mathematics

MDPI

Article

Roundoff Error Analysis of an Algorithm Based on Householder Bidiagonalization for Total Least Squares Problems †

Zhanshan Yang [1] and Xilan Liu [2,*]

[1] School of Mathematics and Statistics, Qinghai Nationalities University, Xining 810007, China; yangzhsh15@lzu.edu.cn
[2] School of Mathematics and Information Science, Baoji University of Arts and Sciences, Baoji 721000, China
* Correspondence: doclanliu@163.com
† Supported by the NNSF of China (11571004, 11701456), Natural Science Foundation of Qinghai Province (2018-ZJ-717), Foundation Sciences Qinghai Nationalities University (2020XJG11, 2019XJZ10).

Abstract: For large-scale problems, how to establish an algorithm with high accuracy and stability is particularly important. In this paper, the Householder bidiagonalization total least squares (HBITLS) algorithm and nonlinear iterative partial least squares for total least squares (NIPALS-TLS) algorithm were established, by which the same approximate TLS solutions was obtained. In addition, the propagation of the roundoff error for the process of the HBITLS algorithm was analyzed, and the mixed forward-backward stability of these two algorithms was proved. Furthermore, an upper bound of roundoff error was derived, which presents a more detailed and clearer approximation of the computed solution.

Keywords: Householder bidiagonalization; NIPALS; roundoff error; total least squares problems

Citation: Yang, Z.; Liu, X. Roundoff Error Analysis of an Algorithm Based on Householder Bidiagonalization for Total Least Squares Problems. *Mathematics* **2021**, *9*, 2550. https://doi.org/10.3390/math9202550

Academic Editors: Mehdi Salimi, Alicia Cordero Barbero and Christopher Goodrich

Received: 16 August 2021
Accepted: 30 September 2021
Published: 12 October 2021

1. Introduction

Consider estimating $\boldsymbol{x}$ from the overdetermined linear system

$$A\boldsymbol{x} \approx \boldsymbol{b} \quad \text{for} \quad A \in \mathbb{R}^{m\times n}, \quad \boldsymbol{b} \in \mathbb{R}^{m} \quad \text{and} \quad \boldsymbol{x} \in \mathbb{R}^{n}, \tag{1}$$

where the error exists in both the right-hand side b and the data matrix A and $m \geq n+1$. In this case, the total least squares (TLS) model should be appropriate to adopt (cf. [1,2]). The TLS approach is just to find a perturbation with the minimum Frobenius norm to make the system (1) a compatible system

$$\min \ \|(E, \boldsymbol{r})\|_F, \ \text{ subject to } \ \boldsymbol{b} + \boldsymbol{r} \in \text{Range}(A + E). \tag{2}$$

The TLS method is widely used in various scientific fields, such as physics, automatic control, signal processing, statistics, economics, biology, medicine etc. In essence, a solution of a TLS problem can be expressed by a singular value decomposition of the augmented matrix $(A, \boldsymbol{b})$. When the dimensions of A are not too large, one can use the truncated-SVD (TSVD) method. When the dimensions of A become large, this approach becomes prohibitive because the SVD algorithm is of complexity $\mathcal{O}(mn^2)$. The above considerations lead us to consider Krylov iterative methods, that do not alter the matrix A. The methods have the attractive feature just like the Lanczos methods—that when n increases, the computed extreme singular elements rapidly become good approximations to the exact ones, and are satisfactorily accurate even if k is far less than n theoretically [1]. Nevertheless, the orthonormal properties of the Krylov basis strongly support the use of these Householder matrix-based algorithms. This is particularly true when we need to be sure that the perturbed problem we are solving has to conserve some spectral similarity properties. This

will be especially relevant when we need to compute approximations of the TLS problems. In view of this, we consider applying the Householder bidiagonalization algorithm and the NIPALS PLS algorithm posed by Å. Björck [3] to TLS problems, the formed Householder bidiagonalization total least squares (HBITLS) algorithm, and NIPALS-TLS algorithm, respectively. Furthermore, we find that the HBITLS and NIPALS-TLS algorithms also compute the same approximate solutions for the TLS problems.

When it comes to practical problems, the arithmetic will be inaccurate and there will be errors in each step of the calculation. Arithmetic operations running on the computer have finite precision, so there will be rounding errors as long as there are numerical computations. These rounding errors cause the calculation quantities to be different from their theoretical values. One of the design principles of the floating-point operation is that it should encourage experts to develop robust, efficient, and portable numerical programs, enable the handling of arithmetic exceptions, and provide for the development of transcendental functions and high-precision arithmetic [4]. The results in the roundoff error analysis in Lanczos-type methods obtained by Paige [5–7] played an important role in interpreting the behavior of the Lanczos method in finite-precision computations. Parlett and Scott [8] used the results of the roundoff error analysis as the basis for suggesting a modification of the Lanczos method, which they called selective orthogonalization [8–10]. In addition, in many practical problems, the stop criterion can be safely selected on the basis of the rounding error analysis of the original problem, thereby diminishing the need for an extremely precise approximation of the algebraic problem solution [4]. As far as we know, the roundoff error analysis of the approximation TLS solutions obtained by using the Householder bidiagonalization procedure was not systematically performed in the literature. Hence, in this paper, we analyzed the propagation of the roundoff error during the process of the HBITLS algorithm and found that the HBITLS algorithm and NIPALS-TLS algorithm are mixed forward–backward stable.

The paper is organized as follows. The HBITLS algorithm and NIPALS-TLS algorithm were established, by which the same approximate TLS solution was obtained in Section 2. Section 3 analyzes the propagation of the roundoff error during the process of the HBITLS algorithm. A brief conclusion is shown in the last section.

2. HBITLS Algorithm and NIPALS-TLS Algorithm

It is well known that algorithms based on a sequence of orthogonal transformations with Householder matrices have very good stability properties; see Higham [4]. Based on this, this paper gives the HBITLS and NIPALS-TLS algorithms and finds that they both compute the same approximate TLS solutions.

2.1. HBITLS Algorithm

Let us first describe the Householder bidiagonalization process just as shown in [3]. However, in this paper, the process is used in the augmented matrix $(\boldsymbol{b}, A)$ for the TLS problem. The idea is to compute orthogonal matrices $U \in \mathbb{R}^{m\times(n+1)}$ and $V \in \mathbb{R}^{(n+1)\times(n+1)}$, such that

$$U^T(\boldsymbol{b}, A)V = \left(\begin{array}{c|cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \beta_3 & \ddots & \\ & & & \ddots & \alpha_n \\ & & & & \beta_{n+1} \end{array}\right) = (\beta_1 \boldsymbol{e}_1, B_n) \equiv C_n. \tag{3}$$

$U = G_1 \cdots G_n = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_{n+1}]$ and $V = H_1 \cdots H_n = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{n+1}]$ can be determined as a product of Householder matrices in each iteration. Generally, G_k introduces zero in the kth column, while H_k sets zero for the appropriate entries in the kth row. This can be done by an algorithm named Householder. Given the reason of space, and known to all, we omit it here, see algorithm 5.4.2 in [1] for details.

From the above process of Householder bidiagonalization, we know that V can be rewritten as $V = \begin{pmatrix} 1 & 0 \\ 0 & V_{n-1} \end{pmatrix}$, $V_{n-1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n]$.

