^T \hat{u}_i = \sum_{j=1}^{i-1} \lambda_j \hat{\nu}_j^T \hat{\nu}_j - \hat{\beta}_{i-1} \hat{u}_i^T \hat{u}_{i-1} + O(\varepsilon) = O(\sqrt{\varepsilon}) .
\]

From (3.20), after being premultiplied by \(\hat{\alpha}_i^T\) and some rearranging, we obtain

\[
\hat{\xi}_ji = \hat{\alpha}^i_0 \hat{u}_i^T \hat{\alpha}_i^T \hat{u}_i - \hat{\alpha}_i^T \hat{u}_i - \sum_{j=1, j \neq i}^{i-2} \hat{\xi}_ji \hat{a}_j - \hat{a}_i^T \hat{f}_i^o .
\]

Now with \(\hat{\alpha}^i_0 \hat{u}_i^T \hat{\alpha}_i^T \hat{u}_i = O(\sqrt{\varepsilon})\) for \(l = 1, \ldots, i - 2\), we can prove that

\[
M = \max_{1 \leq j \leq i-2} |\hat{\xi}_ji| = O(\sqrt{\varepsilon})
\]

using the same method used in the proof of Proposition 3.1 \(\square\)

3.2 Rounding error analysis of the JBD process with semiorthogonalization strategy

From (3.11) and (3.12), we can verify that the process of computing \(U_{k+1}, V_k + B_k\) is equivalent to the lower Lanczos bidiagonalization of \(Q_k\) with semiorthogonalization strategy. Our following analysis will focus on the connections between the process of computing \(\hat{U}_k, \hat{V}_k, \hat{B}_k\) and the upper Lanczos bidiagonalization of \(Q_k\) in finite precision arithmetic.

We first give the relation between the two computed quantities \(B_k\) and \(\hat{B}_k\).

**Theorem 3.1** Given the \(k\)-step JBD process with semiorthogonalization strategy, then

\[
B_k^T B_k + P \hat{B}_k^T \hat{B}_k P = I_k + O((m + p)\varepsilon) .
\]

(3.21)
Proof. Since 

\[ B_k^T B_k = \begin{pmatrix}
\alpha_i^2 + \beta_i^2 & \alpha_i \beta_i \\
\alpha_i \beta_i & \alpha_i^2 + \beta_i^2 \\
\alpha_i \beta_i & \alpha_i^2 + \beta_i^2 \\
\alpha_i \beta_i & \alpha_i^2 + \beta_i^2 \\
\alpha_i \beta_i & \alpha_i^2 + \beta_i^2 \\
\end{pmatrix}, \]

nonzero elements in the left side of (3.21) are contained only in the diagonal and subdiagonal parts.

For the diagonal part, in finite precision arithmetic we have \( \hat{\beta}_i = (\alpha_{i+1} \beta_{i+1} / \tilde{\alpha}_i)(1 + \tau_i) \), where \( |\tau_i| \leq \gamma = 4\varepsilon/(1 - 4\varepsilon) \), see [5], so

\[ \alpha_{i+1} \beta_{i+1} = \tilde{\alpha}_i \hat{\beta}_i - \alpha_{i+1} \beta_{i+1} \tau_i. \]

Since \( |\tilde{\alpha}_i|, |\beta_{i+1}| \leq \| B_k \| \) and \( \| B_k \| \leq 1 + O(\varepsilon) \), we have

\[ \alpha_{i+1} \beta_{i+1} = \tilde{\alpha}_i \hat{\beta}_i + \gamma, \] (3.22)

where \( |\gamma| = (1 + O(\varepsilon))^2 |\tau_i| = \gamma + O(\varepsilon). \)

For the subdiagonal part, from (3.11) we have

\[ \| QAVi \|^2 = \| \alpha_i u_i + \beta_{i+1} u_{i+1} + \sum_{j=1}^{i-1} \xi_{ji} u_j + f_i \|^2 \\
= \alpha_i^2 + \beta_{i+1}^2 + 2\alpha_i \beta_{i+1} u_i^T u_j + 2\alpha_i u_i^T f_i + 2\beta_{i+1} u_{i+1}^T f_i + f_i^2 \\
+ \sum_{j=1}^{i-1} \| \xi_{ji} u_j \|^2 + 2\alpha_i \sum_{j=1}^{i-1} \xi_{ji} u_i^T u_j + 2\beta_{i+1} \sum_{j=1}^{i-1} \xi_{ji} u_{i+1}^T u_j + 2 \sum_{j=1}^{i-1} \xi_{ji} f_i^T u_j. \]

Since \( \xi_{ji} = O(\sqrt{\varepsilon}) \) by Proposition (3.11) and \( u_i^T u_j = O(\sqrt{\varepsilon}(1 \leq l \neq j \leq i + 1)) \), we obtain

\[ \sum_{j=1}^{i-1} \| \xi_{ji} u_j \|^2 + 2\alpha_i \sum_{j=1}^{i-1} \xi_{ji} u_i^T u_j + 2\beta_{i+1} \sum_{j=1}^{i-1} \xi_{ji} u_{i+1}^T u_j + 2 \sum_{j=1}^{i-1} \xi_{ji} f_i^T u_j \\
= 2 \sum_{1 \leq j \leq i-1} \xi_{ji} u_i^T u_j + \sum_{j=1}^{i-1} \| u_j \|^2 + 2\alpha_i \sum_{j=1}^{i-1} O(\varepsilon) + 2\beta_{i+1} \sum_{j=1}^{i-1} O(\varepsilon) + 2 \sum_{j=1}^{i-1} O(\varepsilon \sqrt{\varepsilon}) \\
= O(2\varepsilon \sqrt{\varepsilon}) + O(\varepsilon) + O(2\varepsilon (\alpha_i + \beta_{i+1}) \varepsilon) + O(2\varepsilon \sqrt{\varepsilon}) \\
= O(m\varepsilon), \]

