



Backward error analysis of the Lanczos bidiagonalization with reorthogonalization

Haibo Li ^{a,b,*}, Guangming Tan ^{b,1}, Tong Zhao ^b

^a Computing System Optimization Lab, Huawei Technologies, Beijing, 100094, China

^b Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China

ARTICLE INFO

MSC:
15A23
65F25
65G50

Keywords:

Lanczos bidiagonalization
Rounding error
Reorthogonalization
Backward error analysis
Householder transformation
Singular value decomposition
LSQR

ABSTRACT

The k -step Lanczos bidiagonalization reduces a matrix $A \in \mathbb{R}^{m \times n}$ into a bidiagonal form $B_k \in \mathbb{R}^{(k+1) \times k}$ while generating two orthonormal matrices $U_{k+1} \in \mathbb{R}^{m \times (k+1)}$ and $V_{k+1} \in \mathbb{R}^{n \times (k+1)}$. However, any practical implementation of the algorithm suffers from loss of orthogonality of U_{k+1} and V_{k+1} due to the presence of rounding errors, and several reorthogonalization strategies are proposed to maintain some level of orthogonality. In this paper, we make a backward error analysis of the Lanczos bidiagonalization with reorthogonalization (LBRO) by writing various reorthogonalization strategies in a general form. Our results show that the computed B_k by the k -step LBRO of A with starting vector b is the exact one generated by the k -step Lanczos bidiagonalization of $A + E$ with starting vector $b + \delta_b$ (denoted by $\text{LB}(A + E, b + \delta_b)$), where the 2-norm of perturbation vector/matrix δ_b and E depend on the roundoff unit and orthogonality levels of U_{k+1} and V_{k+1} . The results also show that the 2-norm of $U_{k+1} - \bar{U}_{k+1}$ and $V_{k+1} - \bar{V}_{k+1}$ are controlled by the orthogonality levels of U_{k+1} and V_{k+1} , respectively, where \bar{U}_{k+1} and \bar{V}_{k+1} are the two orthonormal matrices generated by the k -step $\text{LB}(A + E, b + \delta_b)$ in exact arithmetic. Thus, the k -step LBRO is mixed forward-backward stable as long as the orthogonality of U_{k+1} and V_{k+1} are good enough. We use this result to investigate the backward stability of LBRO based SVD computation algorithm and LSQR algorithm. Numerical experiments confirm our results.

1. Introduction

In [1], Golub and Kahan proposed an algorithm for reducing an arbitrary rectangle matrix to upper bidiagonal form, which is the first step for computing the singular value decomposition(SVD) of the given matrix. Later in [2], Paige and Saunders proposed an algorithm which reduces a matrix to lower bidiagonal form iteratively, based on which they proposed the most widely used LSQR algorithm for solving large sparse least squares problems. Although both the upper and lower bidiagonal reductions can be directly achieved by Householder transformations [3], it is advantageous to reduce the matrix to a partial upper or lower bidiagonal form iteratively by using Lanczos-type procedures for large sparse or structured matrices. Such two procedures are proposed in [2], which are called BIDIAG-1 and BIDIAG-2 corresponding to the lower and upper bidiagonal reductions, respectively. For these reasons, the two above procedures are usually called Golub–Kahan–Lanczos bidiagonalizations or simply Lanczos bidiagonalizations, and we focus on the lower Lanczos bidiagonalization in this paper.

Given a matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$. First choose a nonzero $b \in \mathbb{R}^m$ as the starting vector and compute

$$\beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1 \quad (1.1)$$

* Corresponding author at: Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China.

E-mail addresses: haibolee1729@gmail.com (H. Li), tgm@ict.ac.cn (G. Tan), zhaotong@ict.ac.cn (T. Zhao).

¹ This author is supported in part by the National Natural Science Foundation of China under Grant No. 62032023.

<https://doi.org/10.1016/j.cam.2024.116414>

Received 19 October 2022; Received in revised form 30 June 2023

Available online 1 December 2024

0377-0427/© 2024 Elsevier B.V. All rights reserved, including those for text and data mining, AI training, and similar technologies.

such that $\|u_1\| = \|v_1\| = 1$ and $\beta_1, \alpha_1 > 0$, where $\|\cdot\|$ denotes the 2-norm of a vector or matrix. Then for $i = 1, 2, \dots$, the Lanczos bidiagonalization computes vectors u_{i+1} and v_{i+1} of unit length using the recurrences

$$\beta_{i+1}u_{i+1} = Av_i - \alpha_iu_i, \tag{1.2}$$

$$\alpha_{i+1}v_{i+1} = A^T u_{i+1} - \beta_{i+1}v_i, \tag{1.3}$$

where $\alpha_{i+1}, \beta_{i+1} > 0$. If α_{i+1} or β_{i+1} is zero, then the procedure terminates, having found an invariant singular subspace of A , and this is usually called ‘‘lucky terminate’’ [1]. In the paper we suppose that the procedure does not terminate after k steps. In exact arithmetic the Lanczos bidiagonalization computes two group of mutual orthogonal vectors $\{u_1, \dots, u_{k+1}\}$ and $\{v_1, \dots, v_{k+1}\}$ which are called Lanczos vectors, and a lower bidiagonal matrix

$$B_k = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \alpha_k & \\ & & & \beta_{k+1} & \end{pmatrix} \in \mathbb{R}^{(k+1) \times k}.$$

It is proved that $\{u_1, \dots, u_{k+1}\}$ and $\{v_1, \dots, v_{k+1}\}$ are orthonormal bases of Krylov subspaces $\mathcal{K}_{k+1}(AA^T, b)$ and $\mathcal{K}_{k+1}(A^T A, A^T b)$, respectively [4]. The fundamental relations of the k -step Lanczos bidiagonalization can be written in the matrix form

$$U_{k+1}(\beta_1 e_1^{(k+1)}) = b, \tag{1.4}$$

$$AV_k = U_{k+1}B_k, \tag{1.5}$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1}v_{k+1}(e_{k+1}^{(k+1)})^T, \tag{1.6}$$

where $e_i^{(k+1)}$ is the i th column of the identity matrix of order $k + 1$, and $U_{k+1} = (u_1, \dots, u_{k+1})$ and $V_{k+1} = (v_1, \dots, v_{k+1})$ are two orthonormal matrices.

The Lanczos bidiagonalization is widely used for designing efficient algorithms for many type of large sparse or structured matrix problems. Note that B_k is the Ritz–Galerkin projection of A on the subspaces $\mathcal{K}_{k+1}(AA^T, b)$ and $\mathcal{K}_k(A^T A, A^T b)$. Therefore, the extreme singular values and corresponding vectors of A can be well approximated by the SVD of B_k [1,4]. It is shown in [5] that good low-rank approximations of matrices can be directly obtained from the Lanczos bidiagonalization applied to the given matrix without computing any SVD. The LSQR solver, which is most commonly used for large sparse least squares problems of the form $\min_{x \in \mathbb{R}^n} \|Ax - b\|$, is also based on the Lanczos bidiagonalization [2]. Furthermore, for discrete linear ill-posed inverse problems, several Lanczos bidiagonalization based iterative regularization algorithms are developed for solving large sparse problems, where a regularized solution is obtained by a proper early stopping criterion or by a hybrid regularization method; see [6–8] for further discussions.

In finite precision arithmetic, due to the influence of rounding errors, the Lanczos vectors computed by the Lanczos bidiagonalization gradually lose their mutual orthogonality as the iteration progresses [1,4]. This is a typical phenomenon that appears in the Lanczos-type algorithms, which is first observed in the symmetric Lanczos process used for computing some extreme eigenvalues and eigenvectors of a symmetric matrix [9]. The loss of orthogonality of Lanczos vectors will lead to a delay of convergence for approximating eigenvalues and eigenvectors, and sometimes it is also difficult to determine whether some computed approximations are additional copies or genuine close eigenvalues [10–12]. The above properties can be adapted to handle the Lanczos bidiagonalization since the Lanczos bidiagonalization of A with starting vector b can be written as the symmetric Lanczos process of $\begin{pmatrix} O & A \\ A^T & O \end{pmatrix}$ with starting vector $\begin{pmatrix} b \\ 0 \end{pmatrix}$ [4,13]. On the other hand, when using the LSQR to solve least squares problems, the loss of orthogonality may cause the algorithm to require much more iterations to converge; the finite precision behavior of LSQR is very similar to the closely related conjugate gradient (CG) algorithm based on symmetric Lanczos process, and we refer to [14–16]. For discrete linear inverse problems, the Lanczos bidiagonalization based regularization algorithms also suffers from the delay of convergence of regularized solutions, which can make the propagation of noise during iterations rather irregular [17]. For these reasons, the Lanczos bidiagonalization is often performed with reorthogonalization for solving least squares problems and discrete linear inverse problems. There are several reorthogonalization strategies proposed to maintain some level of orthogonality, such as partial reorthogonalization [4] and one-sided reorthogonalization [5].

It is well known that algorithms based on a sequence of Householder transformations have very good stability properties [18], and these properties have been used to show that the Householder (upper) bidiagonal reduction is mixed forward–backward stable [19, Theorem A2]. For Lanczos bidiagonalization with a reorthogonalization strategy, however, very little is known about the numerical stability of it. As far as we know, the only one result is about the one-sided reorthogonalization, which states that the process applied to a matrix C in finite precision arithmetic produces Krylov subspaces generated by a nearby matrix $C + E_1$, where E_1 is a perturbation matrix [20]. In this paper, we write those various types of reorthogonalization strategies in a general form, where at each iteration u_k and v_k are reorthogonalized against some previous vectors among $\{u_1, \dots, u_{k-1}\}$ and $\{v_1, \dots, v_{k-1}\}$, respectively. Note that some vectors may not be included, which means that they are not used in the reorthogonalization step. We analyze numerical stability of the Lanczos bidiagonalization with reorthogonalization (LBRO) by using this form. This is a key feature of this paper.

In this paper, we give a backward error analysis of the LBRO and generalize the result of [20]. Denote the roundoff unit by u . We first establish a relationship between the LBRO and Householder transformation based bidiagonal reduction. Based on this

result, we show that for the k -step LBRO of A with starting vector b (denoted by $\text{LBRO}(A, b)$): (1). the computed B_k is the exact one generated by the k -step $\text{LB}(A + E, b + \delta_b)$, where the perturbation quantity $\|\delta_b\|/\|b\| = O(\mathbf{u})$ and $\|E\|/\|A\|$ is controlled by \mathbf{u} and orthogonality levels of U_{k+1} and V_{k+1} ; (2). if we denote the two orthonormal matrices generated by $\text{LB}(A + E, b + \delta_b)$ in exact arithmetic by \tilde{U}_{k+1} and \tilde{V}_{k+1} , respectively, then $\|U_{k+1} - \tilde{U}_{k+1}\|$ and $\|V_{k+1} - \tilde{V}_{k+1}\|$ are controlled by the orthogonality levels of U_{k+1} and V_{k+1} , respectively. Compared with [20, Theorem 5.2] that can only deal with the n -step procedure with one-sided reorthogonalization, our result can apply to the k -step LBRO for $1 \leq k \leq n$. Following Higham [18, §1.5], our result implies that the k -step LBRO is mixed forward–backward stable as long as the orthogonality of U_{k+1} and V_{k+1} are good enough. We then use this result to investigate backward stability of LBRO based algorithms including SVD computation and LSQR.