In detail, from (3), we have

$$U^T\boldsymbol{b} = U^T(\boldsymbol{b}, A)V\boldsymbol{e}_1 = U^T(\boldsymbol{b}, A)\begin{pmatrix} 1 & 0 \\ 0 & V_{n-1} \end{pmatrix}\boldsymbol{e}_1 = C_n\boldsymbol{e}_1 = \beta_1\boldsymbol{e}_1$$

and

$$A\boldsymbol{v}_i = \alpha_i\boldsymbol{u}_i + \beta_{i+1}\boldsymbol{u}_{i+1}, \quad \text{for } i = 1, \cdots, n. \tag{4}$$

From (3), we have

$$(\boldsymbol{b}, A)^T U = \begin{pmatrix} 1 & 0 \\ 0 & V_{n-1} \end{pmatrix} C_n^T$$

and there comes

$$A^T\boldsymbol{u}_1 = \alpha_1\boldsymbol{v}_1,$$

$$A^T\boldsymbol{u}_i = \alpha_i\boldsymbol{v}_i + \beta_i\boldsymbol{v}_{i-1}, \quad \text{for } i = 2, \cdots, n-1. \tag{5}$$

Let $V_k = H_1H_2\cdots H_k\begin{pmatrix} I_k \\ 0 \end{pmatrix}$, $U_k = G_k\cdots G_2G_1\begin{pmatrix} I_k \\ 0 \end{pmatrix}$, and $C_k = (\beta_1\boldsymbol{e}_1, B_k)$ is a leading principal submatrix of order $k+1$ of the final bidiagonal matrix C_n. As we all know, if the exact arithmetic is used, we have $U_{k+1}^TU_{k+1} = I$, $V_k^TV_k = I$. However, in any case, the previous equations remain within machine precision. Then (4) and (5) can be rewritten as

$$(\boldsymbol{b}, A)V_k = U_{k+1}C_k, \tag{6}$$

$$(\boldsymbol{b}, A)^TU_{k+1} = V_kC_k^T + \alpha_{k+1}\boldsymbol{v}_{k+1}\boldsymbol{e}_{k+1}^T = V_{k+1}\bar{C}_k^T, \tag{7}$$

where $\bar{C}_k = (C_k, \alpha_{k+1}\boldsymbol{e}_k) \in \mathbb{R}^{k\times(k+1)}$.

After performing the k-step Householder bidiagonalization iterations, the TLS problem can be reduced onto the subspace generated by U_{k+1} and V_k. Then the reduced TLS problem (also see [11]) is as follows

$$\min \|U_{k+1}^T((\boldsymbol{b}, A) - (\hat{\boldsymbol{b}}_k, \hat{A}_k))\begin{pmatrix} 1 & 0 \\ 0 & V_k \end{pmatrix}\|_F \quad \text{subject to } U_{k+1}^T\hat{A}_kV_k\boldsymbol{y} = U_{k+1}^T\hat{\boldsymbol{b}}_k, \tag{8}$$

or

$$\min \|(\beta_1\boldsymbol{e}_1, B_k) - (\hat{\boldsymbol{e}}_k, \hat{B}_k)\|_F \quad \text{subject to } \hat{B}_k\boldsymbol{y} = \hat{\boldsymbol{e}}_k, \tag{9}$$

where $\boldsymbol{e}_1 = (1, 0, \cdots, 0)^T$, and $\hat{B}_k$ and $\hat{\boldsymbol{e}}_k$ are generally full. As in LSQR, seek an approximate TLS solution

$$\boldsymbol{x}_k = V_k\boldsymbol{y}_k \in \mathcal{K}_k(A^TA, A^T\boldsymbol{b}),$$

where $\mathcal{K}_k(B, \boldsymbol{y})$ denotes the Krylov subspace $\text{span}\{\boldsymbol{y}, B\boldsymbol{y}, ..., B^{k-1}\boldsymbol{y}\}$.

Let the SVD of $(\beta_1\boldsymbol{e}_1, B_k) = \bar{W}_k\bar{\Sigma}_k\bar{Z}_k^T$ and if let

$$\begin{pmatrix} \gamma_k \\ \boldsymbol{z}_k \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & V_k \end{pmatrix}(\bar{Z}_k\boldsymbol{e}_{k+1}) \tag{10}$$

with

$$\bar{Z}_k = \begin{pmatrix} \bar{Z}_1 & \bar{Z}_2 \end{pmatrix} = \begin{array}{ccc} & k & 1 \\ & \bar{Z}_{11} & \bar{z}_{12} \quad 1 \\ & \bar{Z}_{21} & \bar{z}_{22} \quad k \end{array}, \tag{11}$$

then the approximate TLS solution is given by

$$\boldsymbol{x}_k = -\frac{\boldsymbol{z}_k}{\gamma_k} \in \mathcal{K}_k(A^TA, A^T\boldsymbol{b}). \tag{12}$$

Note that we only need the last singular vector $\bar{Z}_k\boldsymbol{e}_{k+1}$ to compute $\boldsymbol{x}_k$. To this extent, summarizing the above process, we can get the Householder bidiagonalization TLS (HBITLS) algorithm as follows:

Remark 1. *A variant of Algorithm 1 can also be given, in which the product of the Householder transformations applying to vectors are replaced by operations that can be performed concurrently, to a large extent. This variation gives an efficient method for developing parallelism in the case of parallel computing matrix vector products. In regard to this variation of Algorithm 1, one can refer to [12] and we omit it here.*

Algorithm 1 HBITLS

1: Initialize: $C = (\boldsymbol{b}, A)$, $U := I$, $V := I$.
2: for $j = 1, \cdots, n+1$
$[\boldsymbol{v}, \beta] = \text{Householder}(C(j:m, j))$;
$P = I_{m-j+1} - \beta\boldsymbol{v}\boldsymbol{v}^T$;
$C(j:m, j:n+1) = PC(j:m, j:n+1)$;
$G = \text{diag}\{I_{j-1}, P\}$;
$U = UG$;
$\boldsymbol{u}_j = U(:, j)$; if $j \leq n-1$
$[\boldsymbol{v}, \beta] = \text{Householder}(C(j, j+1:n+1))$;
$Q = I_{n-j} - \beta\boldsymbol{v}\boldsymbol{v}^T$;
$C(j:m, j+1:n+1) = C(j:m, j+1:n+1)Q$;
$H = \text{diag}\{I_j, Q\}$;
$V = VH$;
$\boldsymbol{v}_j = V(:, j)$.
end
3: Compute the last singular triplet for matrix C_k by employing the implicit zero-shift QR algorithm.
4: the approximate TLS solution is given by $\boldsymbol{x}_k = -\frac{\boldsymbol{z}_k}{\gamma_k}$, see (10).
5: end

Using the recursions (4) and (5), the following properties of $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i\}$ and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_i\}$ can be proved.

Lemma 1. *The sets $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i\}$ and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_i\}$ generated by Algorithm 1 are the orthonormal basis of $\mathcal{K}_i(A^TA, A^T\boldsymbol{b})$ and $\mathcal{K}_i(AA^T, AA^T\boldsymbol{b})$ respectively.*

Proof. As a result of the facts that $\beta_1\boldsymbol{v}_1 = A^T\boldsymbol{b}$, $\alpha_1\boldsymbol{u}_1 = A\boldsymbol{v}_1 = AA^T\boldsymbol{b}/\beta_1$, $H_1\boldsymbol{v}_1 = \boldsymbol{e}_1$ and the process of Householder bidiagonalization, for $1 \leq k \leq i$, it's easy to know that $\boldsymbol{v}_j = H_1 \cdots H_k\boldsymbol{e}_j$ for $j = 1, \cdots, k$, i.e., that

$$H_1 \cdots H_k = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k, \cdots], \quad 1 \leq k \leq i. \tag{13}$$

Certainly $\mathcal{K}_1(A^TA, A^T\boldsymbol{b}) = \mathbb{R}(\boldsymbol{v}_1)$. It clearly holds if $i = 1$. Suppose for some $i > 1$ that the iteration has produced $V_i = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i]$ with orthonormal columns such that

$$\mathcal{K}_i(A^TA, A^T\boldsymbol{b}) = \text{span}\{\boldsymbol{v}_1, (A^TA)\boldsymbol{v}_1, \cdots, (A^TA)^{i-1}\boldsymbol{v}_1\} = \text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i\}.$$

It is easy to see from (4) that

$$V_i^TA^TAV_i = B_i^TB_i$$

and we have $V_i^T\boldsymbol{r}_i = 0$, where $\boldsymbol{r}_i = A^TA\boldsymbol{v}_i - (\alpha_i^2 + \beta_{i+1}^2)\boldsymbol{v}_i - \alpha_{i-1}\beta_{i-1}\boldsymbol{v}_{i-1}$. If $\boldsymbol{r}_i \neq 0$, then $\boldsymbol{v}_{i+1} = \boldsymbol{r}_i/\|\boldsymbol{r}_i\|_2$ is orthogonal to $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i$. It follows that $\boldsymbol{v}_{i+1} \notin \mathcal{K}_i(A^TA, A^T\boldsymbol{b})$ and

$$\boldsymbol{v}_{i+1} \in \text{span}\{A^TA\boldsymbol{v}_i, \boldsymbol{v}_i, \boldsymbol{v}_{i-1}\} \subseteq \mathcal{K}_{i+1}(A^TA, A^T\boldsymbol{b}).$$

Thus, $V_{i+1}^TV_{i+1}^T = I$ and

$$\text{span}(V_{i+1}) = \mathcal{K}_{i+1}(A^TA, A^T\boldsymbol{b}).$$

On the other hand, if $\boldsymbol{r}_i = 0$, then $A^TAV_i = V_iB_i^TB_i$. This says that $\text{span}(V_i) = \mathcal{K}_i(A^TA, A^T\boldsymbol{b})$ is invariant for A^TA and the induction is complete. The proof of $\text{span}(U_i) = \mathcal{K}_i(AA^T, AA^T\boldsymbol{b})$ is in a similar way. □

The Householder matrices H_i and G_i need not be formed explicitly. In other words, the matrices V_k and U_k can also remain in product form in the HBITLS algorithm. In floating-point operations, the Householder transformation does not have to worry too much about the loss of orthogonality.