where we have used \( 0 < \alpha_i + \beta_{i+1} \leq \| B_k \| \leq 1 + O(\varepsilon) \). By some simple calculation, we obtain

\[ 2\alpha_i \beta_{i+1} u_i^T u_{i+1} + 2\alpha_i u_i^T f_i + 2\beta_{i+1} u_{i+1}^T f_i + f_i^2 = O(m\varepsilon), \]

so

\[ \| QAVi \|^2 = \alpha_i^2 + \beta_{i+1}^2 + O(m\varepsilon). \] (3.23)
Applying Proposition 3.2 using the similar method as above we can obtain
\[ ||Q_k \tilde{v}_i||^2 = \hat{\alpha}_i^2 + \hat{\beta}_{i-1}^2 + O(p\epsilon) \]  
(3.24)

Noticing that
\[ ||Q_k \tilde{v}_i||^2 + ||Q_k \tilde{v}_i||^2 = v_i^T (Q_k^T Q_k + Q_k^T Q_k) v_i = v_i^T Q_k^T Q_k v_i = 1 ; \]
from (3.23) and (3.24) we have
\[ \alpha_i^2 + \beta_{i-1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_{i-1}^2 = 1 + O((m + p)\epsilon) . \]  
(3.25)

Combining (3.22) and (3.25) the result (3.21) follows.

Remark 3.2 The error bound \( O((m + p)\epsilon) \) in (3.21), which is a constant quantity, can be hardly reached in real computation. The real error of (3.21) will only grow slightly as the iteration number \( k \) increases, but not grow obviously as the matrix dimensions \( m \) and \( p \) become bigger for large sparse matrix \( A \) and \( L \), which is just the case where the algorithm is most useful.

We now show the connection between the process of computing \( \hat{U}_k, \hat{V}_k, \hat{B}_k \) and the upper Lanczos bidiagonalization of \( Q_k \) in finite precision arithmetic.

Theorem 3.2 Given the \( k \)-step JBD process with semiorthogonalization strategy, the following relation holds:
\[ Q_k^T \hat{U}_k = \hat{V}_k (B_k^T + \hat{D}_k) + \hat{B}_k \tilde{v}_{k+1} e_k^T + G_k , \]  
(3.26)

where \( \hat{D}_k \) is upper triangular with zero diagonals, and
\[ ||G_k|| = O((m + p)||\hat{B}_k^{-1}||\epsilon) . \]  
(3.27)

Proof. Combining (3.11) and (3.12), we have
\[ Q_k^T Q_k V_k = Q_k^T U_{k+1} (B_k + C_k) + Q_k^T F_k \]
\[ = [\hat{V}_k (B_k^T + D_k) + \alpha_{k+1} v_{k+1} e_{k+1} + G_{k+1}] (B_k + C_k) + Q_k^T F_k \]
\[ = \hat{V}_k B_k^T B_k + \alpha_{k+1} \hat{B}_k v_{k+1} e_{k+1}^T + \hat{V}_k (B_k^T + D_k) C_k + \hat{V}_k D_k B_k + \]
\[ G_{k+1} (B_k + C_k) + Q_k^T F_k . \]

Premultiply (3.13) by \( Q_k^T \), we have
\[ Q_k^T Q_k V_k = [Q_k^T \hat{U}_k (B_k + C_k) + Q_k^T F_k] P . \]

Adding the above two equalities, we obtain
\[ (Q_k^T Q_k + Q_k^T Q_k) V_k = V_k [I_k - \hat{P}_k^T \hat{B}_k P + O((m + p)\epsilon)] + Q_k \hat{U}_k \hat{B}_k P + (\hat{\alpha}_k \hat{B}_k + y_k) v_{k+1} e_k^T + V_k D_k B_k + \]
\[ V_k (B_k^T + D_k) C_k + Q_k^T \hat{U}_k \hat{C}_k P + G_{k+1} (B_k + C_k) + Q_k^T F_k + Q_k^T \hat{F}_k P . \]

Noticing that \( (Q_k^T Q_k + Q_k^T Q_k) V_k = V_k \), after some rearrangement we obtain
\[ \hat{V}_k \hat{B}_k^T \hat{B}_k = Q_k^T \hat{U}_k \hat{B}_k - \hat{\alpha}_k \hat{B}_k \tilde{v}_{k+1} e_k^T + E_1 + E_2 . \]
where
\[ E_1 = \hat{V}_k P [D_k B_k + (B_k^T + D_k)C_k] P + Q_k^T \hat{U}_k \hat{C}_k, \]
and
\[ E_2 = [G_{k+1} (B_k + C_k) + Q_k^T F_k + Q_k^T \hat{F}_k P + V_k O ((m + p) \varepsilon)] P - \gamma_k \hat{v}_{k+1} \varepsilon_k^T. \]