The paper is organized as follows. In Section 2, we review reorthogonalization strategies for the Lanczos bidiagonalization and give some properties. In Section 3, we first establish a relationship between the LBRO and Householder transformation based bidiagonal reduction, then we derive mixed backward–forward error bounds for the LBRO. In Section 4, our result is applied to discuss backward stability of LBRO based SVD computation algorithm and LSQR. In Section 5, we use some numerical examples to illustrate the results. Finally, we conclude the paper in Section 6.

Throughout the paper, we denote by I_k and $O_{k \times l}$ the identity matrix of order k and zero matrix of order $k \times l$, respectively, by $e_i^{(k)}$ the i th column of I_k and by 0_l the zero vector of dimension l . The notation $\|\cdot\|$ always denotes either the vector or matrix 2-norm.

2. The Lanczos bidiagonalization and reorthogonalization strategies

In this section, we review the Lanczos bidiagonalization in finite precision arithmetic and reorthogonalization strategies. From now on, quantities α_i, u_i, B_k , etc. denote the computed ones in finite precision arithmetic. Several types of reorthogonalization strategies have been proposed for maintaining some level of orthogonality of Lanczos vectors [4,5,20], all of which can be written in the following form.

Suppose that at the i th step, the bidiagonalization procedure have computed

$$\beta'_{i+1} u'_{i+1} = Av_i - \alpha_i u_i - f'_i, \tag{2.1}$$

where u'_{i+1} and β'_{i+1} are two temporary quantities and f'_i is a rounding error term. A reorthogonalization strategy applied to u'_{i+1} means that we choose i real numbers $\xi_{1i}, \dots, \xi_{ii}$ and form

$$\beta_{i+1} u_{i+1} = \beta'_{i+1} u'_{i+1} - \sum_{j=1}^i \xi_{ji} u_j - f''_i, \tag{2.2}$$

where f''_i is a rounding error term. Then the algorithm will be continued with u_{i+1} and β_{i+1} instead of u'_{i+1} and β'_{i+1} . The reorthogonalization step of u_{i+1} aims to maintain some level of orthogonality among u_{i+1} and u_j for $j = 1, \dots, i$. In (2.2), the choice of coefficients $\xi_{1i}, \dots, \xi_{ii}$ varies from different types of reorthogonalization strategies, and some values of the coefficients may be zero, which means that the corresponding Lanczos vectors are not used in the reorthogonalization step.

Combining (2.1) and (2.2), the reorthogonalization step of u_{i+1} can be written as the recurrence

$$Av_i = \alpha_i u_i + \beta_{i+1} u_{i+1} + \sum_{j=1}^i \xi_{ji} u_j + f_i, \tag{2.3}$$

where $f_i = f'_i + f''_i$ is the rounding error term. The reorthogonalization step of v_i is similar to that of u_{i+1} and can be written as the recurrence

$$A^T u_i = \alpha_i v_i + \beta_i v_{i-1} + \sum_{j=1}^{i-1} \eta_{ji} v_j + g_i, \tag{2.4}$$

where g_i is the rounding error term. Note that for $i = 1$ reorthogonalization of v_1 is not needed and v_0 is a zero vector.

Remark 2.1. For matrix $A \in \mathbb{R}^{m \times n}$ with $m = n$, the Lanczos bidiagonalization must terminate at $k = n$ in exact arithmetic. In finite precision arithmetic, however, β_{n+1} is usually nonzero. In this case, the computation of u_{n+1} does not make any sense since $U_{n+1} = (U_n, u_{n+1}) \in \mathbb{R}^{n \times (n+1)}$ has deficient column rank, and thus the LBRO should not reorthogonalize u_{n+1} if $m = n$. In practice, the Lanczos bidiagonalization is usually performed in $k \ll n$ steps.

Now we state a set of properties and assumptions on the finite precision behaviors of the Lanczos bidiagonalization, which are from the results of rigorous analysis of the symmetric Lanczos process and Lanczos bidiagonalization [4,5,11,21,22]. They constitute a model for the actual computations and include essential features while discard irrelevant ones. First, by (2.3) and (2.4) the Lanczos bidiagonalization with reorthogonalization can be written in matrix form

$$AV_k = U_{k+1}(B_k + C_k) + F_k, \tag{2.5}$$

$$A^T U_{k+1} = V_k(B_k^T + D_k) + \alpha_{k+1} v_{k+1} (e_{k+1}^{(k+1)})^T + G_{k+1}, \tag{2.6}$$

where

$$C_k = \begin{pmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1k} \\ 0 & \xi_{22} & \cdots & \xi_{2k} \\ & 0 & \ddots & \vdots \\ & & \ddots & \xi_{kk} \\ & & & 0 \end{pmatrix} \in \mathbb{R}^{(k+1) \times k}, \quad D_k = \begin{pmatrix} 0 & \eta_{12} & \eta_{13} & \cdots & \eta_{1k} & \eta_{1k+1} \\ & 0 & \eta_{23} & \eta_{24} & \cdots & \eta_{2k+1} \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & \ddots & \eta_{k-1,k} & \eta_{k-1,k+1} \\ & & & & 0 & \eta_{k,k+1} \end{pmatrix} \in \mathbb{R}^{k \times (k+1)},$$

and $F_k = (f_1, \dots, f_k)$ and $G_{k+1} = (g_1, \dots, g_{k+1})$, satisfying $\|F_k\|, \|G_{k+1}\| = O(\|A\|\mathbf{u})$ [21,22]. Note that the rounding error terms in (2.1) and (2.2) are also satisfied $\|f'_i\|, \|f''_i\| = O(\|A\|\mathbf{u})$ [22]. Then, we always assume that the computed Lanczos vectors are of unit length in order to simplify rounding error analysis. Finally, we assume that

no α_i and β_i ever become negligible,

which is almost always true in practice, and the rare cases where α_i or β_i do become very small are actually the lucky ones, since then the algorithm should be terminated, having found an invariant singular subspace [22].

Following [20,23], we define the orthogonality level of Lanczos vectors as follows.

Definition 2.1. Let $\text{SUT}(\cdot)$ denotes the strictly upper triangular part of a matrix. The orthogonality level of Lanczos vectors $\{u_1, \dots, u_k\}$ or the corresponding matrix $U_k = (u_1, \dots, u_k)$ is

$$\mu_k = \|\text{SUT}(I_k - U_k^T U_k)\|, \tag{2.7}$$

while the orthogonality level of $\{v_1, \dots, v_k\}$ or $V_k = (v_1, \dots, v_k)$ is

$$\nu_k = \|\text{SUT}(I_k - V_k^T V_k)\|. \tag{2.8}$$

By the Cauchy’s interlacing theorem for singular values, it can be verified that $\mu_i \leq \mu_{i+1}$ and $\nu_i \leq \nu_{i+1}$. Let $\sigma_i(\cdot)$ denote the i th largest singular value of a matrix. It is shown in [20] that

$$\sigma_1(V_k) \leq 1 + \nu_k, \tag{2.9}$$

and

$$\sigma_k(V_k) \geq (1 - 2\nu_k)^{1/2} = 1 - \nu_k + O(\nu_k^2) \tag{2.10}$$

if $\nu_k < 1/2$. Similarly, for U_k we have

$$\sigma_1(U_k) \leq 1 + \mu_k, \tag{2.11}$$

and

$$\sigma_k(U_k) \geq (1 - 2\mu_k)^{1/2} = 1 - \mu_k + O(\mu_k^2) \tag{2.12}$$

if $\mu_k < 1/2$. In the rest of the paper, we always assume $\mu_i < 1/2$ and $\nu_i < 1/2$.

Rewrite (2.3) as

$$Av_k = U_{k+1}\tilde{c}_k + f_k, \tag{2.13}$$

where $\tilde{c}_k = (\xi_{1k}, \dots, \xi_{k-1,k}, \alpha_k + \xi_{kk}, \beta_{k+1})^T$. Then using (2.12) we obtain

$$\begin{aligned} \|\tilde{c}_k\| &= \|U_{k+1}^\dagger (Av_k - f_k)\| \leq \sigma_{k+1}(U_{k+1})^{-1} (\|A\| + \|f_k\|) \\ &\leq [1 + \mu_{k+1} + O(\mu_{k+1}^2)] [\|A\| + O(\|A\|\mathbf{u})] \\ &= \|A\| + O(\|A\|(\mathbf{u} + \mu_{k+1})), \end{aligned} \tag{2.14}$$

where we neglect high order terms of \mathbf{u} and μ_{k+1} . Thus, by (2.14) we have

$$\beta_{k+1} \leq \|A\| + O(\|A\|(\mathbf{u} + \mu_{k+1})). \tag{2.15}$$

Similarly, if we rewrite (2.4) as

$$A^T u_k = V_k \tilde{d}_k + g_k, \tag{2.16}$$

where $\tilde{d}_k = (\eta_{1k}, \dots, \eta_{k-2,k}, \beta_k + \eta_{k-1,k}, \alpha_k)^T$, then we can obtain

$$\|\tilde{d}_k\| \leq \|A\| + O(\|A\|(\mathbf{u} + \nu_k)) \tag{2.17}$$

and thus

$$\alpha_k \leq \|A\| + O(\|A\|(\mathbf{u} + \nu_k)). \tag{2.18}$$

The next result is about an upper bound on the coefficients of reorthogonalization step of u_{k+1} , which will be used later in the backward error analysis.

Proposition 2.1. Let $\bar{c}_k = (\xi_{1k}, \dots, \xi_{kk})^T$. Then

$$\|\bar{c}_k\| = O(\|A\|(\mathbf{u} + \mu_{k+1} + \nu_k)). \tag{2.19}$$

Proof. At the k th step, recurrence (2.3) can be written as

$$U_k \bar{c}_k = Av_k - \alpha_k u_k - \beta_{k+1} u_{k+1} - f_k. \tag{2.20}$$

Therefore, we get

$$U_k^T U_k \bar{c}_k = U_k^T Av_k - \alpha_k U_k^T u_k - \beta_{k+1} U_k^T u_{k+1} - U_k^T f_k,$$

which can also be written as

$$U_k^T U_k \bar{c}_k = (B_{k-1}^T + D_{k-1})^T V_{k-1}^T v_k - \alpha_k (U_k^T u_k - e_k^{(k)}) - \beta_{k+1} U_k^T u_{k+1} + G_k^T v_k - U_k^T f_k, \tag{2.21}$$

where we use the relation $U_k^T Av_k = [V_{k-1}^T (B_{k-1}^T + D_{k-1}) + \alpha_k v_k (e_k^{(k)})^T + G_k]^T v_k$ derived from (2.6).