2.2. The NIPALS-TLS Algorithm

For the NIPALS PLS algorithm, one can see in [3,13]. In this paper, we want to use it to solve the TLS problems and then form the NIPALS-TLS algorithm. We can find that the HBITLS algorithm and NIPALS-TLS algorithm generate the same sequences, orthonormal base V_k. From the uniqueness of this base, and combined with the relationship between the two algorithms, we conclude that the two algorithms generate the same numerical solution $\boldsymbol{x}_k$.

In [3], it tells us that we can set $A_0 = A$, $\boldsymbol{b}_0 = \boldsymbol{b}$, for $k = 1, 2, \cdots$, we can produce sequences $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ according to the following form:

$$\begin{aligned} \boldsymbol{v}_k &= A_{k-1}^T\boldsymbol{b}_{k-1}/\mu_k, \ \ \mu_k = \|A_{k-1}^T\boldsymbol{b}_{k-1}\|_2, && (14)\\ \boldsymbol{p}_k &= A_{k-1}\boldsymbol{v}_{k-1}/\rho_k, \ \ \rho_k = \|A_{k-1}\boldsymbol{v}_{k-1}\|_2, && (15)\\ (A_k, \boldsymbol{b}_k) &= (I - \boldsymbol{p}_k\boldsymbol{p}_k^T)(A_{k-1}, \boldsymbol{b}_{k-1}). && (16) \end{aligned}$$

In (16) A_k and $\boldsymbol{b}_k$ are formed by deflated A_{k-1} and $\boldsymbol{b}_{k-1}$ by subtracting their orthogonal projections onto $\boldsymbol{p}_k$. We know that this operation uses elementary orthogonal transformations, such that $S = I - \boldsymbol{p}\boldsymbol{p}^T$, $\|\boldsymbol{p}\|_2 = 1$. The deflation in (16) can also be written as

$$\begin{aligned} A_k &= A_{k-1} - \boldsymbol{p}_k\boldsymbol{s}_k^T, \ \ \boldsymbol{s}_k = A_{k-1}^T\boldsymbol{p}_k, && (17)\\ \boldsymbol{b}_k &= \boldsymbol{b}_{k-1} - \boldsymbol{p}_k\zeta_k^T, \ \ \zeta_k = \boldsymbol{b}_{k-1}^T\boldsymbol{p}_k. && (18) \end{aligned}$$

The process is terminated when it meets either $\|A_{k-1}^T\boldsymbol{b}_{k-1}\|_2 = 0$ or $\|A_{k-1}\boldsymbol{v}_i\|_2 = 0$. We note that if $\boldsymbol{p}_k^TA_{k-1}\boldsymbol{v}_k \neq 0$, then the rank of the matrix A_k is one less than that of A_{k-1} exactly.

Using exact arithmetic, the sets $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$ and $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k\}$ generated by (14) and (15) are the unique orthogonal bases for the Krylov sub-spaces $\mathcal{K}_k(A^TA, A^T\boldsymbol{b})$ and $\mathcal{K}_k(AA^T, AA^T\boldsymbol{b})$, respectively. Summing (17) and (18) generates

$$A = P_kS_k^T + A_k, \qquad \boldsymbol{b} = P_k\boldsymbol{z}_k + \boldsymbol{b}_k, \tag{19}$$

where $P_k = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k]$, $S_k = [\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_k]$, and $\boldsymbol{z}_k = (\zeta_1, \zeta_2, \cdots, \zeta_k)^T$. These relationships maintain working accuracy and do not depend on orthogonality. The matrix $P_k S_k^T$ is a rank-k approximation to the data matrix A. From [3], we have $S_k^T = P_k^T A$ and $S_k^T V_k = R_k$. Thus, in exact arithmetic, the matrix $S_k^T V_k$ is upper bidiagonal with its elements

$$\theta_k = \boldsymbol{s}_{k-1}^T \boldsymbol{v}_k \qquad \rho_k = \boldsymbol{s}_k^T \boldsymbol{v}_k = \|A_{k-1}\boldsymbol{v}_k\|_2, \tag{20}$$

and

$$R_k \equiv \begin{pmatrix} \rho_1 & \theta_2 & & & \\ & \rho_2 & \theta_3 & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1} & \theta_k \\ & & & & \rho_k \end{pmatrix}.$$

By Paige [14], we know that R_k must be identical to the matrix that would be obtained from the conventional QR factorization of B_k, such that

$$Q_k B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}.$$

Then we have

$$\begin{pmatrix} B_k & \beta_1 \boldsymbol{e}_1 \end{pmatrix} = Q_k^T \begin{pmatrix} R_k & \boldsymbol{z}_k \\ & \|\boldsymbol{b}_k\|_2 \end{pmatrix} \equiv Q_k^T N_k. \tag{21}$$

Let $N_k = \check{U}_k \check{\Sigma}_k \check{V}_k$, and $\check{V}_k \boldsymbol{e}_{k+1} \equiv \check{\boldsymbol{v}}_{k+1} = \begin{pmatrix} (\check{\boldsymbol{v}}_{k+1}^{(1)})^T & \check{\boldsymbol{v}}_{k+1}^{(2)} \end{pmatrix}^T$, then the solution of the projected TLS problem (9) is $\boldsymbol{y}_k = -\frac{\check{\boldsymbol{v}}_{k+1}^{(1)}}{\check{\boldsymbol{v}}_{k+1}^{(2)}}$, and the TLS solution is $\boldsymbol{x}_k = -V_k \frac{\check{\boldsymbol{v}}_{k+1}^{(1)}}{\check{\boldsymbol{v}}_{k+1}^{(2)}}$. And the following theorem comes by (12) and (21)

Theorem 1. *the HBITLS and NIPALS-TLS algorithms compute the same approximate solutions* $\boldsymbol{x}_k$.

3. Roundoff Error Analysis

In this section, we analyze the propagation of roundoff error during the process of the HBITLS algorithm and get the mixed forward–backward stability of the HBITLS algorithm and NIPALS-TLS algorithm naturally. The total roundoff error during the process of the HBITLS algorithm can be divided into the following four parts:

First, we can find that the HBITLS algorithm solves the original TLS problem (2) to a perturbed TLS problem. The propagation of the roundoff error of a Householder matrix in the HBITLS algorithm is advantageous when performing numerical computations.

From now one, we will denote by ε the machine precision under consideration. In [15], it shows that the computed Householder matrix $fl(H)$ comes near the exact Householder matrix H itself:

$$\|fl(H) - H\|_2 \leq 84\varepsilon + \mathcal{O}(\varepsilon^2).$$

Moreover, for a vector $\boldsymbol{y}$, the computed updates with $fl(H)$ are very close to the exact updates with H :

$$fl(fl(H)\boldsymbol{y}) = H(\boldsymbol{y} + \boldsymbol{w}), \quad \|\boldsymbol{w}\|_2 \leq 87\varepsilon\|\boldsymbol{y}\|_2 + \mathcal{O}(\varepsilon^2).$$

and, in general,

$$fl(fl(H_1)\cdots fl(H_j)\boldsymbol{y}) = H_1 \cdots H_j(\boldsymbol{y} + \boldsymbol{z}), \quad \|\boldsymbol{z}\|_2 \leq 87j\varepsilon\|\boldsymbol{y}\|_2 + \mathcal{O}(\varepsilon^2). \tag{22}$$

The following lemma tells us that the reduced system calculated by the HBITLS algorithm is equivalent to the system formed after the original system has been disturbed. $\bar{B}_k$, $\hat{Q}_k$ and $\hat{P}_{k+1}$ are the floating-point computation of the matrices B_k, V_k and U_{k+1} in HBITLS algorithm, respectively.