According to the structure of matrices \( C_k \) and \( D_k \) in (3.10), with simple calculation we can verify that \( P [D_k B_k + (B_k^T + D_k)C_k] P \) is an upper triangular matrix with zero diagonals, which is denoted by \( Y_k \). Noticing that the \( i \)-th column of \( Q_k^T \hat{U}_k \hat{C}_k \) is \( \sum_{j=1}^{i-2} \hat{\xi}_{ij} Q_k^T \hat{u}_j \), by Lemma 3.1 there exit coefficients \( \rho_{1i}, \ldots, \rho_{i-1,i} \) such that
\[ \sum_{j=1}^{i-2} \hat{\xi}_{ij} Q_k^T \hat{u}_j = \sum_{j=1}^{i-1} \rho_{ij} \hat{v}_j + O(\varepsilon). \]

Therefore,
\[ Q_k^T \hat{U}_k C_k = \hat{V}_k W_k + O(\varepsilon), \]
where
\[
W_k = \begin{pmatrix}
0 & \rho_{12} & \rho_{13} & \cdots & \rho_{1k} \\
0 & 0 & \rho_{23} & \cdots & \rho_{2k} \\
& & \ddots & \ddots & \ddots \\
& & & \ddots & \rho_{i-1,i} \\
& & & & 0
\end{pmatrix} \in \mathbb{R}^{i \times k}
\]
is upper triangular with zero diagonals.

For \( E_2 \), with simple but tedious calculation, we can prove that
\[ \| E_2 \| = O((m + p) \varepsilon). \]

From the above, we obtain
\[
Q_k^T \hat{U}_k - \hat{V}_k \hat{B}_k^T - \hat{B}_k \hat{v}_{k+1} \varepsilon_k^T \\
= -\hat{V}_k (W_k + Y_k) \hat{B}_k^{-1} - [E_1 + O(\varepsilon)] \hat{B}_k^{-1}.
\]

Noticing that \( -(W_k + Y_k) \hat{B}_k^{-1} \) is upper triangular with zero diagonals, which is denoted by \( \check{D}_k \), we finally obtain
\[ Q_k^T \hat{U}_k = \hat{V}_k (\hat{B}_k^T + \check{D}_k) + \hat{B}_k \hat{v}_{k+1} \varepsilon_k^T + \check{G}_k, \]
where \( \check{G}_k = -[E_1 + O(\varepsilon)] \hat{B}_k^{-1} \) and \( \| \check{G}_k \| = O((m + p) \| \hat{B}_k^{-1} \| \varepsilon). \)

If we write the matrix \( \check{D}_k \) in (3.26) as
\[
\check{D}_k = \begin{pmatrix}
0 & \hat{\eta}_{12} & \hat{\eta}_{13} & \cdots & \hat{\eta}_{1k} \\
0 & \hat{\eta}_{23} & \cdots & \hat{\eta}_{2k} \\
& & \ddots & \ddots & \ddots \\
& & & \ddots & \hat{\eta}_{k-1,k} \\
& & & & 0
\end{pmatrix} \in \mathbb{R}^{k \times k},
\]
then for each $i = 1, \ldots, k$, by (3.26), we have

$$
\hat{\beta}_i v_{i+1} = Q_k^T \hat{u}_i - \hat{\alpha} \hat{v}_i - \sum_{j=1}^{i-1} \hat{\eta}_j \hat{v}_j - \hat{g}_i,
$$

(3.29)

where $\hat{g}_i = O((m + p) \parallel \hat{B}_k^{-1} \parallel \epsilon)$, which corresponds to the semiorthogonalization step of $\hat{v}_i$ with error $\hat{g}_i$. Hence, combining (3.13) and (3.26), we can treat the process of computing $\hat{U}_k$, $\hat{V}_k$ and $\hat{B}_k$ as the upper Lanczos bidiagonalization of $Q_k$ with semiorthogonalization strategy within error $O((m + p) \parallel \hat{B}_k^{-1} \parallel \epsilon)$.

For the $k$-step lower Lanczos bidiagonalization of $Q_k$ with semiorthogonalization strategy, the computed $\hat{B}_k$ is up to roundoff the Ritz-Galerkin projection of $Q_k$ on the subspace $\text{span}(U_{k+1})$ and $\text{span}(V_k)$, see [12,13], which means that the “ghosts” can be avoided from appearing and the final accuracy of approximated quantities will not be affected when we use the SVD of $\hat{B}_k$ to approximate some generalized singular values and vectors of $\{A, L\}$.

Similarly, for $\hat{B}_k$, which is computed by the $k$-step upper Lanczos bidiagonalization of $Q_k$ with semiorthogonalization strategy within error $O((m + p) \parallel \hat{B}_k^{-1} \parallel \epsilon)$ instead of $O(\epsilon)$, we can obtain the following result.

**Theorem 3.3** Given the $k$-step JBD process with semiorthogonalization strategy, suppose that the compact QR factorizations of $\hat{U}_k$ and $\hat{V}_k$ are $\hat{U}_k = \hat{M}_k \hat{R}_k$ and $\hat{V}_k = \hat{N}_k \hat{S}_k$, where the diagonals of the upper triangular matrices $\hat{R}_k$ and $\hat{S}_k$ are nonnegative, then

$$
\hat{M}_k^T \hat{Q}_k \hat{N}_k = \hat{B}_k + \hat{E}_k,
$$

(3.30)

where the elements of $\hat{E}_k$ are of $O((m + p) \parallel \hat{B}_k^{-1} \parallel \epsilon)$.