Now we give an upper bound on $\|B_{k-1}^T + D_{k-1}\|$. Using (2.6) we have

$$V_{k-1}^T V_{k-1} (B_{k-1}^T + D_{k-1}) = V_{k-1}^T A^T U_k - \alpha_k V_{k-1}^T v_k (e_k^{(k)})^T - V_{k-1}^T G_k,$$

which leads to

$$\begin{aligned} \|B_{k-1}^T + D_{k-1}\| &\leq \|(V_{k-1}^T V_{k-1})^{-1}\| \cdot \|V_{k-1}^T A^T U_k - \alpha_k V_{k-1}^T v_k e_k^T - V_{k-1}^T G_k\| \\ &\leq (1 - 2\nu_{k-1})^{-1} [\|A\|(1 + \nu_{k-1})(1 + \mu_k) + \alpha_k \nu_k + (1 + \nu_{k-1})\|G_k\|] \\ &= [1 + 2\nu_{k-1} + O(\nu_{k-1}^2)] \cdot [\|A\|(1 + \nu_{k-1} + \mu_k + \nu_{k-1}\mu_k) + \\ &\quad (\|A\| + O(\|A\|(\mathbf{u} + \nu_k))\nu_k + (1 + \nu_{k-1})O(\|A\|\mathbf{u})) \\ &= \|A\| + O(\|A\|(\mathbf{u} + \nu_k + \mu_k)) \end{aligned} \tag{2.22}$$

by neglecting high order terms, where we use the inequality

$$\|(V_{k-1}^T V_{k-1})^{-1}\| = (\sigma_{k-1}(V_{k-1}))^{-2} \leq (1 - 2\nu_{k-1})^{-1} = 1 + 2\nu_{k-1} + O(\nu_{k-1}^2)$$

derived from (2.10).

By using upper bounds on α_k and β_{k+1} in (2.15) and (2.18), we get

$$\begin{aligned} &\| -\alpha_k (U_k^T u_k - e_k^{(k)}) - \beta_{k+1} U_k^T u_{k+1} + G_k^T v_k - U_k^T f_k \| \\ &\leq \alpha_k \mu_k + \beta_{k+1} \mu_{k+1} + (1 + 1 + \mu_k) O(\|A\|\mathbf{u}) \\ &= O(\|A\|(\mathbf{u} + \mu_{k+1})). \end{aligned}$$

Using the inequality derived from (2.12)

$$\|(U_k^T U_k)^{-1}\| = (\sigma_k(U_k))^{-2} \leq (1 - 2\mu_k)^{-1} = 1 + 2\mu_k + O(\mu_k^2)$$

and neglecting high order terms, we finally obtain from (2.21) that

$$\begin{aligned} \|\bar{c}_k\| &\leq \|(U_k^T U_k)^{-1}\| \cdot [\|B_{k-1}^T + D_{k-1}\| \|V_{k-1}^T v_k\| + O(\|A\|(\mathbf{u} + \mu_{k+1}))] \\ &\leq [1 + 2\mu_k + O(\mu_k^2)] \cdot [\|A\| + O(\|A\|(\mathbf{u} + \nu_k + \mu_{k-1}))\nu_k + O(\|A\|(\mathbf{u} + \mu_{k+1}))] \\ &= O(\|A\|(\mathbf{u} + \mu_{k+1} + \nu_k)), \end{aligned}$$

which is the desired result. \square

At the end of this section, we briefly investigate the orthogonality level between two contiguous Lanczos vectors. It is shown from [11] that the property of local orthogonality holds for Lanczos vectors of symmetric Lanczos process. In fact, this property also applies to the Lanczos bidiagonalization (without reorthogonalization; see (2.1)), which can be written in the following form:

$$\beta'_{i+1} |u_i^T u'_{i+1}| = O(c_1(m, n) \|A\|\mathbf{u}), \tag{2.23}$$

where $c_1(m, n)$ is a moderate constant depending on m and n [11,22]. The property of local orthogonality for v_i is similar and we omit it. For this reason, in some literature it is proposed that u_i is not needed in the reorthogonalization step of u_{i+1} ; see e.g. [22,24]. This case corresponds to choose $\xi_{ii} = 0$ in (2.2). In fact, the orthogonality among u_i and u_{i+1} will not be bad as long as the orthogonality between u_{i+1} and $\{u_1, \dots, u_{i-1}\}$ is in a desired level. If we use

$$\omega_{i+1} = \max_{1 \leq j \leq i-1} |u_j^T u_{i+1}| \tag{2.24}$$

to measure the orthogonality level between u_{i+1} and $\{u_1, \dots, u_{i-1}\}$, then we have the following result.

Proposition 2.2. Suppose that u_i is not used in the reorthogonalization step of u_{i+1} . If $\mu_i \ll i^{-1}$, Then we have

$$\beta_{i+1} |u_i^T u_{i+1}| = O(c_1(m, n) \|A\|\mathbf{u}) + O(\|A\|(\nu_i \mu_i + \mu_i^2 + \omega_{i+1} \mu_i)). \tag{2.25}$$

Proof. Since $\xi_{ii} = 0$, from (2.2) we obtain

$$\beta_{i+1} u_i^T u_{i+1} = \beta'_{i+1} u_i^T u'_{i+1} - \sum_{j=1}^{i-1} \xi_{ji} u_i^T u_j - u_i^T f''_i.$$

Using (2.23), it follows that

$$\beta_{i+1} |u_i^T u_{i+1}| \leq O(c_1(m, n) \|A\| \mathbf{u}) + \max_{1 \leq j \leq i-1} |\xi_{ji}| \cdot i \mu_i + O(\|A\| \mathbf{u}). \tag{2.26}$$

In order to get the desired result we need to find an upper bound on $M = \max_{1 \leq j \leq i-1} |\xi_{ji}|$. Again from (2.2), after some rearranging we obtain

$$\xi_{lj} = \beta'_{i+1} u_i^T u'_{i+1} - \beta_{i+1} u_i^T u_{i+1} - \sum_{j=1, j \neq l}^{i-1} (u_i^T u_j) \xi_{ji} - u_i^T f''_i$$

for $l = 1, \dots, i - 1$. Premultiplying (2.1) by U_i^T , we have

$$\begin{aligned} \beta'_{i+1} U_i^T u'_{i+1} &= U_i^T A v_i - \alpha_i U_i^T u_i - U_i^T f'_i \\ &= (B_{i-1} + D_{i-1}^T) V_{i-1}^T v_i + e_i^{(i)} \alpha_i v_i^T v_i + G_i^T v_i - \alpha_i U_i^T u_i - U_i^T f'_i \\ &= (B_{i-1} + D_{i-1}^T) V_{i-1}^T v_i - \alpha_i (U_i^T u_i - e_i^{(i)}) + G_i^T v_i - U_i^T f'_i. \end{aligned}$$

By using (2.22), after some calculations, we have

$$\beta'_{i+1} |u_i^T u'_{i+1}| \leq \beta'_{i+1} \|U_i^T u'_{i+1}\| = O(\|A\|(\mathbf{u} + \mu_i + v_i)),$$

and thus

$$|\xi_{li}| \leq O(\|A\|(\mathbf{u} + \mu_i + v_i)) + \beta_{i+1} \max_{1 \leq j \leq i-1} |u_j^T u_{i+1}| + i \mu_{i-1} M + O(\|A\| \mathbf{u}).$$

Now the right-hand side does not depend on l anymore, and we obtain by taking the maximum on the left side that

$$\begin{aligned} (1 - i \mu_{i-1}) M &\leq O(\|A\|(\mathbf{u} + \mu_i + v_i)) + \beta_{i+1} \max_{1 \leq j \leq i-1} |u_j^T u_{i+1}| + O(\|A\| \mathbf{u}) \\ &= O(\|A\|(\mathbf{u} + \mu_i + v_i + \omega_{i+1})) \end{aligned}$$

Since $\mu_{i-1} \ll i^{-1}$ we obtain

$$M = O(\|A\|(\mathbf{u} + \mu_i + v_i + \omega_{i+1})).$$

By (2.26) we finally obtain the desired bound. \square

For semiorthogonalization strategy and partial reorthogonalization [4,22,24], the orthogonality levels of U_i and V_i are kept below $O(\sqrt{\mathbf{u}})$, thus at the k th step we have $v_k, \mu_k = O(\sqrt{\mathbf{u}})$. It follows from (2.25) that $\beta_{i+1} |u_i^T u_{i+1}| = O(c_1(m, n) \|A\| \mathbf{u})$ as long as we keep $\omega_{k+1} = O(\sqrt{\mathbf{u}})$. Thus the property of local orthogonality still holds for the semiorthogonalization strategy.

3. Backward error analysis of the LBRO

In this section, we first establish a relationship between the LBRO and Householder transformation based bidiagonal reduction. Then we give a backward error analysis of the LBRO to show the mixed forward-backward stability property.

3.1. Connections with Householder bidiagonal reduction

Before giving the results in finite precision arithmetic, we first show a connection between the Lanczos bidiagonalization and Householder QR factorization in exact arithmetic. It is shown in [25] that the modified Gram-Schmidt(MGS) procedure for the QR factorization of a matrix $C \in \mathbb{R}^{r \times l}$ with $r \geq l$ can be interpreted as the Householder QR factorization applied to the augmented matrix $\bar{C} = \begin{pmatrix} O_{l \times l} \\ C \end{pmatrix}$, which is not only true in exact arithmetic, but also in the presence of rounding errors as well. To see this equivalence, let $q_1, \dots, q_l \in \mathbb{R}^l$ be vectors obtained by applying the MGS procedure to C , and the corresponding compact QR factorization of C is $C = QR$ where $Q = (q_1, \dots, q_l)$. Then the Householder QR factorization of \bar{C} is

$$(W_l \dots W_1) \bar{C} = \begin{pmatrix} R \\ O_{r \times l} \end{pmatrix}, \tag{3.1}$$

where W_j are Householder matrices:

$$W_j = I_{r+l} - w_j w_j^T, \quad w_j = \begin{pmatrix} -e_j^{(l)} \\ q_j \end{pmatrix} \in \mathbb{R}^{r+l}.$$

Thus the k th step Householder transformation $W_k \dots W_1 \bar{C}$ is identical to applying k steps MGS procedure to C [25].

For the LBRO a similar property holds. We prove the result for $m > n$, while we discuss the case of $m = n$ in the remark. In the subsequent part of the paper, in order to avoid notation confusions, all exact quantities are denoted by appending $\hat{\cdot}$ to the corresponding finite precision quantities (e.g. $\hat{\alpha}_i$ is exact and α_i includes rounding errors).