Lemma 2. *Let $\bar{B}_k$ be the computed bidiagonalization matrix $(k+1) \times k$ matrix obtained by the HBITLS algorithm. Then, there comes a perturbation matrix E and exists two column orthogonal matrices $\hat{Q}_k$ and $\hat{P}_{k+1}$ s.t.*

$$(A+E)\hat{Q}_k = \hat{P}_{k+1}\bar{B}_k,$$

and

$$\|E\|_2 \leq \sqrt{n}(174n + 3\sqrt{n} + 87)\varepsilon\|A\|_2 + \mathcal{O}(\varepsilon^2),$$

where n is the number of columns of matrix A. Furthermore, the matrix $\hat{Q}_k$ is an orthonormal basis of $\mathcal{K}_k\big((A+E)^T(A+E), (A+E)^T(\boldsymbol{b}+\boldsymbol{e})\big)$ with a perturbation vector $\boldsymbol{e}$, where $\|\boldsymbol{e}\|_2 \leq 87\varepsilon\|\boldsymbol{b}\|_2 + \mathcal{O}(\varepsilon^2)$.

Proof. We prove this theorem by induction. The key point is that we should show the computed matrix, which will be shown by introduction from (3), for $G_k^T \cdots G_1^T[\boldsymbol{v}_1, A\boldsymbol{v}_1, \cdots, A\boldsymbol{v}_k]$ as follows

$$\hat{P}_{k+1} \cdots \hat{P}_1[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, (A+G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1, \cdots, (A+G_k)\bar{\boldsymbol{v}}_k + \boldsymbol{w}_k].$$

For $k=1$, first, let $\boldsymbol{u}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|_2$, $\bar{\boldsymbol{u}}_1 = fl(\boldsymbol{u}_1)$, a Householder matrix P_1 is found s.t. $P_1\boldsymbol{u}_1 = \boldsymbol{e}_1$. Set $\bar{P}_1 = fl(P_1)$, ref. [16] tells us that, corresponding to matrix $\bar{P}_1$, we can find a Householder matrix to make

$$fl(\bar{P}_1\bar{\boldsymbol{v}}_1) = \hat{P}_1(\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1),$$

with

$$\|\boldsymbol{g}_1\|_2 \leq 87\varepsilon\|\bar{\boldsymbol{v}}_1\|_2 + \mathcal{O}(\varepsilon^2) = 87\varepsilon + \mathcal{O}(\varepsilon^2).$$

Next, let $\boldsymbol{v}_1 = A^T\boldsymbol{u}_1/\|A^T\boldsymbol{u}_1\|_2$, $fl(A^T\boldsymbol{u}_1) = (A+G_0)^T\bar{\boldsymbol{u}}_1$, where $\|G_0\|_2 \leq \varepsilon 3\sqrt{n}\|A\|_2$. Similarly, for $A\boldsymbol{v}_1$, the computed result can be written as $fl(A\boldsymbol{v}_1) = (A+G_1)\bar{\boldsymbol{v}}_1$, where $\|G_1\|_2 \leq \varepsilon 3\sqrt{n}\|A\|_2$. Now, we set the Householder matrix P_2 s.t. $P_2P_1[\boldsymbol{v}_1, A\boldsymbol{v}_1]$ is upper bidiagonal matrix. We know P_2 only works the vector $P_1A\boldsymbol{v}_1$, so there's no change for the 1st column of $fl(\bar{P}_1[fl(\boldsymbol{v}_1), fl(A\boldsymbol{v}_1)])$ when producted by $\bar{P}_2$ and $\hat{P}_2$. Likely, there's a Householder matrix $\hat{P}_2$ s.t. $\hat{P}_2 fl(\bar{P}_1[\bar{\boldsymbol{v}}_1, A\bar{\boldsymbol{v}}_1]$ is bidiagonal matrix in theory, but the algorithm computes a matrix $\bar{P}_2$ in practice [16] such that

$$fl(\bar{P}_2\bar{P}_1 fl(A\boldsymbol{v}_1)) = \hat{P}_2\hat{P}_1((A+G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1),$$

where

$$\|\boldsymbol{w}_1\|_2 \leq 174\varepsilon(\|A\|_2 + \|G_1\|_2)\|\bar{\boldsymbol{v}}_1\|_2 = 174\varepsilon\|A\|_2 + \mathcal{O}(\varepsilon^2).$$

Finally, we have

$$\begin{aligned} fl(\bar{P}_2\bar{P}_1[\bar{\boldsymbol{v}}_1, fl(A\boldsymbol{v}_1)]) &= \big[fl(\bar{P}_1\bar{\boldsymbol{v}}_1), fl(\bar{P}_2 fl(\bar{P}_1 fl(A\boldsymbol{v}_1)))\big] \\ &= \big[\hat{P}_1(\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1), \hat{P}_2\hat{P}_1((A+G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1)\big] \\ &= \hat{P}_2\hat{P}_1\big[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, (A+G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1\big]. \end{aligned}$$

For the kth step, assume that the HBITLS algorithm has calculated the matrices $\bar{P}_1, \cdots, \bar{P}_k$, associated with the Householder matrices $\hat{P}_1, \cdots, \hat{P}_k$. Then, after k steps, we can get the following result:

$$\begin{aligned} &fl(\bar{P}_k \cdots \bar{P}_1[\bar{\boldsymbol{v}}_1, fl(A\boldsymbol{v}_1), \cdots, fl(A\boldsymbol{v}_k)]) \\ &= \hat{P}_k \cdots \hat{P}_1\big[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, (A+G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1, \cdots, (A+G_k)\bar{\boldsymbol{v}}_k + \boldsymbol{w}_k\big], \end{aligned}$$

where $\|\boldsymbol{w}_i\|_2 \leq 87(i+1)\varepsilon\|A\|_2 + \mathcal{O}(\varepsilon^2)$.

We know $\boldsymbol{v}_k = Q_1 \cdots Q_k \boldsymbol{e}_k$, and $\bar{\boldsymbol{v}}_k = fl(\bar{Q}_1 \cdots \bar{Q}_k \boldsymbol{e}_k) = \hat{Q}_1 \cdots \hat{Q}_k(\boldsymbol{e}_k + \boldsymbol{f}_k)$, where $\|\boldsymbol{f}_k\|_2 \leq 87k\varepsilon + \mathcal{O}(\varepsilon^2)$. For $A\boldsymbol{v}_k$, the floating-point vector is $fl(A\boldsymbol{v}_k) = (A + G_k)\bar{\boldsymbol{v}}_k$, where $\|G_k\|_2 \leq \varepsilon 3\sqrt{n}\|A\|_2$. Likely, there is a Householder matrix P_{k+1}, which only works on the vector $P_k \cdots P_1 A\boldsymbol{v}_k$, s.t. $P_{k+1}P_k \cdots P_1[\boldsymbol{v}_1, A\boldsymbol{v}_1, \cdots, A\boldsymbol{v}_k]$ is the upper bidiagonal matrix. The algorithm computes a matrix $\bar{P}_{k+1}$ in practice so that

$$fl(\bar{P}_{k+1} \cdots \bar{P}_1 fl(A\boldsymbol{v}_k)) = \hat{P}_{k+1} \cdots \hat{P}_1((A + G_k)\bar{\boldsymbol{v}}_k + \boldsymbol{w}_k),$$

where

$$\|\boldsymbol{w}_k\|_2 \leq 87(k+1)\varepsilon(\|A\|_2 + \|G_k\|_2)(1 + \|\boldsymbol{f}_k\|_2) \leq 87(k+1)\varepsilon\|A\|_2 + \mathcal{O}(\varepsilon^2).$$

Then the floating-point matrix is obtained, such that

$$\hat{P}_{k+1} \cdots \hat{P}_1[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, (A + G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1, \cdots, (A + G_k)\bar{\boldsymbol{v}}_k + \boldsymbol{w}_k].$$

Let $\hat{P}^{(k)}$, $\hat{Q}^{(k)}$ be the matrices, such that $(\hat{P}^{(k)})^T = \hat{P}_k \cdots \hat{P}_1$ and $(\hat{Q}^{(k)})^T = \hat{Q}_k \cdots \hat{Q}_1$, respectively, we find that the first $(i-1)$ rows of each $\hat{Q}_i$ is $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{i-1}$. Let $\boldsymbol{q}_i^{(j)}$ be the i-th column of $\hat{Q}^{(j)}$, then the results are as follows

$$\boldsymbol{q}_i^{(j)} = \boldsymbol{q}_i^{(k)}, \quad \forall k \geq j, \quad i = 1, \cdots, j.$$