**Theorem 3.4** indicates that the computed $\hat{B}_k$ is approximate the Ritz-Galerkin projection of $Q_k$ on the subspace $\text{span}(\hat{U}_k)$ and $\text{span}(\hat{V}_k)$ within error Lanczos bidiagonalization of $Q_k$ with semiorthogonalization strategy. Therefore, if we use the SVD of $\hat{B}_k$ to approximate some generalized singular values and vectors of $\{A, L\}$, the “ghosts” can be avoided from appearing and the final accuracy of approximated quantities is high enough, as long as $\parallel \hat{B}_k^{-1} \parallel$ does not grow too big.

### 4 The JBD process with partial reorthogonalization

In order to implement the semiorthogonalization strategy, we need to decide when to reorthogonalize, and which Lanczos vectors are necessary to include in the reorthogonalization step. By the analysis in the previous section, the process of computing $U_{k+1}$, $V_k$ and $B_k$ is equivalent to the lower Lanczos bidiagonalization of $Q_k$, so our reorthogonalization method can be based on the partial reorthogonalization of $u_i$ and $v_j$, see Larsen [23,12]. The central idea is that the levels of orthogonality among the Lanczos vectors satisfy the following coupled recurrence relations.

**Theorem 4.1** Let $\mu_{ji} = u_j^T u_i$, $v_{ji} = v_j^T v_i$ and $\mu_{j0} \equiv 0$, $v_{j0} \equiv 0$, then $\mu_{jj} = 1$ for $1 \leq j \leq i + 1$ and $v_{jj} = 1$ for $1 \leq j \leq i$, while

$$
\beta_{j+1} \mu_{j,j+1} = \alpha_j v_{ji} + \beta_j v_{j-1,i} - \alpha_i \mu_{ji} - u_j^T f_i + v_j^T g_i
$$

(4.1)

for $1 \leq j \leq i$, and

$$
\alpha_j v_{ji} = \beta_{j+1} \mu_{j+1,j} + \alpha_j \mu_{ji} - \beta_j v_{j+1,i} + u_j^T f_i + v_j^T g_i
$$

(4.2)

for $1 \leq j \leq i - 1$. 
Theorem 4.1 shows that the inner products $\alpha_{t}u_{t+1}$ and $\nu_{t}v_{t}$ are simply linear combinations of the inner products from the previous Lanczos vectors, therefore we can estimate quantities $\mu_{j,t+1}$ and $v_{j,t}$ by setting $|\alpha_{t}f_{j} - \nu_{t}g_{j}|$ and $|\alpha_{t}f_{j} - \nu_{t}g_{j}|$ appeared in (4.1) and (4.3) to $O(\varepsilon)$. One proper estimations of $|\alpha_{t}f_{j} - \nu_{t}g_{j}|$ and $|\alpha_{t}f_{j} - \nu_{t}g_{j}|$ have been given by Larsen [12].

Notice that $\tilde{v}_{i}v_{j} = v_{i}v_{j} + O(\varepsilon)$, the estimated $v_{j,t}$ is also a good estimate of $\tilde{v}_{i}v_{j}$. Therefore, using these estimates, we can monitor the loss of orthogonality of Lanczos vectors $u_{t}$ and $\tilde{v}_{i}$ directly without forming inner products, which enables us to determine when and against which of the previous Lanczos vectors to reorthogonalize. The partial reorthogonalization method is described in Algorithm [2].

Algorithm 2 The k-step JBDPRO

1: Choosing a starting vector $b \in \mathbb{R}^{m}$, $\beta_{1}u_{1} = b$, $\beta_{1} = \|b\|
2: \alpha_{1}v_{1} = \frac{Q_{T}^{T}\left(\alpha_{1}\right)}{\beta_{1}}
3: \hat{\alpha}_{i}u_{i} = v_{i}(m+1:m+p)
4: for $i = 1, 2, \ldots, k$ do
5: $r_{t+1} = \tilde{v}_{i}(1: m) - \alpha_{i}u_{i}$
6: Update $\mu_{j} \rightarrow \mu_{j+1}, j = 1, \ldots, i$
7: Determine a set of indices $T_{i} \subseteq \{j|1 \leq j \leq i-1\}$
8: for $j \in T_{i}$ do
9: $r_{t+1} = \hat{r}_{i} - (u_{i}^{T}r_{t+1})u_{i}$
10: Reset $\mu_{j+1}$ to $O(\varepsilon)$
11: end for
12: $\beta_{t+1}u_{t+1} = r_{t+1}$
13: $p_{t+1} = \frac{Q_{T}^{T}\left(\alpha_{t+1}\right)}{\beta_{t+1}v_{t}} - \beta_{t+1}v_{t}$
14: Update $\nu_{j} \rightarrow \nu_{j+1}, j = 1, \ldots, i$
15: Determine a set of indices $S_{i} \subseteq \{j|1 \leq j \leq i-1\}$
16: for $j \in S_{i}$ do
17: $p_{t+1} = p_{t+1} - (\nu_{i}^{T}p_{t+1})\nu_{j}$
18: Reset $\nu_{j+1}$ to $O(\varepsilon)$
19: end for
20: $\alpha_{i+1}v_{i+1} = p_{i}$
21: $\tilde{\beta} = (\alpha_{i+1}\beta_{i+1})/\tilde{\alpha}_{i}$
22: $\hat{\alpha}_{i}u_{i+1} = (-1)^{i}v_{i+1}(m+1:m+p) - \tilde{\beta}\hat{u}_{i}$
23: end for