Proposition 3.1. For the k -step Lanczos bidiagonalization in exact arithmetic, let

$$\hat{P}_i = I_{m+n+1} - \hat{p}_i \hat{p}_i^T, \quad \hat{p}_i = \begin{pmatrix} -e_i^{(n+1)} \\ \hat{u}_i \end{pmatrix} \in \mathbb{R}^{m+n+1}. \tag{3.2}$$

Then we have

$$\begin{pmatrix} O_{(n+1) \times k} \\ A\hat{V}_k \end{pmatrix} = \hat{P}_1 \cdots \hat{P}_{k+1} \begin{pmatrix} \hat{B}_k \\ O_{s \times k} \end{pmatrix}, \quad s = m + n - k. \tag{3.3}$$

Proof. After performing the procedure in exact arithmetic n steps (if the procedure terminates at some step, we can choose a new starting vector and continue on), we have

$$(b, A\hat{V}_n) = \hat{U}_{n+1} \begin{pmatrix} \hat{\beta}_1 & \hat{\alpha}_1 & & & \\ & \hat{\beta}_2 & \ddots & & \\ & & \ddots & \hat{\alpha}_n & \\ & & & & \hat{\beta}_{n+1} \end{pmatrix},$$

which is the QR factorization of $(b, AV_n) \in \mathbb{R}^{m \times (n+1)}$. Therefore, by (3.1) we have

$$\begin{pmatrix} O_{n+1} & O_{(n+1) \times n} \\ b & A\hat{V}_n \end{pmatrix} = \hat{P}_1 \cdots \hat{P}_{n+1} \begin{pmatrix} \hat{\beta}_1 e_1^{(n+1)} & \hat{B}_n \\ 0_m & O_{m \times n} \end{pmatrix}.$$

Equating from the second to $(k + 1)$ -th column of the above matrix yields

$$\begin{pmatrix} O_{(n+1) \times k} \\ AV_k \end{pmatrix} = (\hat{P}_1 \cdots \hat{P}_{k+1})(\hat{P}_{k+2} \cdots \hat{P}_{n+1}) \begin{pmatrix} B_k \\ O_{s \times k} \end{pmatrix} = \hat{P}_1 \cdots \hat{P}_{k+1} \begin{pmatrix} B_k \\ O_{s \times k} \end{pmatrix},$$

which is the desired result. \square

Remark 3.1. If $m = n$, we have $(b, AV_n) \in \mathbb{R}^{n \times (n+1)}$, and the property (3.1) cannot be directly used. In fact, the procedure must terminate at step n , and thus $\beta_{n+1} = 0$ and $u_{n+1} = 0$. In this case, the form of (3.3) should be rewritten after some adjustments. Let \hat{B}_n be the $n \times n$ lower bidiagonal form by discarding $\hat{\beta}_{n+1}$ and let $\hat{p}_i = \begin{pmatrix} -e_i^{(n)} \\ \hat{u}_i \end{pmatrix} \in \mathbb{R}^{m+n}$ for $i = 1, \dots, n$ and $P_{n+1} = I_{m+n}$. Then we can verify that

$$\begin{pmatrix} O_{n \times k} \\ A\hat{V}_k \end{pmatrix} = \hat{P}_1 \cdots \hat{P}_{k+1} \begin{pmatrix} \hat{B}_k \\ O_{s \times k} \end{pmatrix},$$

where $s = m + n - k - 1$ for $k = 1, \dots, n - 1$ and $s = m$ for $k = n$.

Now we give a corresponding version of Proposition 3.1 in finite precision arithmetic. Similar to the above, we prove the result for $m > n$ and discuss the case of $m = n$ in the remark. The following lemma is needed, which is a generalization of [20, Lemma 4.4] for one-sided reorthogonalization for upper Lanczos bidiagonalization.

Lemma 3.1. For the k th step LBRO, define the orthogonal matrix \hat{P}_{k+1} that is a product of Householder transformations as

$$\hat{P}_{k+1} = P_1 \cdots P_{k+1}, \quad P_i = I_{m+n+1} - p_i p_i^T, \quad p_i = \begin{pmatrix} -e_i^{(n+1)} \\ u_i \end{pmatrix} \in \mathbb{R}^{m+n+1}, \tag{3.4}$$

then we have

$$\hat{P}_{k+1} \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ AV_k \end{pmatrix} + x_k, \quad s = m + n - k, \tag{3.5}$$

where $x_k \in \mathbb{R}^{m+n+1}$ and

$$\|x_k\| = O(\|A\|(\kappa u + \kappa v_k + \mu_{k+1})). \tag{3.6}$$

Proof. By the definition of P_i and together with (2.20) we have

$$\begin{aligned} P_{k+1} \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} &= \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} - p_{k+1}^T \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} P_{k+1} \\ &= \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} + \beta_{k+1} \begin{pmatrix} -e_{k+1}^{(n+1)} \\ u_{k+1} \end{pmatrix} \\ &= \begin{pmatrix} \alpha_k e_k^{(n+1)} \\ \beta_{k+1} u_{k+1} \end{pmatrix} \end{aligned}$$

$$= \begin{pmatrix} \alpha_k e_k^{(n+1)} \\ Av_k - \alpha_k u_k \end{pmatrix} - \begin{pmatrix} 0_{n+1} \\ U_k \bar{c}_k \end{pmatrix} - \begin{pmatrix} 0_{n+1} \\ f_k \end{pmatrix}.$$

We also have

$$\begin{aligned} P_k \begin{pmatrix} \alpha_k e_k^{(n+1)} \\ Av_k - \alpha_k u_k \end{pmatrix} &= P_k \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} - \alpha_k P_k \begin{pmatrix} -e_k^{(n+1)} \\ u_k \end{pmatrix} \\ &= \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} - (u_k^T Av_k) p_k + \alpha_k p_k \\ &= \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} - u_k^T (\alpha_k u_k + \beta_{k+1} u_{k+1} + U_k \bar{c}_k + f_k) p_k + \alpha_k p_k \\ &= \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} - (\beta_{k+1} u_k^T u_{k+1} + u_k^T U_k \bar{c}_k + (u_k^T f_k) p_k), \end{aligned}$$

and

$$P_k \begin{pmatrix} 0_{n+1} \\ U_k \bar{c}_k \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ U_k \bar{c}_k \end{pmatrix} - u_k^T U_k \bar{c}_k p_k, \quad P_k \begin{pmatrix} 0_{n+1} \\ f_k \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ f_k \end{pmatrix} - (u_k^T f_k) p_k.$$

Therefore, we obtain

$$P_k P_{k+1} \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} - \beta_{k+1} u_k^T u_{k+1} p_k - \begin{pmatrix} 0_{n+1} \\ U_k \bar{c}_k \end{pmatrix} - \begin{pmatrix} 0_{n+1} \\ f_k \end{pmatrix}. \tag{3.7}$$

Let $\bar{w}_k = -\beta_{k+1} u_k^T u_{k+1} p_k - \begin{pmatrix} 0_{n+1} \\ U_k \bar{c}_k \end{pmatrix} - \begin{pmatrix} 0_{n+1} \\ f_k \end{pmatrix}$. Notice that $\|p_k\| = \sqrt{2}$. Using Proposition 2.1 and the upper bound on β_{k+1} in (2.15), we have

$$\|\bar{w}_k\| = O(\|A\|(\mathbf{u} + \mu_{k+1} + \nu_k)).$$

For $i = 1, \dots, k - 1$, we have

$$P_i \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} - (u_i^T Av_k) p_i.$$

Using (2.17) we get

$$\begin{aligned} \|(u_i^T Av_k) p_i\| &\leq \sqrt{2} \|v_k^T (V_i \bar{d}_i + g_i)\| = \sqrt{2} \|(v_k^T V_i) \bar{d}_i + v_k^T g_i\| \\ &\leq \sqrt{2} [\nu_k (\|A\| + O(\|A\|(\mathbf{u} + \nu_i))) + O(\|A\|\mathbf{u})] \\ &= O(\|A\|(\mathbf{u} + \nu_k)), \end{aligned}$$

and we further have

$$P_i \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} + \bar{w}_i, \quad i = 1, \dots, k - 1, \tag{3.8}$$

where $\bar{w}_i = -(u_i^T Av_k) p_i$ and $\|\bar{w}_i\| = O(\|A\|(\mathbf{u} + \nu_k))$.

Therefore, by (3.7) and (3.8), we obtain

$$\hat{P}_{k+1} \begin{pmatrix} \alpha_k e_k^{(k)} \\ \beta_{k+1} e_1^{(s+1)} \end{pmatrix} = P_1 \cdots P_{k-1} \left(\begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} + \bar{w}_k \right) = \begin{pmatrix} 0_{n+1} \\ Av_k \end{pmatrix} + x_k,$$

where $x_k = \sum_{i=1}^k (P_1 \cdots P_{i-1}) \bar{w}_i$. Notice that P_i are Householder matrices and thus $\|P_i\| = 1$. We finally obtain

$$\|x_k\| \leq \sum_{i=1}^k \|\bar{w}_i\| = O(\|A\|(k\mathbf{u} + k\nu_k + \mu_{k+1})),$$

which is the desired result. \square

Remark 3.2. For the one-sided reorthogonalization we have $\bar{c}_k = 0$, which does not hold for a general reorthogonalization strategy. This makes the proof of [20, Lemma 4.4] cannot be applied to Lemma 3.1. The upper bound on \bar{c}_k plays a key role in our proof.

The following theorem is a naturally corollary of Lemma 3.1, which generalizes [20, Theorem 4.1]. The proof is similar and we omit it.

Theorem 3.1. For the k -step LBRO, we have

$$\begin{pmatrix} O_{(n+1) \times k} \\ AV_k \end{pmatrix} + X_k = \hat{P}_{k+1} \begin{pmatrix} B_k \\ O_{s \times k} \end{pmatrix}, \quad s = m + n - k, \tag{3.9}$$

where $X_k = (x_1, \dots, x_k)$ and

$$\|X_k\| = O(\|A\| \sqrt{k} (k\mathbf{u} + k\nu_k + \mu_{k+1})). \tag{3.10}$$

(1). There exist a matrix $\tilde{X} \in \mathbb{R}^{(m+n+1) \times n}$ such that

$$\begin{pmatrix} O_{(n+1) \times n} \\ A\tilde{V} \end{pmatrix} + \tilde{X} = \tilde{P} \begin{pmatrix} \tilde{B} \\ O_{m \times n} \end{pmatrix} \tag{3.12}$$

and

$$\|\tilde{X}\| = O(\|A\| \sqrt{n(k\mathbf{u} + kv_{k+1} + \mu_{k+1})}), \tag{3.13}$$

where

$$\tilde{P} = \tilde{P}_{k+1} \tilde{P}_{k+2} \cdots \tilde{P}_{n+1}, \quad \tilde{P}_i = I_{m+n+1} - \tilde{p}_i \tilde{p}_i^T, \quad \tilde{p}_i = \begin{pmatrix} -e_i^{(n+1)} \\ \tilde{u}_i \end{pmatrix}. \tag{3.14}$$

(2). If \tilde{P} is partitioned as

$$\tilde{P} = \begin{pmatrix} n+1 & m \\ \tilde{P}_{11} & \tilde{P}_{12} \\ \tilde{P}_{21} & \tilde{P}_{22} \end{pmatrix} \begin{matrix} n+1 \\ m \end{matrix}, \tag{3.15}$$

then we have

$$\tilde{P}_{21} = \tilde{U}(I_{n+1} - \tilde{P}_{11}), \quad \|\tilde{P}_{11}\| \leq 2\mu_{k+1}. \tag{3.16}$$

Proof (1). We use the following procedure to construct vectors $\tilde{u}_{k+2}, \dots, \tilde{u}_{n+1}$ and $\tilde{v}_{k+2}, \dots, \tilde{v}_n$. After k steps of the LBRO, we have computed $B_k, \alpha_{k+1}, U_{k+1}$ and V_{k+1} . At step $i \geq k+1$, vector \tilde{u}_{i+1} is generated as

$$r_i = A\tilde{v}_i - \tilde{\alpha}_i \tilde{u}_i, \quad \tilde{\beta}_{i+1} \tilde{u}_{i+1} = r_i - \sum_{j=1}^{k+1} (u_j^T r_i) u_j - \sum_{j=k+2}^i (\tilde{u}_j^T r_i) \tilde{u}_j \tag{3.17}$$

such that $\|\tilde{u}_{i+1}\| = 1$, where for $i = k+1$ we let $\tilde{u}_{k+1} = u_{k+1}$ and $\tilde{v}_{k+1} = v_{k+1}$. This procedure is equivalent to the one step Lanczos bidiagonalization of \tilde{u}_{i+1} with full reorthogonalization. The construction of \tilde{v}_{i+1} is similar:

$$s_i = A^T \tilde{u}_{i+1} - \tilde{\beta}_{i+1} \tilde{v}_i, \quad \tilde{\alpha}_{i+1} \tilde{v}_{i+1} = s_i - \sum_{j=1}^{k+1} (v_j^T s_i) v_j - \sum_{j=k+2}^i (\tilde{v}_j^T s_i) \tilde{v}_j, \tag{3.18}$$

which is identical to the one step Lanczos bidiagonalization of \tilde{v}_{i+1} with full reorthogonalization. If the procedure terminates at some step, it can be continued by choosing a new starting vector. Note that the above procedure can be treated as that of first performing the k -step LBRO in finite precision arithmetic then performing the Lanczos bidiagonalization with full reorthogonalization in exact arithmetic for $i \geq k+1$.