Then there comes

$$(\hat{P}^{(k+1)})^T[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, (A + G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1, \cdots, (A + G_k)\bar{\boldsymbol{v}}_k + \boldsymbol{w}_k],$$

it is an $n \times (k+1)$ upper bidiagonal matrix. And, $\forall j \leq k$, we obtain

$$\begin{aligned}
(A + G_j)\bar{\boldsymbol{v}}_j + \boldsymbol{w}_j &= (A + G_j)\hat{Q}_1 \cdots \hat{Q}_j(\boldsymbol{e}_j + \boldsymbol{f}_j) + \boldsymbol{w}_j \\
&= A\hat{Q}^{(j)}\boldsymbol{e}_j + G_j\hat{Q}^{(j)}\boldsymbol{e}_j + A\hat{Q}^{(j)}\boldsymbol{f}_j + G_j\hat{Q}^{(j)}\boldsymbol{f}_j + \boldsymbol{w}_j \\
&= Aq_j^{(j)} + G_j q_j^{(j)} + A\hat{Q}^{(j)}\boldsymbol{f}_j + G_j\hat{Q}^{(j)}\boldsymbol{f}_j + \boldsymbol{w}_j.
\end{aligned}$$

Since $\forall k \leq j$, $\boldsymbol{q}_j^{(j)}$ is the j-th column $\boldsymbol{q}_j$ of the matrix $\hat{Q}^{(j)}$, if we denote by $\boldsymbol{y}_j = G_j\boldsymbol{q}_j^{(j)} + A\hat{Q}^{(j)}\boldsymbol{f}_j + G_j\hat{Q}^{(j)}\boldsymbol{f}_j + \boldsymbol{w}_j$, we have

$$(A + G_j)\bar{\boldsymbol{v}}_j + \boldsymbol{w}_j = A\boldsymbol{q}_j + \boldsymbol{y}_j,$$

and so we can obtain

$$\begin{aligned}
& (\hat{P}^{(k+1)})^T[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, (A + G_1)\bar{\boldsymbol{v}}_1 + \boldsymbol{w}_1, \cdots, (A + G_k)\bar{\boldsymbol{v}}_k + \boldsymbol{w}_k] \\
= \; & (\hat{P}^{(k+1)})^T[\bar{\boldsymbol{v}}_1 + \boldsymbol{g}_1, A\boldsymbol{q}_1 + \boldsymbol{y}_1, \cdots, A\boldsymbol{q}_k + \boldsymbol{y}_k] \\
= \; & (\hat{P}^{(k+1)})^T\big[(\bar{\boldsymbol{v}}_1, A\boldsymbol{q}_1, \cdots, A\boldsymbol{q}_k) + (\boldsymbol{g}_1, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_k)\big].
\end{aligned}$$

If we cut off the first column of the matrix, we can set $\tilde{B}_k$ with $n \times k$ such that $\tilde{B}_k = (\bar{B}_k^T, 0)^T$, here $\bar{B}_k$ is an $(k+1) \times k$ upper bidiagonal matrix. If we denote $F_i = [\boldsymbol{y}_l, \cdots, \boldsymbol{y}_i]$, $\forall i$, then $\tilde{B}_k = (\hat{P}^{(k+1)})^T\big(A[\boldsymbol{q}_1, \cdots, \boldsymbol{q}_k] + F_k\big)$. In addition, $\forall j \leq k$, let $\hat{Q}_j$ be the matrix made up of the first j columns of $\hat{Q}^{(k)}$, we obtain $\hat{P}^{(k+1)}\tilde{B}_k = A\hat{Q}^{(k)} + F_k$, and, from the structure of $\tilde{B}_k$, there comes $\hat{P}_{k+1}\bar{B}_k = A\hat{Q}_k + F_k$. Then we can write $F_k = F_n\hat{Q}_n^T\hat{Q}_k$ owing to $\hat{Q}_n^T\hat{Q}_k = \begin{pmatrix} I_k \\ 0 \end{pmatrix}$, and so

$$\hat{P}_{k+1}\bar{B}_k = (A + F_n\hat{Q}_n^T)\hat{Q}_k.$$

If $E = F_n \hat{Q}_n^T$, we can finish the proof of the first part of the lemma, because

$$\begin{aligned}
\|E\|_2 &\leq \|E\|_F = \|F_n\|_F \\
&= \sqrt{\sum_{j=1}^{n} \|\boldsymbol{y}_j\|_2^2} \\
&\leq \sqrt{\sum_{j=1}^{n} \|G_j \boldsymbol{q}_j^{(j)} + A\hat{Q}^{(j)} \boldsymbol{f}_j + \boldsymbol{w}_j\|_2^2 + \mathcal{O}(\varepsilon^2)} \\
&\leq \sqrt{n} \max_j \|G_j \boldsymbol{q}_j^{(j)} + A\hat{Q}^{(j)} \boldsymbol{f}_j + \boldsymbol{w}_j\|_2 + \mathcal{O}(\varepsilon^2) \\
&\leq \sqrt{n}(174n + 3\sqrt{n} + 87)\varepsilon \|A\|_2 + \mathcal{O}(\varepsilon^2).
\end{aligned}$$

Finally, we prove that the subspace spanned by the columns of the matrix $\hat{Q}_k$ is an orthogonal basis of a Krylov space. Let $\tilde{A} = A + E$, $\tilde{b} = \|A^T \boldsymbol{b}\|_2 \hat{Q}_k \boldsymbol{e}_1$ and form $\mathcal{K}_k(\tilde{A}^T\tilde{A}, \tilde{A}^T\tilde{\boldsymbol{b}})$. We know that $\tilde{\boldsymbol{b}} = \|A^T \boldsymbol{b}\|_2 \hat{Q}_k \boldsymbol{e}_1 = \|A^T \boldsymbol{b}\|_2 \hat{Q}_1 \boldsymbol{e}_1$ and set $e = \tilde{\boldsymbol{b}} - A^T \boldsymbol{b}$, then we have $\|\boldsymbol{e}\|_2 = \|A^T \boldsymbol{b}\|_2 \|(\hat{Q}_1 - Q_1)\boldsymbol{e}_1\|_2 \leq 87\varepsilon \|A^T \boldsymbol{b}\|_2 + \mathcal{O}(\varepsilon^2)$.

We still prove the rest of the theorem by induction; that is, to prove

$$(\tilde{A}^T\tilde{A})^i \tilde{A}^T \tilde{\boldsymbol{b}} = \hat{Q}_k \boldsymbol{r}_i, \quad \forall i \leq k-1,$$

where each vector $\boldsymbol{r}_i$ has only the first $(i+1)$ components, which are different from zero.

For $i = 1$, we have

$$\tilde{A}^T\tilde{A}\tilde{A}^T\tilde{\boldsymbol{b}} = \tilde{A}^T\tilde{A}\|A^T \boldsymbol{b}\|_2 \hat{Q}_k \boldsymbol{e}_1 = \|A^T \boldsymbol{b}\|_2 \hat{Q}_k \bar{B}_k^T \bar{B}_k \boldsymbol{e}_1 = \hat{Q}_k \boldsymbol{r}_1,$$

since the last component of the vector $\bar{B}_k^T \bar{B}_k \boldsymbol{e}_1$ is zero, in addition, except for the first two components, the rest of the components of vector $\boldsymbol{r}_1$ are all zero.

Suppose for a given i the following relation is true,

$$(\tilde{A}^T\tilde{A})^{i+1} \tilde{A}^T \tilde{\boldsymbol{b}} = (\tilde{A}^T\tilde{A})\hat{Q}_k \boldsymbol{r}_i = \hat{Q}_{k+1} \bar{B}_k^T \bar{B}_k \boldsymbol{r}_i,$$

and in the next step, we will show it is true for $i+1$.