In Algorithm [2] we need to determine two sets $T_{i}$ and $S_{i}$ at each iteration $i = 1, 2, \ldots, k$. The methods of choosing which previous Lanczos vectors to reorthogonalize has been discussed in detail by Simon [23] and Larsen [12], for symmetric Lanczos process and Lanczos bidiagonalization, respectively. They introduced the $\eta$-criterion. We now use the reorthogonalization of $u_{i+1}$ to explain it. At the $i$-th iteration, we only need to reorthogonalize against the vectors where $\mu_{j,i+1}$ is larger than some constant $\varepsilon < \eta < \omega_{0} = \sqrt{\varepsilon/(2k+1)}$. It is sufficient to choose the vectors where $\mu_{j,i+1}$ exceeds $\sqrt{\varepsilon/(2k+1)}$ and their neighbors exceed $\eta$, to be included in the reorthogonalization step, while a few isolated components that exceed-
ing \( \eta \) are quite harmless. Thus the indices sets \( T_i \) and \( S_i \) can be described by the formulas

\[
T_i = \bigcup_{\mu_{j+1} > \eta_k} \{ l | 1 \leq j - r \leq l \leq j + s \leq i - 1, \mu_{i,j+1} > \eta \},
\]

(4.3)

\[
S_i = \bigcup_{\nu_{j+1} > \eta_k} \{ l | 1 \leq j - r \leq l \leq j + s \leq i - 1, \nu_{i,j+1} > \eta \}.
\]

(4.4)

Simon \cite{23} demonstrated that using the \( \eta \)-criterion in partial reorthogonalization could significantly reduce the amount of extra reorthogonalization work. Experimentally he found that \( \eta = \varepsilon^{1/4} \) was the value that minimized the total amount of reorthogonalization work for the symmetric Lanczos process. In Algorithm \cite{2} we choose \( \eta = \varepsilon^{1/4} \) to implement the partial reorthogonalization.

For the JBDPRO algorithm with \( \eta \)-criterion, the orthogonality levels of \( u_i \) and \( \tilde{v}_i \) will be \( O(\eta) \). By using the same method appeared in the proof of Proposition \cite{11} and Proposition \cite{2,22} we can verify that \( D_k = O(\eta) \) and \( C_k = O(\eta) \).

Notice that our partial reorthogonalization algorithm does not reorthogonalize \( \hat{u}_i \), which avoids designing the complicated reorthogonalization step for \( \hat{u}_i \) and saves a big amount of work. Furthermore, the orthogonality of \( \hat{u}_i \) will be at a desired level although we do not reorthogonalize any \( \hat{u}_i \).

**Theorem 4.2** For the k-step JBDPRO algorithm, we have

\[
\| I_k - \hat{U}_k^T \hat{U}_k \| = O(\| \hat{B}_k^{-1} \|^2 \eta).
\]

(4.5)

**Proof.** Since we do not reorthogonalize any \( \hat{u}_i \), which means that \( \hat{C}_k = 0 \), by \cite{11} we have

\[
\hat{B}_k^T \hat{U}_k^T \hat{U}_k \hat{B}_k = (Q_k \hat{V}_k - \hat{F}_k)(Q_k \hat{V}_k - \hat{F}_k),
\]

so

\[
\hat{B}_k^T (I_k - \hat{U}_k^T \hat{U}_k) \hat{B}_k = \hat{B}_k^T \hat{B}_k - (Q_k \hat{V}_k - \hat{F})^T (Q_k \hat{V}_k - \hat{F}_k)
\]

\[
= I_k - PB_k^T B_k P + O((m + p)\varepsilon) - \hat{V}_k^T Q_k^T Q_k \hat{V}_k + \hat{V}_k^T Q_k^T \hat{F}_k + \hat{F}_k^T Q_k \hat{V}_k - \hat{F}_k^T \hat{F}_k
\]

(4.6)

By \cite{11}, we have

\[
\hat{V}_k^T Q_k A \hat{V}_k = [U_{k+1} B_k + D_k] + F_k^T [U_{k+1} (B_k + D_k) + F_k]
\]

(4.7)

where

\[
E_k = D_k^T U_{k+1} U_{k+1} B_k + B_k^T U_{k+1} U_{k+1} D_k + (B_k + D_k)^T U_{k+1} U_{k+1} F_k
\]

\[
+ F_k^T U_{k+1} (B_k + D_k) + D_k^T U_{k+1} U_{k+1} D_k + F_k^T F_k.
\]

Since \( D_k = O(\eta) \), with simple computation we can obtain

\[
\| E_k \| = O(\eta).
\]
Substituting (4.7) into (4.6), we obtain
\[
\tilde{B}_k^T (I_k - U_k^T U_k) \tilde{B}_k = I_k - \tilde{V}_k^T \tilde{V}_k - PB_k^T (I_{k+1} - U_{k+1}^T U_{k+1}) B_k P \\
+ \tilde{V}_k^T Q_k^T \hat{F}_k + \hat{F}_k^T Q_k \hat{V}_k - \hat{F}_k^T \hat{F}_k + O(m + p) \varepsilon + PE_k P.
\]