By the above construction, we know that $\|\tilde{u}_i\| = \|\tilde{v}_i\| = 1$ and \tilde{u}_i is orthogonal to U_{k+1} and \tilde{u}_j while \tilde{v}_i is orthogonal to V_{k+1} and \tilde{v}_j for $i \neq j, i, j > k+1$, respectively. Then we can carry out the same method as that in the proof of Lemma 3.1. For the i th step procedure with $i \geq k+1$, we can prove that $\tilde{w}_i = O(\|A\|(\mathbf{u} + v_{k+1} + \mu_{k+1}))$, and $\tilde{w}_j = 0$ with $j = 1, \dots, i-1$ for $i > k+1$ and $\tilde{w}_j = O(\|A\|(\mathbf{u} + v_{k+1}))$ with $j = 1, \dots, i-1$ for $i = k+1$. This can be checked easily following the proof method of Lemma 3.1. Therefore, (3.5) in Lemma 3.1 becomes

$$\tilde{P}_{k+1} \tilde{P}_{k+2} \cdots \tilde{P}_{i+1} \begin{pmatrix} \tilde{\alpha}_i e_i^{(i)} \\ \tilde{\beta}_{i+1} e_1^{(s+1)} \end{pmatrix} = \begin{pmatrix} 0_{n+1} \\ A\tilde{v}_i \end{pmatrix} + \tilde{x}_i, \quad s = m + n - i \tag{3.19}$$

with $\|\tilde{x}_i\| = O(\|A\|(\mathbf{u} + kv_{k+1} + \mu_{k+1}))$ for $i = k+1$ or $\|\tilde{x}_i\| = O(\|A\|(\mathbf{u} + v_{k+1} + \mu_{k+1}))$ for $i > k+1$. Note that for $i = k+1$ we let $\tilde{v}_{k+1} = v_{k+1}$ and $\tilde{\alpha}_{k+1} = \alpha_{k+1}$. Therefore, equality (3.9) in Theorem 3.1 becomes (3.12), where $\tilde{X} = (x_1, \dots, x_k, \tilde{x}_{k+1}, \dots, \tilde{x}_n)$ and thus

$$\|\tilde{X}\| \leq \sqrt{n} \max_{\substack{1 \leq i \leq k \\ k+1 \leq j \leq n}} \{\|x_i\|, \|\tilde{x}_j\|\} = O(\|A\| \sqrt{n(k\mathbf{u} + kv_{k+1} + \mu_{k+1})}).$$

(2). Let $M = \text{SUT}(\tilde{U}^T \tilde{U})$. By Theorem 3.2, we have

$$\tilde{P}_{11} = (I_{n+1} + M)^{-1} M, \quad \tilde{P}_{21} = \tilde{U}(I_{n+1} - \tilde{P}_{11}).$$

By inequality (3.11) we obtain

$$\|\tilde{P}_{11}\| \leq 2\mu_{k+1},$$

since the orthogonality level of \tilde{U} is $\|M\| = \mu_{k+1}$. \square

Now we are ready to give a mixed forward-backward error bound of the k -step LBRO. We only prove the result for $m > n$, for the case of $m = n$ the result can be proved similarly after some adjustments.

Theorem 3.3. For the k -step LBRO, there exist two orthonormal matrices $\bar{U}_{k+1} = (\bar{u}_1, \dots, \bar{u}_{k+1}) \in \mathbb{R}^{m \times (k+1)}$ and $\bar{V}_{k+1} = (\bar{v}_1, \dots, \bar{v}_{k+1}) \in \mathbb{R}^{n \times k}$ such that

$$\bar{U}_{k+1}(\beta_1 e_1^{(k+1)}) = b + \delta_b, \tag{3.20}$$

$$(A + E)\bar{V}_k = \bar{U}_{k+1}B_k, \tag{3.21}$$

$$(A + E)^T \bar{U}_{k+1} = \bar{V}_k B_k^T + \alpha_{k+1} \bar{v}_{k+1} (e_{k+1}^{(k+1)})^T, \tag{3.22}$$

and

$$\|\bar{U}_{k+1} - U_{k+1}\| \leq 2\mu_{k+1} + O(\mu_{k+1}^2), \quad \|\bar{V}_{k+1} - V_{k+1}\| \leq \nu_{k+1} + O(\nu_{k+1}^2), \tag{3.23}$$

where E and δ_b are perturbation matrix and vector, respectively, satisfying

$$\|E\| = O(\|A\| \sqrt{n(k\mathbf{u} + k\nu_{k+1} + \mu_{k+1})}), \quad \|\delta_b\| = O(\|b\|\mathbf{u}). \tag{3.24}$$

Proof. In finite precision arithmetic, we have $b + \delta_0 = \beta_1 u_1$, where $\|\delta_0\| = O(\|b\|\mathbf{u})$. By the definition of \tilde{P} , we have

$$\begin{pmatrix} 0_{n+1} \\ b \end{pmatrix} + \delta_1 = P_1 \begin{pmatrix} \beta_1 \\ 0_{m+n} \end{pmatrix} = \tilde{P} \begin{pmatrix} \beta_1 \\ 0_{m+n} \end{pmatrix}, \quad \delta_1 = \begin{pmatrix} 0_{n+1} \\ \delta_0 \end{pmatrix}.$$

Combining with (3.12), we have

$$\begin{pmatrix} 0_{n+1} & O_{(n+1) \times n} \\ b & A\tilde{V} \end{pmatrix} + \begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{pmatrix} = \begin{pmatrix} \tilde{P}_{11} & \tilde{P}_{12} \\ \tilde{P}_{21} & \tilde{P}_{22} \end{pmatrix} \begin{pmatrix} \tilde{B} \\ O_{m \times n} \end{pmatrix} = \begin{pmatrix} \tilde{P}_{11} \tilde{B} \\ \tilde{P}_{21} \tilde{B} \end{pmatrix} \tag{3.25}$$

where

$$\begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{pmatrix} = (\delta_0, \tilde{X}), \quad \tilde{B} = (\beta_1 e_{n+1}^{(n+1)}, \tilde{B}).$$

Since \tilde{P} is orthogonal, we have

$$\tilde{P}_{11}^T \tilde{P}_{11} + \tilde{P}_{21}^T \tilde{P}_{21} = I_{n+1}. \tag{3.26}$$

Now we construct two matrices \bar{U}_{n+1} and \bar{V}_n .

Let the compact SVD of \tilde{P}_{21} is $\tilde{P}_{21} = Y_1 \Sigma Z^T$, where $Y = (Y_1, Y_2) \in \mathbb{R}^{m \times m}$ and $Z \in \mathbb{R}^{(n+1) \times (n+1)}$ are orthogonal matrices and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{n+1})$ satisfying $0 \leq \sigma_{n+1} \leq \dots \leq \sigma_1 \leq 1$. Then by (3.26) we have

$$Z^T \tilde{P}_{11}^T \tilde{P}_{11} Z = (I_{n+1} + \Sigma)(I_{n+1} - \Sigma).$$

Define $\bar{U}_{n+1} = Y_1 Z^T \in \mathbb{R}^{m \times (n+1)}$. Then \bar{U}_{n+1} is an orthonormal matrix, and by the above equality we have

$$\bar{U}_{n+1} - \tilde{P}_{21} = Y_1 (I_{n+1} - \Sigma) Z^T = Y_1 (I_{n+1} + \Sigma)^{-1} Z^T \tilde{P}_{11}^T \tilde{P}_{11}. \tag{3.27}$$

By (3.16) and (3.27) we obtain

$$\begin{aligned} \|\bar{U}_{n+1} - \tilde{U}\| &= \|\bar{U}_{n+1} - \tilde{P}_{21} - \tilde{U} \tilde{P}_{11}\| \\ &\leq \|\bar{U}_{n+1} - \tilde{P}_{21}\| + \|\tilde{U} \tilde{P}_{11}\| \\ &\leq \|\tilde{P}_{11}\|^2 + \|\tilde{P}_{11}\| (1 + \mu_{k+1}) \\ &\leq 4\mu_{k+1}^2 + 2\mu_{k+1} (1 + \mu_{k+1}) \\ &= 2\mu_{k+1} + O(\mu_{k+1}^2). \end{aligned} \tag{3.28}$$

By (3.25) and (3.27) we obtain

$$\bar{U}_{n+1} \tilde{B} - (b, A\tilde{V}) = \bar{U}_{n+1} \tilde{B} - (\tilde{P}_{21} \tilde{B} - \tilde{X}_2) = (\bar{U}_{n+1} - \tilde{P}_{21}) \tilde{B} + \tilde{X}_2$$

and

$$(\bar{U}_{n+1} - \tilde{P}_{21}) \tilde{B} = Y_1 (I_{n+1} + \Sigma)^{-1} Z^T \tilde{P}_{11}^T \tilde{P}_{11} \tilde{B} = Y_1 (I_{n+1} + \Sigma)^{-1} Z^T \tilde{P}_{11}^T \tilde{X}_1,$$

which can be rewritten as

$$\bar{U}_{n+1} \tilde{B} - (b, A\tilde{V}) = (C, I_m) \begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_1 \end{pmatrix} = (C, I_m) (\delta_0, \tilde{X}), \tag{3.29}$$

where $C = Y_1 (I_{n+1} + \Sigma)^{-1} Z^T \tilde{P}_{11}^T$. Define

$$\delta_b = \beta_1 \bar{u}_1 - b, \quad E_1 = \bar{U}_{n+1} \tilde{B} - A\tilde{V}. \tag{3.30}$$

