Received: 3 August 2017 | Revised: 26 August 2018 | Accepted: 10 September 2018

DOI: 10.1002/nla.2219

RESEARCH ARTICLE

WILEY

Randomized algorithms for total least squares problems

Pengpeng Xie[1] | Hua Xiang[2] | Yimin Wei[3]

[1]School of Mathematical Sciences, Ocean University of China, Qingdao 266100, China

[2]School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

[3]Key Laboratory of Mathematics for Nonlinear Sciences, Fudan University, Shanghai 200433, China

Correspondence
Hua Xiang, School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China.
Email: hxiang@whu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 11271084, 11801534, 11701538, 11571265, 11471253 and 11771099; Fundamental Research Funds for the Central Universities, Grant/Award Number: 201562012; NSFC-RGC(China-Hong Kong), Grant/Award Number: 11661161017; International Cooperation Project of Shanghai Municipal Science and Technology Commission, Grant/Award Number: 16510711200

AMS Classification: 15A09, 65F35

Summary

Motivated by the recently popular probabilistic methods for low-rank approximations and randomized algorithms for the least squares problems, we develop randomized algorithms for the total least squares problem with a single right-hand side. We present the Nyström method for the medium-sized problems. For the large-scale and ill-conditioned cases, we introduce the randomized truncated total least squares with the known or estimated rank as the regularization parameter. We analyze the accuracy of the algorithm randomized truncated total least squares and perform numerical experiments to demonstrate the efficiency of our randomized algorithms. The randomized algorithms can greatly reduce the computational time and still maintain good accuracy with very high probability.

KEYWORDS

Golub–Kahan bidiagonalization, Lanczos tridiagonalization, Nyström method, randomized algorithms, singular value decomposition, total least squares, truncated total least squares

1 | INTRODUCTION

Given an overdetermined set of m linear equations $Ax \approx b$ in n unknowns x, the total least squares (TLS) problem can be formulated as[1]

$$\{x_{\mathrm{TLS}},\ E_{\mathrm{TLS}},\ f_{\mathrm{TLS}}\} := \arg\min_{x,E,f} \|[E\ f]\|_F \quad \text{s.t.} \quad (A+E)x = b+f, \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm.

The term "total least squares" was coined in the work of Golub and Van Loan.[2] It has also been known as the errors-in-variables model or orthogonal regression in the statistical literature. When the coefficient matrix A is affected by the sampling, modeling, or measurement errors, the TLS method is more realistic, whereas the underlying assumption in the least squares (LS) problem is that the error only occurs in the right-hand side vector b. If the errors in the observation matrix A and observation vector b are independent random variables with zero mean and equal variance, TLS gives better estimates than LS.[1(p5)] The algebraic relations between the TLS and LS problems can be found in the works of Van Huffel and Vandewalle,[1] Golub and Van Loan,[2] and Wei.[3] There are many applications mentioned in the monograph of Van Huffel and Vandewalle,[1] especially those arising in signal processing, geophysics, etc. TLS is also one of the

methods for blind deconvolution in image deblurring.[4] The book edited by Van Huffel[5] focuses more on applications. The applications and theory associated with the TLS are still being studied (see, for example, the works of Hansen et al.[6] and Lee et al.[7]).

We concentrate on the TLS problem $Ax \approx b$ with a single right-hand side in this paper. For the TLS problems with multiple right-hand sides, we refer the reader to other works.[8–11] Besides, a generalization to the so-called nongeneric problems and the multiple right-hand side problems is shown in the work of Van Huffel and Vandewalle.[1] For the numerical solution of the TLS problem, a simple solver[1] based on the singular value decomposition (SVD) of the augmented matrix $[A,\ b]$ can be used. When A is large, a complete SVD will be very costly. One improvement is to compute a partial SVD (PSVD) based on Householder transformation[12] or Lanczos bidiagonalization[13–15] introduced by Golub and Kahan.[16] For more on Golub–Kahan bidiagonalization, we recommend the work of Liesen and Strakoš.[17]

For the large-scale TLS problem with a very ill-conditioned coefficient matrix whose singular values decay gradually, the task is even more challenging. In this paper, we consider another variant based on randomized algorithms. The randomized algorithms open the possibility of dealing with large-scale problems and have been receiving increasing attention in the matrix approximation in the last decade.[18] Compared with standard deterministic ones, randomized algorithms are often faster and perhaps surprisingly more robust. As pointed out in the work of Halko et al.,[18] randomized methods have at least two powerful advantages over Krylov methods: inherent stability without depending on the subtle spectral properties of the input matrix and the matrix–vector multiplication's compatibility with the computational platform such as parallel and distributed machines.