From the inductive hypothesis, we know that only the first $(i+1)$ components of $\boldsymbol{r}_i$ are not zero; therefore, the last component is zero of the vector $\bar{B}_k^T \bar{B}_k \boldsymbol{r}_i$ with $(i+2)$ non-zero elements. Then there comes a conclusion that

$$(\tilde{A}^T\tilde{A})^{i+1} \tilde{A}^T \tilde{\boldsymbol{b}} = \hat{Q}_k \boldsymbol{r}_{i+1}$$

and, hence, the lemma is proved. □

Based on Lemma 2 and Algorithm 1, for the bidiagonalization matrix $\bar{B}_k$ obtained by the HBITLS algorithm, one can find an orthonormal matrix $\hat{Q}_k$ s.t.

$$(A + E)\hat{Q}_k = \hat{P}_{k+1} \bar{B}_k,$$

where $\hat{Q}_k$ is just an orthogonal basis of $\mathcal{K}_k((A+E)^T(A+E), (A+E)^T(\boldsymbol{b}+\boldsymbol{e}))$. Based on this, we know that the first part of HBITLS , in exact arithmetic, gives the exact basis of the perturbed Krylov space $\mathcal{K}_k((A+E)^T(A+E), (A+E)^T(\boldsymbol{b}+\boldsymbol{e}))$. A perturbation bound for TLS solutions is given by Xie and Wei [17], see Lemma 3, which is related to the smallest singular value σ_{n+1} of $(A, \boldsymbol{b})$, the TLS solution $\boldsymbol{x}$, and the residual $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$. Let $\hat{A} = A + \Delta A$ and $\hat{\boldsymbol{b}} = \boldsymbol{b} + \Delta \boldsymbol{b}$. Then, the unique solution of the perturbed TLS problem can be expressed as $\hat{x}$. Denote $\kappa_{\boldsymbol{b}} = \frac{\|\boldsymbol{b}\|_2}{\|\boldsymbol{x}\|_2} \|B_\lambda^{-1} A^T\|_2$ and $\kappa_A = \frac{\|A\|_2}{\|\boldsymbol{x}\|_2} \left(\|\boldsymbol{r}\|_2 \|B_\lambda^{-1}\|_2 + \|x\|_2 \|B_\lambda^{-1} A^T\|_2\right)$ with $B_\lambda = A^T A - \sigma_{n+1}^2 I$. The perturbation bound is obtained under the genericity condition $\bar{\sigma}_n > \sigma_{n+1}$, where $\bar{\sigma}_n$ is the smallest singular value of A

Lemma 3 ([17]). *Consider the TLS problem (2) and assume that the genericity condition holds. If* $\|(\Delta A, \Delta \boldsymbol{b})\|_F$ *is sufficiently small, then we obtain that*

$$\frac{\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}\|_2} \lesssim \kappa_{\boldsymbol{b}} \frac{\|\Delta \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2} + \kappa_A \frac{\|\Delta A\|_2}{\|A\|_2}.$$

Suppose that $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the exact TLS solutions of $A\boldsymbol{x} \approx \boldsymbol{b}$ and $(A+E)\hat{\boldsymbol{x}} \approx \boldsymbol{b}+\boldsymbol{e}$ respectively. The error introduced in this part of the HBITLS algorithm is the inherent error, so we can give $\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2$ by Lemma 3 and Lemma 2 easily, see Theorem 2.

Secondly, let us consider the error between the TLS solution of the system $(A+E)\hat{\boldsymbol{x}} \approx \boldsymbol{b}+\boldsymbol{e}$ and the approximation solution $\hat{\boldsymbol{x}}_k = \hat{V}_k \hat{\boldsymbol{y}}_k$ of the system computed by the HBITLS algorithm at step k with the exact arithmetic, i.e., $\hat{\boldsymbol{y}}_k$ is the exact solution of the reduced TLS problem

$$\min \|(\beta_1 \boldsymbol{e}_1, \bar{B}_k) - (\check{\boldsymbol{e}}_k, \check{B}_k)\|_F \quad \text{subject to} \quad \check{B}_k \hat{\boldsymbol{y}}_k = \check{\boldsymbol{e}}_k, \tag{23}$$

For convenience, define $\bar{V}_k = \begin{pmatrix} 1 & 0 \\ 0 & V_k \end{pmatrix}$ and let

$$(\boldsymbol{b}, A) = W \Sigma Z^T, \tag{24}$$

where W and $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_{n+1}]$ are orthogonal matrices of dimension m and $n+1$, respectively, Σ is an $m \times (n+1)$ diagonal matrix whose diagonal entries σ_i are the singular values of $(\boldsymbol{b}, A)$, sorted in non-increasing order. Let $\theta(\boldsymbol{u}, \boldsymbol{v})$ be the subspace angle between $\mathbb{R}(\boldsymbol{u})$ and $\mathbb{R}(\boldsymbol{v})$.

Lemma 4. *Let* $\boldsymbol{x}$ *denotes the essential TLS solution to the linear system (2) satisfying genericity condition and* $\boldsymbol{x}_k$ *be the approximation solution obtained from Algorithm 1. Then*

$$\sin\theta(\boldsymbol{z}, \boldsymbol{z}^{(k)}) \le \|\boldsymbol{x}-\boldsymbol{x}_k\| \le \sin\theta(\boldsymbol{z}, \boldsymbol{z}^{(k)}) \sqrt{1+\|\boldsymbol{x}\|^2} \sqrt{1+\|\boldsymbol{x}_k\|^2}, \tag{25}$$

where

$$\begin{aligned} \boldsymbol{z} \equiv \boldsymbol{z}_{n+1} &= \begin{pmatrix} \boldsymbol{z}_{12} \\ \boldsymbol{z}_{22} \end{pmatrix} \in \mathbb{R}^{n+1}, \\ \boldsymbol{z}^{(k)} \equiv \begin{pmatrix} \boldsymbol{z}_{12}^{(k)} \\ \boldsymbol{z}_{22}^{(k)} \end{pmatrix} &= \bar{V}_k \bar{\boldsymbol{z}}_{k+1} = \begin{pmatrix} 1 & 0 \\ 0 & V_k \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{z}}_{12} \\ \bar{\boldsymbol{z}}_{22} \end{pmatrix} \in \mathbb{R}^{n+1}. \end{aligned} \tag{26}$$

Proof. It is easy to know that

$$\begin{pmatrix} -1 \\ \boldsymbol{x} \end{pmatrix} - \begin{pmatrix} -1 \\ \boldsymbol{x}_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{z}_{12}^{(k)} \\ \boldsymbol{z}_{22}^{(k)} \end{pmatrix} (\boldsymbol{z}_{12}^{(k)})^{-1} - \begin{pmatrix} \boldsymbol{z}_{12} \\ \boldsymbol{z}_{22} \end{pmatrix} \boldsymbol{z}_{12}^{-1}. \tag{27}$$

Then we have an orthonormal matrix $G \in \mathbb{R}^{(n+1)\times n}$ with the partition

$$G = \begin{matrix} n \\ \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \\ n \end{matrix} \begin{matrix} \\ 1 \\ \end{matrix}$$

such that $G^T((\boldsymbol{z}_{12}^{(k)})^T(\boldsymbol{z}_{22}^{(k)})^T)^T = 0$ (i.e., G "forms a complete space"). From Equation (27) there comes

$$G^T \left[\begin{pmatrix} -1 \\ \boldsymbol{x} \end{pmatrix} - \begin{pmatrix} -1 \\ \boldsymbol{x}_k \end{pmatrix} \right] = -G^T \begin{pmatrix} \boldsymbol{z}_{12} \\ \boldsymbol{z}_{22} \end{pmatrix} \boldsymbol{z}_{12}^{-1}, \tag{28}$$

and, therefore

$$G_2^T(\boldsymbol{x}-\boldsymbol{x}_k) = -G^T\begin{pmatrix} z_{12} \\ z_{22} \end{pmatrix} z_{12}^{-1}. \tag{29}$$

From the CS theorem [18], we know that $\sigma_{min}(G_2)^{-1} = (z_{12}^{(k)})^{-1}$. Then

$$\begin{aligned} \sigma_{min}(G_2)\|(\boldsymbol{x}-\boldsymbol{x}_k)\| &\leq \|G_2^T(\boldsymbol{x}-\boldsymbol{x}_k)\| \\ &= \|G^T\begin{pmatrix} z_{12} \\ z_{22} \end{pmatrix} z_{12}^{-1}\| \\ &\leq \sin\theta(z, z^{(k)}) z_{12}^{-1}. \end{aligned}$$