Then (3.29) implies that

$$\delta_b = (C, I_m) \delta_0, \quad E_1 = (C, I_m) \tilde{X}.$$

Notice that

$$CC^T + I_m = I_m + Y_1(I_{n+1} - \Sigma)(I_{n+1} + \Sigma)^{-1}Y_1^T,$$

and thus $\|(C, I_m)\| \leq \sqrt{2}$. Therefore,

$$\|\delta_b\| \leq \|(C, I_m)\| \|\delta_0\| \leq \sqrt{2} \|\delta_0\| = O(\|b\|u),$$

$$\|E_1\| \leq \|(C, I_m)\| \|\tilde{X}\| \leq \sqrt{2} \|\tilde{X}\|.$$

Let the SVD of \tilde{V} is $\tilde{V} = K \Sigma_1 J^T$. Define $\tilde{V}_n = K J^T \in \mathbb{R}^{n \times n}$. then \tilde{V}_n is orthogonal and

$$\|\tilde{V}_n - \tilde{V}\| = \|I_{n+1} - \Sigma_1\| \leq 1 - (1 - 2\nu_{k+1})^{1/2} = \nu_{k+1} + O(\nu_{k+1}^2), \tag{3.31}$$

where we have used $\sigma_n(\tilde{V}) \geq (1 - 2\nu_{k+1})^{1/2}$ since the orthogonality level of \tilde{V} is ν_{k+1} ; see (2.10). By the definition of E_1 in (3.29), we have $A\tilde{V} + E_1 = \tilde{U}_{n+1}\tilde{B}$, which can be rewritten as

$$(A + E)\tilde{V}_n = \tilde{U}_{n+1}\tilde{B} \text{ or } (A + E)^T\tilde{U}_{n+1} = \tilde{V}_n\tilde{B}^T, \tag{3.32}$$

where

$$E = [A(\tilde{V} - \tilde{V}_n) + E_1]\tilde{V}_n^T,$$

and thus

$$\|E\| \leq \|A\|[\nu_{k+1} + O(\nu_{k+1}^2)] + \sqrt{2}\|\tilde{X}\| = O(\|A\|\sqrt{n}(k\mathbf{u} + k\nu_{k+1} + \mu_{k+1})).$$

Let $\tilde{U}_{k+1} = \tilde{U}_{n+1}(:, 1 : k + 1)$ and $\tilde{V}_{k+1} = \tilde{V}_n(:, 1 : k + 1)$. By using $\|\tilde{U}_{k+1} - U_{k+1}\| \leq \|\tilde{U}_{n+1} - \tilde{U}\|$ and $\|\tilde{V}_{k+1} - V_{k+1}\| \leq \|\tilde{V}_n - \tilde{V}\|$ we obtain (3.23). By equating the first k and $k + 1$ columns of the two equalities of (3.32) respectively, we obtain (3.21) and (3.22). Note that $\beta_1 \tilde{u}_1 = \tilde{U}_{k+1}(\beta_1 e_1^{(k+1)})$. By (3.30) we finally obtain (3.20). \square

Theorem 3.3 is a generalization of [20, Theorem 5.2] that deals with upper Lanczos bidiagonalization with one-sided reorthogonalization. There are three main improvements. First, our result can apply to a general reorthogonalization strategy, and by letting $\tilde{c}_k = 0_k$ for $k = 1, 2, \dots$ that corresponds to the one-sided reorthogonalization case we can obtain a similar upper bound on $\|E\|$ as [20, Theorem 5.2]. Second, apart from giving an upper bound on $\|\tilde{V}_{k+1} - V_{k+1}\|$, our result also gives a similar upper bound on $\|\tilde{U}_{k+1} - U_{k+1}\|$. Finally, although [20, Theorem 5.2] is about the k -step process, upper bounds on $\|E\|$ and $\|\tilde{V}_k - V_k\|$ there depend on the orthogonality level of V_n that can only be known after the n -step process has finished, while our result, in contrast, can really apply to the k -step LBRO for $1 \leq k \leq n$ due to the help of Lemma 3.2.

Notice that the relations (3.20)–(3.22) are matrix-form recurrences of the k -step Lanczos bidiagonalization of $A + E$ with starting vector $b + \delta_b$ (denoted by $\text{LB}(A + E, b + \delta_b)$) in exact arithmetic. If the orthogonality levels of U_{k+1} and V_{k+1} are kept around $O(u)$, which corresponds to the full reorthogonalization case, then B_k is the exact one generated by the k -step $\text{LB}(A + E, b + \delta_b)$ with $\|E\|/\|A\|, \|\delta_b\|/\|b\| = O(u)$. Following Higham [18, §1.5], we could call the k -step LBRO *mixed forward-backward stable* as long as the orthogonality of U_{k+1} and V_{k+1} are good enough.² This result can be used to analyze backward stability of LBRO based algorithms, such as bidiagonalization based algorithms for computing a partial SVD or LSQR for iteratively solving least squares problems.

4. Applications to SVD computation and LSQR

In this section, we use the above results to investigate backward stability of LBRO based algorithms including partial SVD computation and LSQR. For simplicity, we only consider rounding errors in the LBRO, while other parts of an algorithm is supposed to be performed in exact arithmetic.

We first review the partial SVD computation of A based on the Lanczos bidiagonalization in exact arithmetic. We add “ $\hat{\cdot}$ ” to α_i, u_i, B_k , etc. to denote the corresponding quantities in exact arithmetic. In order to approximate some singular values and corresponding vectors of A , we first compute the compact SVD of \hat{B}_k :

$$\hat{B}_k = H_k \Theta_k Z_k^T, \quad \Theta_k = \text{diag}(s_1^{(k)}, \dots, s_k^{(k)}), \quad s_1^{(k)} > \dots > s_k^{(k)} > 0, \tag{4.1}$$

where $H_k = (h_1^{(k)}, \dots, h_k^{(k)}) \in \mathbb{R}^{(k+1) \times k}$ and $Z_k = (z_1^{(k)}, \dots, z_k^{(k)}) \in \mathbb{R}^{k \times k}$ are orthonormal, and $\Theta_k \in \mathbb{R}^{k \times k}$. Then the approximate singular values of A are $s_i^{(k)}$, which are often called Ritz values, and the corresponding approximate left and right singular vectors are $x_i^{(k)} = \hat{U}_{k+1} h_i^{(k)}$ and $y_i^{(k)} = \hat{V}_k z_i^{(k)}$, respectively, where $i = 1, \dots, k$. The above decomposition can be achieved by a direct method since \hat{B}_k is a matrix of small scale, and it is appropriate to compute some extreme singular values and vectors [4].

In finite precision arithmetic, the above procedure suffers from the “ghost” phenomenon, which means that some of the converged Ritz values suddenly “jump” to become a “ghost” and then converge to the next larger or smaller singular values after a few iterations. These redundant copies of Ritz values are usually called “ghosts” and this phenomenon results in many unwanted spurious copies of singular values and make it difficult to determine whether these spurious copies are real multiple singular values. This problem is closely related to the loss of orthogonality of Lanczos vectors, which can be avoided by using some types of

² The mixed forward-backward stability can be illustrated by the following example. Suppose a method is used to compute $y = f(x)$ for $f : \mathbb{R} \rightarrow \mathbb{R}$. Then the method is called mixed forward-backward stable if the computed value \hat{y} satisfies $\hat{y} + \Delta y = f(x + \Delta x)$ where $|\Delta y|/|y|$ and $|\Delta x|/|x|$ are sufficiently small.

reorthogonalization strategies [4,5]. The full reorthogonalization strategy is often used to mimic the Lanczos bidiagonalization in exact arithmetic. Next, we give an analysis of full reorthogonalization for partial SVD computation in finite precision arithmetic.

Suppose the Lanczos bidiagonalization is implemented with full reorthogonalization and the orthogonality levels of U_{k+1} and V_{k+1} are kept around $O(\mathbf{u})$. By Theorem 3.3, $s_i^{(k)}$ are Ritz values of $\tilde{A} = A + E$ and they will converge to the singular values of \tilde{A} . Notice that $\|E\| = O(\|A\|\sqrt{nk}\mathbf{u})$. By the perturbation theory of singular values we have

$$\frac{|\sigma_i(A) - \sigma_i(\tilde{A})|}{\sigma_i(A)} = \frac{|\sigma_i(A) - \sigma_i(\tilde{A})|}{\|A\|} \cdot \frac{\sigma_i(A)}{\sigma_i(A)} \leq \frac{\sigma_i(A)}{\sigma_i(A)} \cdot O(\sqrt{nk}\mathbf{u}) \tag{4.2}$$

for $\sigma_i(A) > 0$. Therefore, the several largest Ritz values $s_i^{(k)}$ will approximate largest singular values of A with relative error around $O(\mathbf{u})$. If $\sigma_i(A)$ has multiplicity bigger than one, then the several converged Ritz values corresponding to $\sigma_i(A)$ are not strictly equal. By (4.2), with the help of full reorthogonalization, errors between these Ritz values will be so small that they all look like approximations to the same $\sigma_i(A)$, but they actually differ by a value of $O(\sqrt{nk}\mathbf{u})$. If the algorithm is implemented in a lower precision arithmetic, then separations between those converged Ritz values corresponding to a singular value with multiplicity greater than one will be observed more obviously. If that singular value denoted by $\sigma_i(A)$ is very small, then the converged Ritz values corresponding to $\sigma_i(A)$ will have bigger relative errors since $\sigma_1(A)/\sigma_i(A)$ is very big. The above discussion is illustrated by a numerical example described by Fig. 3 in Section 5.

Now we investigate the LSQR that is based on the Lanczos bidiagonalization for iteratively solving a large scale linear least squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|. \tag{4.3}$$

In exact arithmetic, the problem is projected to the Krylov subspace $\mathcal{K}_k(A^T A, A^T b) = \text{span}(\hat{V}_k)$ to become a k -dimension small scale problem. Denoting a vector in $\text{span}(\hat{V}_k)$ by $x = \hat{V}_k y$ where $y \in \mathbb{R}^k$, we have

$$\begin{aligned} \min_{x = \hat{V}_k y} \|Ax - b\| &= \min_{y \in \mathbb{R}^k} \|A\hat{V}_k y - \hat{\beta}_1 U_{k+1} e_1^{(k+1)}\| \\ &= \min_{y \in \mathbb{R}^k} \|\hat{U}_{k+1} \hat{B}_k y - \hat{\beta}_1 \hat{U}_{k+1} e_1^{(k+1)}\| \\ &= \min_{y \in \mathbb{R}^k} \|\hat{B}_k y - \hat{\beta}_1 e_1^{(k+1)}\|, \end{aligned}$$

and thus the k th iterative solution of the LSQR is

$$\hat{x}_k = \hat{V}_k \hat{y}_k, \quad \hat{y}_k = \arg \min_{y \in \mathbb{R}^k} \|\hat{B}_k y - \hat{\beta}_1 e_1^{(k+1)}\|. \tag{4.4}$$

In [2], the authors proposed a formula that recursively update \hat{x}_k to \hat{x}_{k+1} very efficiently to avoid solving the k -dimension small-scale problem at each iteration.