Drineas et al.[19] introduced the randomized Hadamard transformation for LS problems. Rokhlin and Tygert[20] applied a randomized transform named SRFT for overdetermined systems. Avron et al.[21] derived a randomized LS solver BLENDENPIK, which outperforms LAPACK by large factors for dense highly overdetermined systems. An algorithm based on random normal projections was introduced in the work of Coakley et al.[22] A parallel iterative LS solver LSRN is also based on random normal projections and can be found in the work of Meng et al.[23] Approximate factorizations derived from randomization in the aforementioned works[19–23] are used to precondition the iterative methods. Such algorithms differ from that in the work of Halko et al.,[18] in that the approximation error from randomization only influences the rate of convergence of the iterative method to the exact solution. Xiang and Zou[24,25] and Wei et al.[26] used randomized strategies in the work of Halko et al.[18] for the regularized LS solutions of large-scale discrete inverse problems. Such randomized algorithms can greatly reduce the computational time and still keep good accuracy with very high probability. However, the corresponding randomized algorithms for the TLS problems are still lacking. Motivated by these randomized matrix algorithms, we present randomized algorithms for the solution of TLS problems, including the well-conditioned and ill-conditioned cases. Numerical experiments and detailed error analysis show that these random sampling techniques can be quite effective and more efficient than the classical competitors in many aspects.

Throughout this paper, $\mathbb{R}^{m\times n}$ denotes the set of $m \times n$ matrices with real entries, and I_n stands for the identity matrix with order n. As usual, $\mathbf{0}$ denotes the zero matrix with the corresponding size easily known from the context. For a matrix $A \in \mathbb{R}^{m\times n}$, A^{T} is the transpose of A, and $\|A\|_2$ and $\|A\|_\infty$ denote the spectral norm and the infinity norm of A, respectively. The symbol $A^\dagger$ represents the Moore–Penrose inverse of A,[27] and $\mathrm{Cond}(A)$ denotes the spectral condition number of A. The remaining sections of this paper are organized as follows. Section 2 introduces some basic results. In Section 3, we present the randomized algorithms. In Section 4, we turn to the algorithm analysis. The numerical results are presented in Section 5, and conclusions are made in Section 6.

2 | PRELIMINARIES

Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ with $m \geq n$. Let $[A, b]$ and A have SVDs, respectively,

$$U^{\mathrm{T}}[A,\ b]V = \mathrm{diag}(\sigma_1, \sigma_2, \ \dots, \sigma_t) = \Sigma,$$
$$\overline{U}^{\mathrm{T}} A \overline{V} = \mathrm{diag}(\overline{\sigma}_1, \overline{\sigma}_2, \ \dots, \overline{\sigma}_n),$$

where $t = \min\{m, n+1\}$, and for the case $m > n$, orthonormal matrix $U \in \mathbb{R}^{m\times(n+1)}$, orthogonal matrix $V \in \mathbb{R}^{(n+1)\times(n+1)}$, and diagonal matrix $\Sigma \in \mathbb{R}^{(n+1)\times(n+1)}$ are partitioned as follows:

$$U = [U_1, u_{n+1}], \quad V = \begin{bmatrix} V_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \sigma_{n+1} \end{bmatrix}, \tag{2}$$

where $u_{n+1} \in \mathbb{R}^m,\ v_{12} \in \mathbb{R}^n,\ v_{21} \in \mathbb{R}^{1\times n}$. Setting $C = [A,\ b]$, the eigenvalue decomposition of the positive semidefinite matrix $C^{\mathrm{T}}C$ is then $C^{\mathrm{T}}C = V\Sigma^{\mathrm{T}}\Sigma V^{\mathrm{T}}$, which will be used in later algorithms. We do not consider the case $m = n$ in the following. The genericity condition

$$\overline{\sigma}_n > \sigma_{n+1} \tag{3}$$

ensures the existence and uniqueness of the TLS solution (see the work of Golub and Van Loan[2]). According to the so-called core problem concept,[28] there exist orthogonal matrices P and Q such that

$$P^{\mathrm{T}}[b, A]\begin{bmatrix} 1 & 0 \\ 0 & Q \end{bmatrix} = \begin{bmatrix} b_1 & A_{11} & 0 \\ 0 & 0 & A_{22} \end{bmatrix},$$

and then, (3) is equivalent to $\sigma_{\min}(A_{22}) > \sigma_{\min}([b_1, A_{11}]) = \sigma_{n+1}$.[8]

From best rank-1 approximation[27] of matrix $[A,\ b]$, we know that

$$[E,\ f] = -U\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_{n+1} \end{bmatrix} V^{\mathrm{T}} = -\sigma_{n+1}u_{n+1}\begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix}^{\mathrm{T}} = -\sigma_{n+1}u_{n+1}v_{n+1}^{\mathrm{T}},$$

where $v_{n+1} = [v_{12}^{\mathrm{T}},\ v_{22}]^{\mathrm{T}}$. Therefore, we have the TLS solution

$$x_{\mathrm{TLS}} = -\frac{v_{12}}{v_{22}}, \quad \begin{bmatrix} x_{\mathrm{TLS}} \\ -1 \end{bmatrix} = -\frac{v_{n+1}}{v_{22}}. \tag{4}$$

It follows from theorem 2.7 in the work of Van Huffel and Vandewalle[1] that the solution x_{TLS} can also be expressed as a function of $[A,\ b]$, that is,

$$x_{\mathrm{TLS}} = \left(A^{\mathrm{T}}A - \sigma_{n+1}^2 I\right)^{-1}A^{\mathrm{T}}b. \tag{5}$$