With simple computation we can obtain
\[
\| \tilde{V}_k^T Q_k^T \hat{F}_k + \hat{F}_k^T Q_k \hat{V}_k - \hat{F}_k^T \hat{F}_k \| = O(\varepsilon),
\]
thus
\[
\tilde{B}_k^T (I_k - U_k^T U_k) \tilde{B}_k = (I_k - \tilde{V}_k^T \tilde{V}_k) - PB_k^T (I_{k+1} - U_{k+1}^T U_{k+1}) B_k P + O(\eta).
\]

Since \( u_i^T u_j \) and \( v_i^T v_j \) are all of order \( O(\eta) \), \( \tilde{v}_i = (-1)^{i-1} v_i \), we have \( \| I_k - \tilde{V}_k^T \tilde{V}_k \| = O(\eta) \) and \( \| I_{k+1} - U_{k+1}^T U_{k+1} \| = O(\eta) \). We finally obtain
\[
\| I_k - U_k^T U_k \| \\
\leq \| \tilde{B}_k^{-1} \|^2 [O(\eta) + (1 + O(\varepsilon))^2 O(\eta)] \\
= O(\| \tilde{B}_k^{-1} \|^2 \eta).
\]

Remember that the orthogonal level of \( \{ \hat{u}_1, \ldots, \hat{u}_k \} \) is defined as
\[
\hat{\omega}_k = \max_{i \leq p, j \leq k} |\hat{u}_i^T \hat{u}_j|,
\]
so we have
\[
\hat{\omega}_k \leq \| I_k - U_k^T U_k \| \leq \sqrt{k(k-1)} \hat{\omega}_k. \tag{4.8}
\]

Let \( \delta = O(m + p) \| \tilde{B}_k^{-1} \| \varepsilon \), since the orthogonal level of \( \hat{v}_i \) is \( O(\eta) = O(\delta^{3/4}) < \sqrt{\delta/(2k + 1)} \), by Theorem 3.3, the relation (3.30) holds as long as \( \hat{\omega}_k \) is below \( \sqrt{\delta/(2k + 1)} \). Combining Theorem 4.2 and the inequality (4.8), the following condition should be satisfied:
\[
\| \tilde{B}_k^{-1} \|^2 \varepsilon^{3/4} \leq \sqrt{\frac{\delta}{2k + 1}};
\]
which leads to
\[
\| \tilde{B}_k^{-1} \|^3 \leq \frac{m + p}{2k + 1} \frac{1}{\sqrt{\varepsilon}}. \tag{4.9}
\]

For the JBDPRO algorithm, the convergence or final accuracy of Ritz values computed from \( \tilde{B}_k \) will not be affected by rounding errors, as long as the growth of \( \| \tilde{B}_k^{-1} \| \) is controlled by (4.9). For the GSVD computation or the linear discrete ill-posed problem with general Tikhonov regularization, there at least one of the matrices \( A \) and \( L \) to be well conditioned, which results to that at least one of \( Q_L \) and \( Q_A \) is well conditioned. If \( Q_L \) is the well conditioned one, we implement JBDPRO of \{A, L\}, which leads \( \tilde{B}_k \) to be a well conditioned matrix; otherwise if \( Q_A \) is the well conditioned one, we implement JBDPRO of \{L, A\}, which also leads \( \tilde{B}_k \) to be a well conditioned matrix. By this modification, we could always make sure that \( \| \tilde{B}_k^{-1} \| \) can not grow too big.
5 Numerical experiments

In this section we provide several numerical examples to illustrate our theory about the properties of the semiorthogonalization strategy and JBDPRO algorithm. The matrices are constructed by ourselves or chosen from the University of Florida Sparse Matrix Collection \(^2\). For the first pair, the matrices \(A\) and \(L\), which are denoted by \(A_1\) and \(L_1\) respectively, are constructed by ourselves. Let \(n = 800\), construct \(C = \text{diag}(c)\), where \(c = (\frac{3n}{2}, \frac{3n}{2} - 1, \ldots, \frac{n}{2} + 1)/2n\). Then let \(s = (\sqrt{1 - c_1^2}, \ldots, \sqrt{1 - c_n^2})\), construct \(S = \text{diag}(s)\). Let \(D\) be the matrix generated by the MATLAB function \(D = \text{gallery('orthog', n, 2)}\), which means that \(D\) is a symmetric orthogonal matrix. Finally let \(A = CD\) and \(L = SD\). By the construction, we know that the generalized singular value of \(\{A, L\}\) is \(\{c_i/s_i\}\) and the corresponding right vector \(g_i\) is the \(i\)-th column of \(D^T\), where \(i = 1, \ldots, n\). For the second pair, \(A\) and \(L\) are the square matrices \(dw2048\) and \(rdb2048\) from electromagnetics problems and computational fluid dynamics problems respectively. For the third pair, \(A\) is the square matrix \(ex31\) from computational fluid dynamics problems, \(L_{ex} = \text{diag}(l)\), where \(l = (3m, 3m - 1, \ldots, 2m + 1)/4000\) and \(m\) is the row number of \(A\). For the fourth pair, \(A\) is the square matrix \(rdb5000\) from computational fluid dynamics problems, \(L = L_1\), which is the discrete approximation of the first order derivative operator.

\[
L_1 = \begin{pmatrix}
1 & -1 \\
1 & -1 \\
\ddots & \ddots \\
1 & -1
\end{pmatrix} \in \mathbb{R}^{(n-1) \times n},
\tag{5.1}
\]

The properties of our test matrices are described in table 1, where \(\text{cond}()\) means the condition number.