In finite precision arithmetic, the convergence of LSQR can be slowed down significantly due to the loss of orthogonality of Lanczos vectors. Maintaining a certain level of orthogonality among Lanczos vectors will accelerate the convergence at the expense of a bit more computational costs and storage requirements.³ For example, the LSQR is often implemented with full reorthogonalization to maintain stability of convergence for solving discrete linear ill-posed problems, since it usually takes not too many iterations to reach the semi-convergence point [6,17].

We analyze the LSQR with full reorthogonalization for solving least squares problems. In finite precision arithmetic, the k -step LSQR computes

$$x_k = V_k y_k, \quad y_k = \arg \min_{y \in \mathbb{R}^k} \|B_k y - \beta_1 e_1^{(k+1)}\|.$$

By (3.20) and (3.21), we have

$$y_k = \arg \min_{y \in \mathbb{R}^k} \|\bar{U}_{k+1} B_k y - \bar{U}_{k+1} \beta_1 e_1^{(k+1)}\| = \arg \min_{y \in \mathbb{R}^k} \|(A + E)\bar{V}_k y - (b + \delta_b)\|.$$

Therefore, we get $\bar{x}_k = \arg \min_{x = \bar{V}_k y} \|(A + E)x - (b + \delta_b)\|$ where $\bar{x}_k = \bar{V}_k y_k$, which implies that \bar{x}_k is the k th LSQR solution in exact arithmetic to the perturbed problem

$$\min_{x \in \mathbb{R}^n} \|(A + E)x - (b + \delta_b)\|. \tag{4.5}$$

Notice from (3.23) that

$$\|\bar{x}_k - x_k\| \leq \|V_k - \bar{V}_k\| \|y_k\| \leq \|y_k\| (v_k + O(v_k^2)).$$

Using $\|y_k\| = \|\bar{x}_k\|$ we obtain

$$\frac{\|x_k - \bar{x}_k\|}{\|\bar{x}_k\|} \leq v_k + O(v_k^2). \tag{4.6}$$

If we keep the orthogonality of V_k around $O(\mathbf{u})$, then $x_k \approx \bar{x}_k$ and thus x_k can be treated as the k th LSQR solution to the perturbed problem (4.5). This indicates the backward stability property of LSQR if full reorthogonalization is used.

³ In full reorthogonalization, u_k and v_k are reorthogonalized against all previous vectors $\{u_1, \dots, u_{k-1}\}$ and $\{v_1, \dots, v_{k-1}\}$ as soon as they have been computed. This adds an arithmetic cost of about $4(m+n)k^2$ flops, which is affordable if $k \ll \min\{m, n\}$.

Table 1
Properties of the test matrices.

| Index | A | $m \times n$ | $\ A\ $ | $\kappa(A)$ | Description |
|-------|----------|--------------|---------|-------------|-----------------------|
| 1 | nos3 | 960 × 960 | 689.904 | 37723.6 | Structural problem |
| 2 | well1850 | 1850 × 712 | 1.79433 | 111.313 | Least squares problem |
| 3 | lshp2614 | 2614 × 2614 | 6.98798 | 5197.35 | Thermal problem |
| 4 | c-23 | 3969 × 3969 | 1089.71 | 22795.9 | Optimization problem |

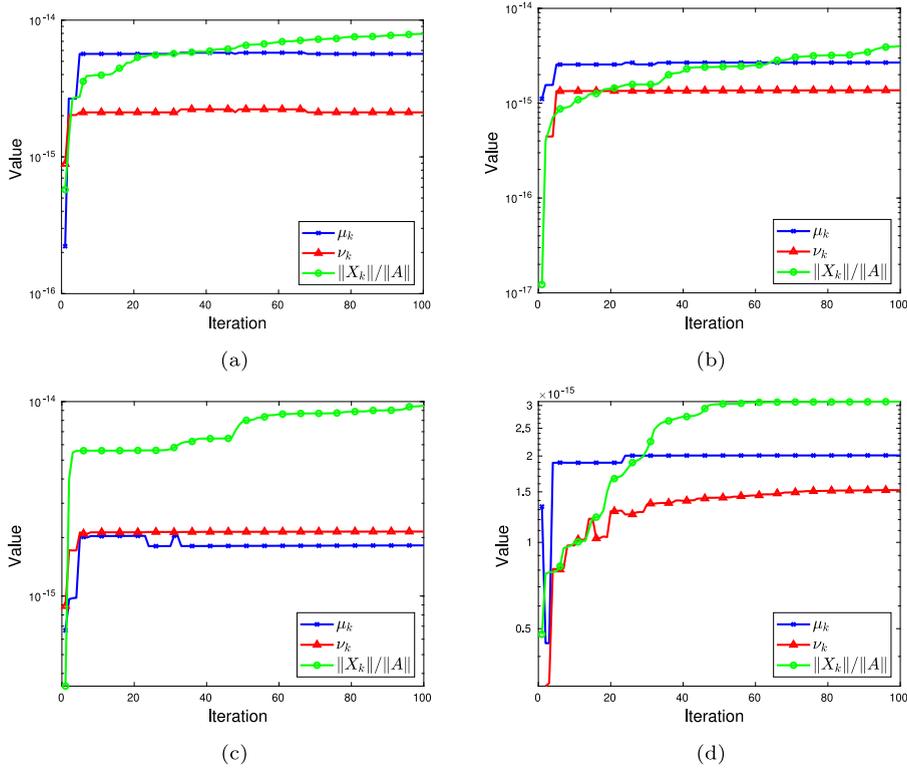


Fig. 1. Values of $\|X_k\|/\|A\|$ for full reorthogonalization: (a) nos3; (b) well1850; (c) lshp2614; (d) c-23.

5. Numerical experiments

In this section, we use several numerical examples to illustrate our results. Since the error matrix $\|E\|$ in [Theorem 3.3](#) cannot be explicitly calculated, we only show the values of $\|X_k\|$ in [Theorem 3.1](#). In order to confirm the upper bound on $\|E\|$ implicitly, we use an example to illustrate the relation [\(4.2\)](#) for partial SVD computations. The following four matrices are taken from [\[26\]](#), and a description of them is in [Table 1](#), where $\kappa(\cdot)$ is the condition number of a matrix. All the computations are carried out in MATLAB R2019b, where the roundoff unit for double precision arithmetic is $u = 2^{-53} \approx 1.11 \times 10^{-16}$. All starting vectors are chosen as $b = (1, \dots, 1) \in \mathbb{R}^m$.

[Fig. 1](#) depicts the values of $\|X_k\|/\|A\|$ and orthogonality levels of U_k and V_k as the iterations progress from 1 to 100, where the LBRO is implemented using full reorthogonalization. For the four matrices, the orthogonality levels of U_k and V_k are maintained around $O(u)$. Therefore, by the upper bound in [\(3.10\)](#), $\|X_k\|/\|A\|$ should be around $O(u)$, which can be observed from the four subfigures.

Then we use partial reorthogonalization to implement the LBRO by using `lanbpro.m` in the PROPACK package [\[4\]](#). [Fig. 2](#) depicts the values of $\|X_k\|/\|A\|$ as well as μ_k and ν_k . We also depict the orthogonality levels of u_k and v_k defined as

$$\omega(u_k) = \max_{1 \leq i < k} |u_i^T u_k|, \quad \omega(v_k) = \max_{1 \leq i < k} |v_i^T v_k|,$$

respectively. Note that $\omega(u_i) \leq \mu_k \leq k \max_{1 \leq i \leq k} \omega(u_i)$ and this quantity is used to measure orthogonality between u_k and previous Lanczos vectors. From the figure we find that the values of μ_k and ν_k grow slowly and eventually stabilize around a value. This is because we set `OPTIONS.eta=1e-10` in `lanbpro.m`, which can keep orthogonality levels below 10^{-10} after reorthogonalization. We can also find that $\|X_k\|/\|A\|$ grows in consistent with μ_k and ν_k , which confirms the upper bound [\(3.10\)](#) again. We find that $\omega(u_k)$ and $\omega(v_k)$ grow gradually at first but jump down when reorthogonalization is used. Then, they stay low for a few steps but

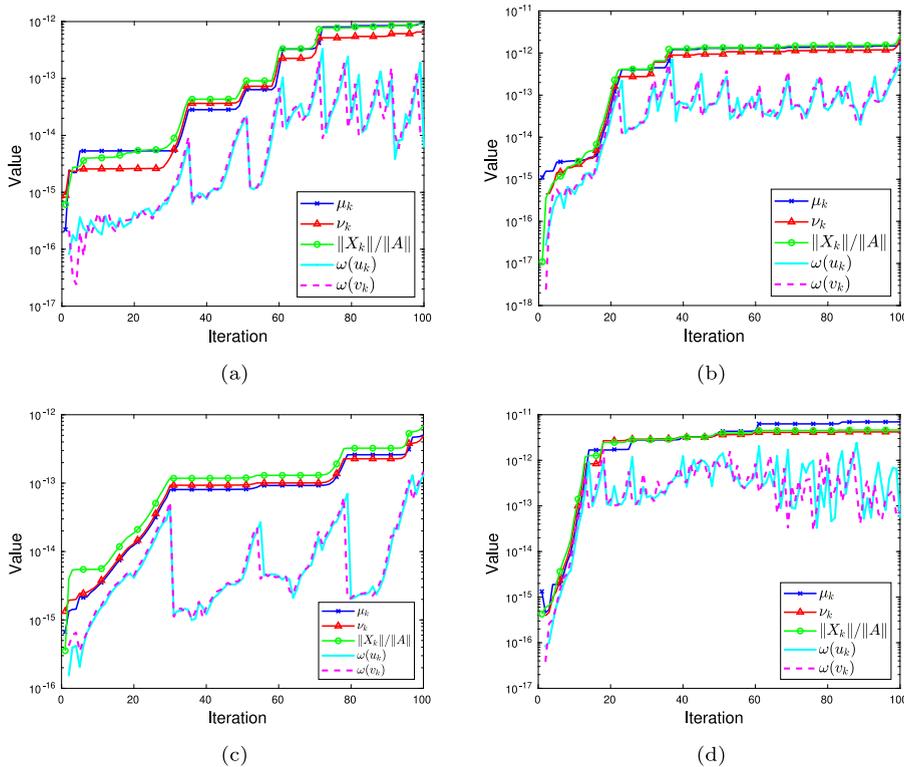


Fig. 2. Values of $\|X_k\|/\|A\|$ for partial reorthogonalization: (a) nos3; (b) well1850; (c) lshp2614; (d) c-23.

increase again until reorthogonalization is used again. This process repeats. This is because partial reorthogonalizations occurs only when necessary and only a part of previous Lanczos vectors are included in the reorthogonalization steps.