It was noticed that

$$\sin\theta(z, z^{(k)}) = \|G^T\begin{pmatrix} z_{12} \\ z_{22} \end{pmatrix}\|$$

denotes the sine of the subspace angle between $\mathbb{R}(z)$ and $\mathbb{R}(z^{(k)})$. Hence, the upper bound can be proved as follows

$$\begin{aligned} \|\boldsymbol{x}-\boldsymbol{x}_k\| &\leq \sin\theta(z, z^{(k)}) z_{12}^{-1} (z_{12}^{(k)})^{-1} \\ &= \sin\theta(z, z^{(k)}) \sqrt{1+\|\boldsymbol{x}\|^2}\sqrt{1+\|\boldsymbol{x}_k\|^2}. \end{aligned}$$

For the lower bound, we have

$$\begin{aligned} \|\boldsymbol{x}-\boldsymbol{x}_k\| &\geq \|G_2^T(\boldsymbol{x}-\boldsymbol{x}_k)\| \\ &= \|G^T\begin{pmatrix} z_{12} \\ z_{22} \end{pmatrix} z_{12}^{-1}\| \\ &\geq z_{12}^{-1}\sin\theta(z, z^{(k)}) \\ &\geq \sin\theta(z, z^{(k)}). \end{aligned}$$

Since $z_{12}^{-1} \geq 1$, and this proves the upper bound case. □

Thirdly, we need to consider how to solve problem (9) and show that the error is between the solution obtained by this method and the theoretical solution. Let the computed solution be

- $\bar{\boldsymbol{x}}_k = \hat{V}_k \bar{\boldsymbol{y}}_k$, where $\bar{\boldsymbol{y}}_k$ is the computed solution of the problem (23).

In [19], James and Kahan posed an algorithm named QR iteration with a zero shift, which guaranteed forward stability. Furthermore, an implicit algorithm about it is given. Error analysis including the singular values and singular vectors are also given, which is just what we've needed.

Lemma 5 ([19]). *Let the matrix $\bar{B}$ obtained by running the implicit zero-shift QR algorithm on a bidiagonal matrix B with $n \times n$. Suppose that all perturbation angles θ emerged from the operations of the algorithm satisfy $\sin^2\theta \leq \tau < 1$. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ and $\bar{\sigma}_1 \geq \bar{\sigma}_2 \geq \cdots \geq \bar{\sigma}_n$ are the singular values of B and $\bar{B}$ respectively. If*

$$\omega \equiv \frac{88n\varepsilon}{(1-\tau)^2} < 1, \tag{30}$$

then we have:

$$|\sigma_i - \bar{\sigma}_i| \leq \frac{\omega}{1-\omega}\sigma_i.$$

Moreover, let $\sigma_{k1} \geq \sigma_{k2} \geq \cdots \geq \sigma_{kn}$ *be the singular values of* B_k *produced after k steps of the implicit zero-shift QR algorithm. Then if condition (30) holds, and all perturbation angles* θ *satisfy* $\sin^2\theta \leq \tau < 1$, *we obtain*

$$|\sigma_i - \sigma_{ki}| \leq \left(\frac{1}{(1-\omega)^k - 1}\sigma_i\right) \approx \frac{88kn\varepsilon}{(1-\tau)^2}\sigma_i.$$

James and Kahan [19] also give the relative differences between the singular vectors of B and the ones of $\bar{B}$.

Lemma 6 ([19]). *Let* σ_i *be the singular value of be an unreduced bidiagonal matrix B with* $\boldsymbol{u}_i$ *and* $\boldsymbol{v}_i$ *being its corresponding left and right singular vectors, respectively. Let* $\bar{\boldsymbol{u}}_i$ *and* $\bar{\boldsymbol{v}}_i$ *be the singular vectors computed by the implicit zero-shift QR algorithm. Then the bound of the errors in* $\bar{\boldsymbol{v}}_i$ *are shown by*

$$\theta(\bar{\boldsymbol{v}}_i, \boldsymbol{v}_i) \leq p(n)\varepsilon/rel_{gap} \equiv p(n)\varepsilon/\min(|\sigma_i - \sigma_{i+1}|/\sigma_i). \tag{31}$$

Then, combining with the perturbation bound of TLS given in [20] as shown in Lemma 7, we can give the error estimate $\|\hat{\boldsymbol{y}}_k - \bar{\boldsymbol{y}}_k\|_2$.

If let $(\tilde{A}, \tilde{\boldsymbol{b}})$ is a rank-k matrix approximation to $(A, \boldsymbol{b})$, and $(\Delta\tilde{A}, \Delta\tilde{\boldsymbol{b}}) = (A, \boldsymbol{b}) - (\tilde{A}, \tilde{\boldsymbol{b}})$. Let $(\bar{A}, \bar{\boldsymbol{b}}) = (A, \boldsymbol{b}) + (\Delta A, \Delta\boldsymbol{b})$ represent a perturbation of $(A, \boldsymbol{b})$, $(\check{A}, \check{\boldsymbol{b}})$ denote a rank-k matrix approximation to $(\bar{A}, \bar{\boldsymbol{b}})$ and define $(\Delta\check{A}, \Delta\check{\boldsymbol{b}}) = (\bar{A}, \bar{\boldsymbol{b}}) - (\check{A}, \check{\boldsymbol{b}})$, then

Lemma 7 ([20]). *Let* $\boldsymbol{x}$ *and* $\bar{\boldsymbol{x}}$ *denote the TLS solution and the perturbed TLS solution. If* $\max(\|\Delta\tilde{A}\|, \|\Delta\check{A}\| + \|\Delta A\| < \sigma_k^{'}$*(the k-th singular value of A) may be provided. Then*

$$\sin\theta(\boldsymbol{v}, \bar{\boldsymbol{v}}) \leq \|\boldsymbol{x} - \bar{\boldsymbol{x}}\| \leq \sin\theta(\boldsymbol{v}, \bar{\boldsymbol{v}})\sqrt{1 + \|\boldsymbol{x}\|^2}\sqrt{1 + \|\bar{\boldsymbol{x}}\|^2}, \tag{32}$$

where $\boldsymbol{v}$ *and* $\bar{\boldsymbol{v}}$ *are the smallest right singular vectors of* $(A, \boldsymbol{b})$ *and* $(\bar{A}, \bar{\boldsymbol{b}})$ *respectively.*

In summary, if let $\tilde{\boldsymbol{x}}_k = fl(\bar{V}_k\bar{\boldsymbol{y}}_k)$ be the final computed solution at the k-th step, then roundoff error analysis of HBITLS algorithm for TLS problem can be shown as follows

Theorem 2. *Considering the HBITLS algorithm at step k , the roundoff error emerged during the algorithm can be bounded as follows:*

$$\begin{aligned}\|\boldsymbol{x} - \tilde{\boldsymbol{x}}_k\| \quad &\leq \quad \kappa_{\boldsymbol{b}}87\varepsilon + \kappa_A\sqrt{n}(174n + 3\sqrt{n} + 87)\varepsilon \\ &+ \quad \sin\theta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{z}}^{(k)})\sqrt{1 + \|\hat{\boldsymbol{x}}\|^2}\sqrt{1 + \|\hat{\boldsymbol{x}}_k\|^2} \\ &+ \quad \sin\theta(\hat{\boldsymbol{z}}^{(k)}, \check{\boldsymbol{z}})\sqrt{1 + \|\hat{\boldsymbol{y}}_k\|^2}\sqrt{1 + \|\bar{\boldsymbol{y}}_k\|^2} \\ &+ \quad 87k\varepsilon\|\bar{\boldsymbol{y}}_k\| + \mathcal{O}(\epsilon^2),\end{aligned}$$

where $\hat{\boldsymbol{z}}$ *and* $\hat{\boldsymbol{z}}^{(k)}$ *are defined in (26) similarly,* $\check{\boldsymbol{z}}$ *is the computed smallest right singular vector of* $(\beta_1\boldsymbol{e}_1, \bar{B}_k)$.

Proof. The roundoff error can be composed of the following parts

$$\boldsymbol{x} - \tilde{\boldsymbol{x}}_k = (\boldsymbol{x} - \hat{\boldsymbol{x}}) + (\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_k) + (\hat{\boldsymbol{x}}_k - \hat{V}_k\bar{\boldsymbol{y}}_k) + (\hat{V}_k\bar{\boldsymbol{y}}_k - fl(\bar{V}_k\bar{\boldsymbol{y}}_k)$$

and we analyze these errors separately.