| Example | A   | \(m \times n\) | \(\text{cond}(A)\) | \(L_1\) | \(p \times n\) | \(\text{cond}(L)\) |
|---------|-----|-----------------|---------------------|--------|-----------------|---------------------|
| 1       | \(A_n\) | 800 \times 800  | 2.99                | \(L_n\) | 800 \times 800  | 1.46                |
| 2       | \(dw2048\) | 2048 \times 2048 | 5301.50             | \(rdb2048\) | 2048 \times 2048 | 2026.80             |
| 3       | \(ex31\) | 3909 \times 3909 | 1.01 \times 10^6    | \(L_{ex}\) | 3909 \times 3909 | 1.50                |
| 4       | \(rdb5000\) | 5000 \times 5000 | 4304.90              | \(L_1\) | 4999 \times 5000 | 3183.1              |

The numerical experiments are performed on an Intel (R) Core (TM) i7-7700 CPU 3.60GHz with the main memory 8GB using the Matlab R2017a with the machine precision \(\varepsilon = 2.22 \times 10^{-16}\) under the Windows 10 operating system. Remember that our theory bases on the assumption that \(Q\) is strictly orthogonal, thus we explicitly compute the QR factorization of \(\{A, L\}\) and \(Q\) is directly available.

Figure 1 depicts the the variation of \(\|E_k\| = \|I_k - B_k^T B_k - P B_k^T B_k P\|\) and its estimated bound \(\hat{O}(\|m + p\|\varepsilon)\) as the iteration number \(k\) increases from 1 to 200. From the four examples, we find that \(\|m + p\|\varepsilon\), which is a constant number for the \(m \times n\) matrix \(A\) and \(p \times n\) matrix \(L\), is indeed a bound of \(\|E_k\|\). The bound \(\|m + p\|\varepsilon\) is much bigger than the real value because the constant factor \(\|m + p\|\) coming from equalities (5.15) and (5.16) which hold for general dense matrices. For sparse matrices, the factor \(\|m + p\|\) can be replaced by \((s_m + s_p)\), where \(s_m\) and \(s_p\) denote the maximum number of nonzero elements per column of \(A\) and \(L\).
Fig. 1 Error bound of $\|I_k - B_k^T B_k - \hat{P}_k^T \hat{B}_k^T \hat{P}_k\|$: (a) $\{A_c, L_s\}$; (b) $\{dw2048,rdb2048\}$; (c) $\{ex31, L_m\}$; (d) $\{rdb5000, L_1\}$.

respectively. This insures that $\|E_k\|$ can only grow slightly as the matrix dimension $m$ and $p$ become bigger for large sparse matrices $A$ and $L$.

Fig. 2 Orthogonal levels of $u_i$ and $\tilde{v}_i$: (a) $\{ex31, L_m\}$; (b) $\{rdb5000, L_1\}$.

Figure 2 depicts the orthogonal levels of $u_i$ and $\tilde{v}_i$ computed by the JBDPRO algorithm. We use matrix pairs $\{ex31, L_m\}$ and $\{rdb5000, L_1\}$ to illustrate, and the cases of $\{A_c, L_s\}$ and $\{dw2048,rdb2048\}$ are similar and not be shown here. The $\eta$-criterion is used and $\eta = \varepsilon^{3/4} \approx 10^{-12}$. In the first few iteration steps, the orthogonal levels of $u_i$ and $\tilde{v}_i$ grows gradually until they exceed $\eta$, which means the loss of orthogonality due to the influence of rounding errors. Then, the partial reorthogonalization is applied to $u_i$ and $\tilde{v}_i$, making the
orthogonal levels suddenly jumping down, and the reorthogonalization is not used in a few later steps until the orthogonal levels exceed $\eta$ again. The algorithm continues in this way and the orthogonal levels of $u_i$ and $\tilde{v}_i$ fluctuate around $\eta$ as the iteration number $k$ continues increasing.

Figure 3 and 4 depict the orthogonal level of $\hat{u}_i$ computed by the JBDPRO algorithm and its estimated bound appeared in (4.5). We find that that the orthogonal level of $\hat{u}_i$ is mainly controlled by the growth of $\|\hat{B}_k\|$. Notice that we can implement the JBDPRO
algorithm of \( \{L_1, \text{rdb5000}\} \) instead of \( \{\text{rdb5000}, L_1\} \), which avoids \( \| \hat{B}_k^{-1} \| \) from growing too big. Therefore, the orthogonality of \( \hat{u}_i \) will be at a desired level although we do not reorthogonalize any \( \hat{u}_i \).

Finally, we compare the JBDPRO algorithm with the joint bidiagonalization with full reorthogonalization (JBDFO). The JBDFO algorithm uses the full reorthogonalization strategy for \( u_i, \hat{v}_i \) and \( \hat{u}_i \) at each step, and the computed \( U_{k+1}, \hat{V}_k \) and \( \hat{U}_k \) are orthogonalized to machine precision \( \varepsilon \). From Figure 5 we find that \( \| \hat{E}_k \| \) computed by JBDPRO and JBDFO are almost the same. The quantity \( \| \hat{E}_k \| \) does not deviate too far from \( \varepsilon \) as long as \( \| \hat{B}_k^{-1} \| \) does not grow too big, which ensures that the convergence of Ritz values computed from \( \hat{B}_k \) will not be affected by rounding errors and the final accuracy is high enough.