Finally we use an example to illustrate the relation (4.2). The matrix A is constructed as follows. Let $m = n = 800$. First construct a row vector $s = (s_1, s_2, \dots, s_n)$. The first two (s_1, s_2) and the last two (s_{n-1}, s_n) are repeated (have multiplicity 2) with values 1.0 and 10^{-4} . The remaining values are as follows: $s_3 = 0.95$, $s_{n-2} = 0.10$ and the rest decrease linearly from $s_4 = 0.90$ to $s_{n-3} = 0.15$ with uniform steps. Let $S = \text{diag}(s)$, and let P and Q be two symmetric orthogonal matrices generated by the MATLAB built-in functions $P = \text{gallery}(\text{orthog}', n, 1)$ and $Q = \text{gallery}(\text{orthog}', n, 2)$, respectively. Finally let $A = PSQ^T$. By the construction, the i th largest singular value of A is $\sigma_i = s_i$, and the multiplicities of singular values σ_1 and σ_n are 2. In the implementations, we first store both A and b in double precision, and then the LBRO using full reorthogonalization is implemented in double and single precision arithmetic, respectively, and all the other computations are taken in double precision.

Fig. 3 depicts the convergence history of the first four largest approximations and the error curve corresponding to the 2-multiplicity singular value σ_1 . The black circles along the right vertical axis in subfigures (3(a)) and (3(c)) indicate the values of σ_i for $i = 1, \dots, n$. The experimental results for smallest approximations are similar and we omit them. Subfigures (3(a)) and (3(c)) show that the multiplicities of σ_1 can be determined correctly from the convergence history of $s_1^{(k)}$ and $s_2^{(k)}$. However, in subfigures (3(b)) and (3(d)) we find that $s_1^{(k)}$ is not rigorously equal to $s_2^{(k)}$ although they are both used to approximate σ_1 . This can be observed more obviously in subfigure (3(d)) where the LBRO is performed in single precision arithmetic with roundoff unit $u = 2^{-24} \approx 5.96 \times 10^{-8}$.

Table 2 shows the errors of approximations to both the 2-multiplicity singular values $\sigma_1 = \sigma_2$ and $\sigma_n = \sigma_{n-1}$. For approximations to σ_1 we set $k = 100$, while the smallest two Ritz values converge to σ_n more slowly and we set $k = 250$ in this case. We can find that both $s_1^{(k)}$ and $s_2^{(k)}$ as well as $s_k^{(k)}$ and $s_{k-1}^{(k)}$ differs by a value of $O(u)$, which is consistent with the upper bound on $\|E\|$. For $\sigma_n = 10^{-4}$, the relative errors of approximations are much bigger than that of $\sigma_1 = 1.0$ due to a large value of σ_1/σ_n , which is consistent with (4.2).

6. Conclusion

By writing various types of reorthogonalization strategies in a general form, we have made a backward error analysis of the LBRO in finite precision arithmetic. For the k -step $\text{LBRO}(A, b)$, we have shown that: (1). the computed B_k is the exact one generated by the k -step $\text{LB}(A + E, b + \delta_b)$, where $\|\delta_b\|/\|b\| = O(u)$ and $\|E\|/\|A\|$ has the same order of magnitude as the sum of u and orthogonality levels of U_{k+1} and V_{k+1} ; (2). if we denote the two orthonormal matrices generated by $\text{LB}(A + E, b + \delta_b)$ in exact arithmetic by \bar{U}_{k+1} and

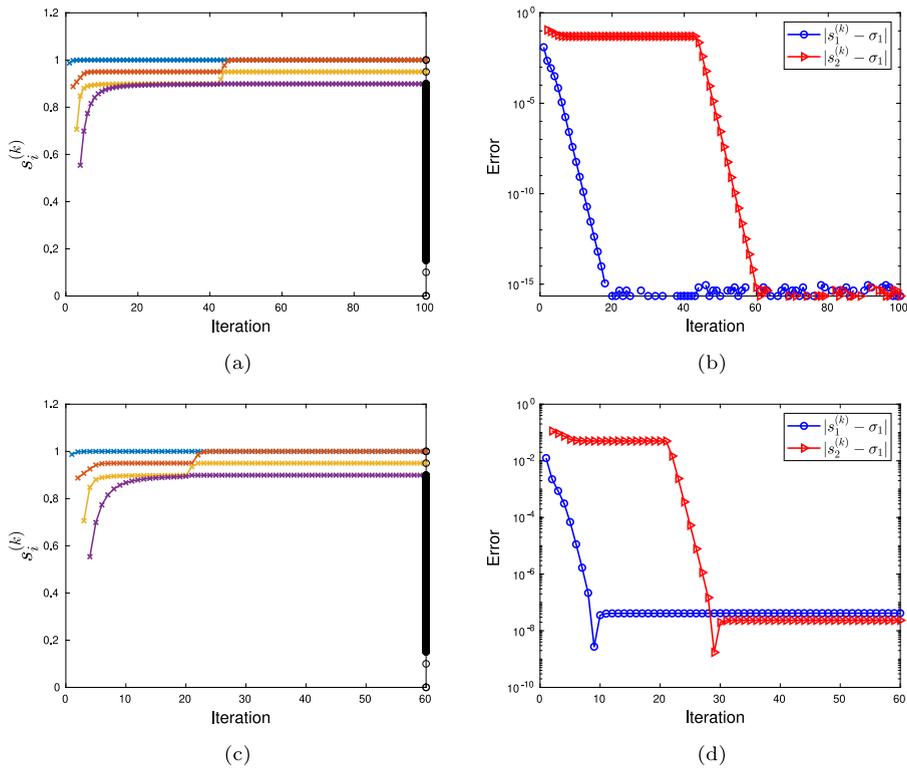


Fig. 3. Convergence and accuracy of approximate largest singular values: (a), (b) full reorthogonalization in double precision arithmetic; (c), (d) full reorthogonalization in single precision arithmetic.

Table 2
Accuracy of approximate singular values computed in double and single precision arithmetic.

| Work precision | $ s_1^{(100)} - \sigma_1 /\sigma_1$ | $ s_2^{(100)} - \sigma_2 /\sigma_2$ | $ s_1^{(100)} - s_2^{(100)} $ |
|----------------|---|---|---------------------------------------|
| double | 2.22×10^{-16} | 2.22×10^{-16} | 4.44×10^{-16} |
| single | 4.17×10^{-8} | 2.33×10^{-8} | 1.83×10^{-8} |
| | $ s_{250}^{(250)} - \sigma_n /\sigma_n$ | $ s_{249}^{(250)} - \sigma_{n-1} /\sigma_{n-1}$ | $ s_{250}^{(250)} - s_{249}^{(250)} $ |
| double | 1.30×10^{-12} | 1.08×10^{-12} | 2.38×10^{-16} |
| single | 7.93×10^{-6} | 3.27×10^{-5} | 2.48×10^{-9} |

\bar{V}_{k+1} , respectively, then the first order errors of $\|U_{k+1} - \bar{U}_{k+1}\|$ and $\|V_{k+1} - \bar{V}_{k+1}\|$ are of the same orders of magnitude as orthogonality levels of U_{k+1} and V_{k+1} , respectively. Thus, the k -step LBRO is mixed forward-backward stable as long as the orthogonality of U_{k+1} and V_{k+1} are good enough. This result is then used to show backward stability of the LBRO based SVD computation algorithm and LSQR algorithm. For the Lanczos bidiagonalization with full reorthogonalization, we have shown that the LBRO based SVD actually computes approximations to the singular values of $A + E$ with $\|E\| = O(\|A\|\mathbf{u})$, and the LBRO based LSQR computes iterative solution that approximately equals to the exact solution of a slightly perturbed least squares problem within error $O(\mathbf{u})$. Several numerical experiments are made to confirm the results. All these results can help us to gain more insights on the finite precision behavior of the Lanczos bidiagonalization algorithm.

Acknowledgments

The authors would like to thank Dr. Long Wang and Prof. Weile Jia for their generous supports and stimulating discussions. We are grateful to the anonymous referees and Editor Prof. Michael Ng for their insightful comments that helped to improve the quality of the paper.

Data availability

Data will be made available on request.

References

- [1] G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, *J. SIAM: Ser. B, Numer. Anal.* 2 (2) (1965) 205–224.
- [2] C.C. Paige, M.A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software* 8 (1982) 43–71.
- [3] G.H. Golub, C.F. Van Loan, *Matrix Computations*, fourth ed., The Johns Hopkins University Press, Baltimore, 2013.
- [4] R.M. Larsen, Lanczos bidiagonalization with partial reorthogonalization, *DAIMI Rep. Ser.* (1998) URL <http://sun.stanford.edu/~rmunk/PROPACK/>.
- [5] H.D. Simon, H. Zha, Low-rank matrix approximation using the lanczos bidiagonalization process with applications, *SIAM J. Sci. Comput.* 21 (6) (2000) 2257–2274.
- [6] Å. Björck, A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations, *BIT* 28 (3) (1988) 659–670.
- [7] P.C. Hansen, *Rank-deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.
- [8] P.C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010.
- [9] C. Lanczos, An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, United States Governm. Press Office, Los Angeles, CA, 1950, pp. 255–282.
- [10] C.C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices* (Ph.D. thesis), University of London, 1971.
- [11] C.C. Paige, Error analysis of the lanczos algorithm for tridiagonalizing a symmetric matrix, *IMA J. Appl. Math.* 18 (3) (1976) 341–349.
- [12] C.C. Paige, Accuracy and effectiveness of the lanczos algorithm for the symmetric eigenproblem, *Linear Algebr. Appl.* 34 (1980) 235–258.
- [13] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [14] A. Greenbaum, Z. Strakoš, Predicting the behavior of finite precision lanczos and conjugate gradient computations, *SIAM J. Matrix Anal. Appl.* 13 (1) (1992) 121–137.
- [15] G. Meurant, Z. Strakoš, The lanczos and conjugate gradient algorithms in finite precision arithmetic, *Acta Numer.* 15 (2006) 471–542.
- [16] A. Greenbaum, H. Liu, T. Chen, On the convergence rate of variants of the conjugate gradient algorithm in finite precision arithmetic, *SIAM J. Sci. Comput.* 43 (5) (2021) S496–S515.
- [17] I. Hnětynková, M. Plešinger, Z. Strakoš, The regularizing effect of the golub-kahan iterative bidiagonalization and revealing the noise level in the data, *BIT* 49 (4) (2009) 669–696, <http://dx.doi.org/10.1007/s10543-009-0239-7>.
- [18] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2002.
- [19] R. Byers, H. Xu, A new scaling for Newton’s iteration for the polar decomposition and its backward stability, *SIAM J. Matrix Anal. Appl.* 30 (2) (2008) 822–843.
- [20] J.L. Barlow, Reorthogonalization for the Golub–Kahan–Lanczos bidiagonal reduction, *Numer. Math.* 124 (2) (2013) 237–278.
- [21] B.N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, 1998.
- [22] H.D. Simon, Analysis of the symmetric lanczos algorithm with reorthogonalization methods, *Linear Algebr. Appl.* 61 (1984) 101–131.
- [23] 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 (2) (2009) 565–583.
- [24] H.D. Simon, The lanczos algorithm with partial reorthogonalization, *Math. Comput.* 42 (165) (1984) 115–142.
- [25] Å. Björck, C.C. Paige, Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm, *SIAM J. Matrix Anal. Appl.* 13 (1) (1992) 176–190.
- [26] T.A. Davis, Y. Hu, The University of Florida sparse matrix collection, *ACM Trans. Math. Softw.* 38 (1) (2011) 1–25, URL <http://www.cise.ufl.edu/research/sparse/matrices/>.