For the first part, $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the TLS solutions of the systems $A\boldsymbol{x} \approx \boldsymbol{b}$ and $(A + E)\hat{\boldsymbol{x}} \approx \boldsymbol{b} + \boldsymbol{e}$, respectively, in line with Lemma 2, so the error of this part is the inherent error. Then, combining with Lemma 3, we have

$$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\| \leq \kappa_{\boldsymbol{b}}87\varepsilon + \kappa_A\sqrt{n}(174n + 3\sqrt{n} + 87)\varepsilon,$$

where κ_A and $\kappa_{\boldsymbol{b}}$ see Lemma 3.

For the second part, this error is owing to the approximate solution of $(A+E)\hat{\boldsymbol{x}} \approx \boldsymbol{b}+\boldsymbol{e}$ obtained by using HBITLS algorithm after k steps with the exact arithmetic. Lemma 4 tells us that

$$\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_k\| \leq \sin\theta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{z}}^{(k)})\sqrt{1+\|\hat{\boldsymbol{x}}\|^2}\sqrt{1+\|\hat{\boldsymbol{x}}_k\|^2}.$$

For the third part, it is noticed that $\hat{\boldsymbol{x}}_k = \hat{V}_k\hat{\boldsymbol{y}}_k$, we have that $\|\hat{\boldsymbol{x}}_k - \hat{V}_k\bar{\boldsymbol{y}}_k\| = \|\hat{\boldsymbol{y}}_k - \bar{\boldsymbol{y}}_k\|$, where $\|\hat{\boldsymbol{y}}_k - \bar{\boldsymbol{y}}_k\|$ is the roundoff error stem from the projected TLS solution. Since B_k is a special form of bidiagonal matrices, we consider using the implicit zero-shift QR algorithm to perform singular value decomposition. (31) gives an upper bound of the angle between the solution vectors, and combining Lemma 7, we know

$$\|\hat{\boldsymbol{x}}_k - \hat{V}_k\bar{\boldsymbol{y}}_k\| = \|\hat{\boldsymbol{y}}_k - \bar{\boldsymbol{y}}_k\| \leq \sin\theta(\hat{\boldsymbol{z}}^{(k)}, \check{\boldsymbol{z}})\sqrt{1+\|\hat{\boldsymbol{y}}_k\|^2}\sqrt{1+\|\bar{\boldsymbol{y}}_k\|^2},$$

where $\theta(\hat{\boldsymbol{z}}^{(k)}, \check{\boldsymbol{z}})$ is the subspace angle between the sub-spaces produced by the smallest right singular vector and the computed smallest right singular vector of $(\beta_1\boldsymbol{e}_1, \bar{B}_k)$, respectively.

For the last part, we know $V_k = H_1H_2\cdots H_k\begin{pmatrix} I_k \\ 0 \end{pmatrix}$, where $H_1H_2\cdots H_k$ is the product of k Householder matrices. So, on the basis of (22), we obtain

$$\|\hat{V}_k\bar{\boldsymbol{y}}_k - fl(\bar{V}_k\bar{\boldsymbol{y}}_k)\| \leq 87k\varepsilon\|\bar{\boldsymbol{y}}_k\| + \mathcal{O}(\epsilon^2).$$

□

By theorem 2, we get the mixed forward–backward stability of the HBITLS algorithm and NIPALS-TLS algorithm naturally. The backward stability will generate perturbation that will marginally influence the theoretical convergence of the residual to zero.

Remark 2. *The bound we introduced in Theorem 2 shows that the total roundoff errors are dominated by the approximation errors $\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_k\|$. From this, we can know that, in many practical problems, we can safely select the stopping criteria required by the algorithm based on the theoretical nature of the original problem. This shows that, in a great deal of practical studies, the stopping criteria may be effectively selected based on the theoretical properties of the problem itself, thereby reducing the cost required to pursue an extremely accurate approximate solution to the original problem.*

4. Conclusions

For large-scale problems, how to give an algorithm with good accuracy and stability is particularly important. In this paper, the Householder bidiagonalization total least squares (HBITLS) algorithm and nonlinear iterative partial least squares (NIPALS-TLS) algorithm are given. The HBITLS uses the Householder bidiagonalization algorithm for reducing (b, A) to upper bidiagonal form and then runs the implicit zero-shift QR algorithm to compute the smallest right singular vector of the reduced form for the approximation solutions. The NIPALS-TLS is based on rank-reducing orthogonal projections. The two algorithms compute the same approximate TLS solutions. By analyzing the propagation of the roundoff error during the process of the HBITLS algorithm, we find that the HBITLS algorithm and the NIPALS-TLS algorithm are to be mixed forward–backward stable. In addition, in many practical problems, the stop criterion can be safely selected on the basis of the rounding error analysis of the original problem. The upper bound of our roundoff error gives a more detailed and clearer approximation of the computed solution.

Author Contributions: Conceptualization, Z.Y. and X.L.; methodology, Z.Y. and X.L.; formal analysis, Z.Y.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, Z.Y. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the NNSF of China (11571004, 11701456), Natural Science Foundation of Qinghai Province (2018-ZJ-717), Foundation Sciences Qinghai Nationalities University(2020XJG11, 2019XJZ10), Innovation team of Qinghai Nationalities University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Golub, G.H.; Loan, C.F.V. *Matrix Computations*, 4th ed.; The John Hopkins University Press: Baltimore, MD, USA, 2013.
2. Huffel, S.V.; Vandwalle, J. The total least squares problem, computational aspects and analysis. In *Frontiers in Applied Mathematics*; SIAM: Philadelphia, PA, USA, 1991; Volume 9.
3. Björck, Å. Stability of two direct methods for bidiagonalization and partial least squares. *SIAM J. Matrix Anal. Appl.* **2014**, *35*, 279–291.
4. Higham, N.J. *Accuracy and Stability of Numerical Algorithms*, 2nd ed.; SIAM Press: Philadelphia, PA, USA, 2002.
5. Paige, C.C. The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices. Ph.D. Thesis, University of London, London, UK, 1971.
6. Paige, C.C. Computational variants of the Lanczos method for the eigenproblem. *J. Inst. Math. Appl.* **1972**, *10*, 373–381. [CrossRef]
7. Paige, C.C. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *J. Inst. Math. Appl.* **1976**, *18*, 341–349. [CrossRef]
8. Parlett, B.N.; Scott, D.S. The Lanczos algorithm with selective orthogonalization. *Math. Comput.* **1979**, *33*, 217–238. [CrossRef]
9. Ikramov, K.D. Sparse matrices. *Itogi Nauki Tekh. Mat. Anal.* **1982**, *20*, 189–259. (In Russian)
10. Parlett, B.N. *The Symmetric Eigenvalue Problem*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1980.
11. Björck, Å. *Numerical Methods for Least Squares Problems*; SIAM: Philadelphia, PA, USA, 1996.
12. Walker, H.F. Implementation of the GMRES method using Householder transformations. *SIAM J. Sci. Statist. Comput.* **1988**, *9*, 152–163. [CrossRef]
13. Wold, H. Estimation of principal components and related models by iterative least squares. In *Multivariate Analysis*; Krishnaiah, P.R., Ed.; Academic Press: New York, NY, USA, 1966; pp. 391–420.
14. Paige, C.C.; Saunders, M.A. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software* **1982**, *8*, 43–71. [CrossRef]
15. Wilkinson, J.H. *The Algebraic Eigenvalue Problem*; Oxford University Press: London, UK, 1965.
16. Lawson, C.; Hanson, R. *Solving Least Squares Problems*; Prentice Hall: Englewood Cliffs, NJ, USA, 1974.
17. Xie, P.; Xiang, H.; Wei, Y. A contribution to perturbation analysis for total least squares problems. *Numer. Algorithm.* **2017**, *75*, 381–395. [CrossRef]
18. Paige, C.C.; Saunders, M.A. Towards a generalized singular value decomposition. *SIAM J. Numer. Anal.* **1981**, *18*, 398–405. [CrossRef]
19. Demmel, J.; Kahan, W. Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Statist. Comput.* **1990**, *11*, 873–912. [CrossRef]
20. Fierro, R.D.; Bunch, J.R. Perturbation theory for orthogonal projection methods with application to least squares and total least squares. *Linear Algebra Appl.* **1996**, *234*, 71–96. [CrossRef]