Table 2 compares the running time of 200-step JBDPRO and JBDFO for our four test examples. For each case, we run the JBDPRO or JBDFO 10 times and take the average over all the 10 running time. From the table, we find that the running time of JBDPRO is

\begin{table} [h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Example & A & L & JBDPRO & JBDFO & ratio \\
\hline
1 & & & & & \\
2 & \text{dl2048} & \text{rdb2048} & 1.8634 & 2.3722 & 78.55\% \\
3 & \text{ex31} & \text{Lm} & 5.3573 & 7.2775 & 73.61\% \\
4 & \text{rdb5000} & \text{L1} & 8.3501 & 11.1115 & 75.15\% \\
\hline
\end{tabular}
\caption{Running time comparison (measured in seconds)}
\end{table}
only about 70%–80% of that of JBDFRO. Therefore, the JBDPRO is more efficient than JBDFRO while can computes approximate generalized singular values and vectors with the same accuracy.

6 Conclusion

We have proposed a semiorthogonalization strategy for the JBD process to maintain some level of orthogonality of the Lanczos vectors. Our rounding error analysis builds connections between the JBD process with semiorthogonalization strategy and Lanczos bidiagonalization process with semiorthogonalization strategy. We have proved that if the Lanczos vectors are kept semiorthogonal, the computed $\hat{B}_k$ is the Ritz-Galerkin projection of $Q_L$ on the subspaces $\text{span}(\hat{U}_k)$ and $\text{span}(\hat{V}_k)$ within error $\hat{\delta} = O((m + p) \| \hat{B}_k^{-1} \| \varepsilon)$. Therefore, the convergence of Ritz values computed from $\hat{B}_k$ will not be affected by rounding errors and the final accuracy of computed quantities is high enough as long as $\| \hat{B}_k^{-1} \|$ does not grow too big.

Based on the semiorthogonalization strategy, we have developed the JBDPRO algorithm. The JBDPRO algorithm can keep the Lanczos vectors at a desired level and saves much unnecessary reorthogonalization work compared with the JBDFRO algorithm. Several numerical examples have been used to confirm our theory and algorithm.

References

1. Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
2. 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/
3. Golub G.H., Kahan W.: Calculating the singular values and pseudo-inverse of a matrix. SIAM J. Numer. Anal. 2, 205–224 (1965)
4. Golub, G.H., van Loan, C.F.: Matrix Computations. John Hopkins University Press (2012)
5. Hansen, P.C.: Regularization, GSVD and truncated GSVD. BIT 29, 491–504 (1989)
6. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion. SIAM, Philadelphia (1998)
7. Hansen, P.C.: Discrete Inverse Problems: Insight and Algorithms. SIAM, Philadelphia (2010)
8. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd ed. SIAM, Philadelphia (2002)
9. Jia, Z., Yang, Y.: A joint bidiagonalization based algorithm for large scale linear discrete ill-posed problems in general-form regularization. arXiv preprint arXiv:1807.08319v2 (2018)
10. Kilmer, M.E., Hansen, P.C., Espanol, M.I.: A projection-based approach to general-form Tikhonov regularization. SIAM J. Sci. Comput. 29, 315–330 (2007)
11. Lanczos, C.: An iteration method for the solution of eigenvalue problem of linear differential and integral operators. J. Res. Nat. Bur. 45, 255–282 (1950)
12. R. M. Larsen, R.M.: Lanczos bidiagonalization with partial reorthogonalization. Department of Computer Science, University of Aarhus (1998)
13. Li, H.: A rounding error analysis of the joint bidiagonalization process with applications to the GSVD computation. arXiv preprint arXiv:1912.08505v1 (2019)
14. Meurant, G., Strakos, Z.: The Lanczos and conjugate gradient algorithms in finite precision arithmetic. Acta Numerica. 15, 471–542 (2006)
15. Paige, C.C.: The computation of eigenvalues and eigenvectors of very large sparse matrices. PhD thesis, University of London (1971)
16. Paige, C.C.: Computational variants of the Lanczos method for the eigenproblem. J. Inst. Math. Appl. 10, 373–381 (1972)
17. Paige, C.C.: Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. J. Inst. Math. Appl. 18, 341–349 (1976)
18. Paige, C.C.: Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. Linear Algebra Appl. 34, 235–258 (1980)
19. Paige, C.C., Saunders, M.A.: Towards a generalized singular value decomposition. SIAM J. Numer. Anal. 18, 398-405 (1981)
20. Paige, C.C., Saunders, M.A.: LSQR, an algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Soft. 8, 43–71 (1982)
21. Parlett, B.N.: The Symmetric Eigenvalue Problem. SIAM, Philadelphia (1998)
22. Simon, H.D.: The Lanczos algorithm with partial reorthogonalization. Math. Comput. 42, 115–142 (1984)
23. Simon, H.D.: Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. Linear Algebra Appl. 61, 101–131 (1984)
24. Simon, H.D., Zha, H.: Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. SIAM J. Sci. Comput. 21, 2257–2274 (2000)
25. van Loan, C.F.: Generalizing the singular value decomposition. SIAM J. Numer. Anal. 13, 76–83 (1976)
26. van Loan, C.F.: Computing the CS and generalized singular value decomposition. Numer. Math. 46, 479–491 (1985)
27. Zha, H.: Computing the generalized singular values/vectors of large sparse or structured matrix pairs. Numer. Math. 72, 391–417 (1996)