Now, consider the TLS problem (1) and assume that

$$\overline{\sigma}_q > \sigma_{q+1} = \cdots = \sigma_{n+1} \tag{6}$$

with $q \le n$. In this case, we partition V as follows:

$$V = \begin{bmatrix} \mathbf{V_{11}} & \mathbf{V_{12}} \\ \mathbf{v_{21}} & \mathbf{v_{22}} \end{bmatrix} \begin{matrix} n \\ 1 \end{matrix}. \qquad \begin{matrix} q & n+1-q \end{matrix} \tag{7}$$

Here, $\mathbf{v_{22}}$ is a row vector corresponding to the multiple minimum singular value σ_{q+1}. Condition (6) is equivalent to that $\sigma_q > \sigma_{q+1} = \cdots = \sigma_{n+1}$, and $\mathbf{v_{22}}$ is of full row rank, that is, $\mathbf{v_{22}}$ is not a zero vector.[1] In this circumstance, the classical TLS solution exists, but we have more than one solution. Using the block partitioning of V, we can choose the minimum norm one as in theorem 3.7 in the work of Van Huffel and Vandewalle.[1]

We can determine the parameter q such that the last $n + 1 - q$ ($q < n$) singular values of $[A,\ b]$ are quite small, but not exactly equal as (6). Then, we can neglect these small singular values and truncate out the whole block of V correspondingly and then get the so-called truncated total least squares (TTLS) solution. The traditional TTLS solution is given by the following algorithm (see section 3.6.1 in the work of Van Huffel and Vandewalle[1]).

Algorithm TTLS: Classical TTLS for $q < n$

Inputs: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $q \in \mathbb{N}$.

Output: $x_{\mathrm{TTLS}} \in \mathbb{R}^n$.

1: Compute the SVD: $[A,\ b] = U\Sigma V^{\mathrm{T}}$, and partition the matrix V as in (7).

2: Form the minimum-norm TLS solution: $x_{\mathrm{TTLS}} = -\mathbf{V_{12}}\mathbf{v_{22}}^{\dagger}$.

In this algorithm, the truncation parameter q is user specified or determined adaptively.[13] It is chosen such that the first q largest singular values dominate and $\|\mathbf{v_{22}}\|_2 \neq 0$. Here, the Moore–Penrose inverse $\mathbf{v_{22}}^{\dagger} = \mathbf{v_{22}}^{\mathrm{T}}\|\mathbf{v_{22}}\|_2^{-2}$. According to theorem 3.10 in the work of Van Huffel and Vandewalle,[1(p62)] the minimum-norm TLS solution x_{TTLS} under condition (6) can be also expressed by $x_{\mathrm{TTLS}} = (\mathbf{V_{11}}^{\mathrm{T}})^{\dagger}\mathbf{v_{21}}^{\mathrm{T}}$. For the case where $q = n$ and the genericity condition (3) holds, this algorithm reduces to the classical TLS. For the case where $q < n$ and condition (6) holds without adopting the truncation, this algorithm reduces to the TLS after the formula in step 2 is replaced by the formula in theorem 3.7 in the work of Van Huffel and Vandewalle.[1]

The idea of TTLS is to treat the small singular values of the augmented matrix $[A,\ b]$ as identical and compute the above formula, which is, in fact, the minimum-norm solution of a nearby system if $\sigma_{q+1}, \ldots, \sigma_{n+1}$ do not coincide.[1(p66)] For the discrete ill-posed problems where the singular values of the coefficient matrices decay gradually, TTLS can be applied, where parameter q then plays the role of the regularization parameter. In practical applications, the smallest singular values of $[A,\ b]$ rarely coincide.[29] However, if one considers the TLS problem as an approximation to the corresponding unobservable exact relation $A_0x = b_0$ with $\text{rank}([A_0,\ b_0]) = \text{rank}\,(A_0) = q \leq n$, then $\sigma_{q+1}, \ldots, \sigma_{n+1}$ are just the perturbations of zero.[1,29] In this case, it is realistic to define an error bound ϵ such that all singular values σ_i, satisfying $|\sigma_i - \sigma_{n+1}| < \epsilon$, are considered to coincide with σ_{n+1}. Therefore, we can use the formula $x_{\text{TTLS}} = -\mathbf{V_{12}}\mathbf{V_{22}}^{\dagger}$.

3 | RANDOMIZED ALGORITHMS

There are already many efficient algorithms, as discussed in the Introduction. In the following, we will investigate randomized algorithms for the solution of the TLS problem, especially for the large-scale case. The first one is the randomized algorithm for TLS based on the Nyström technique (NTLS for short). The second one is the randomized algorithm for TTLS (RTTLS for short). The randomized algorithms are effective for the problems that can be well approximated by low-rank matrices. These can be used for the cases where the numerical rank is known or can be estimated in advance. The randomized algorithms are inherently stable and independent of the spectral properties of the input matrix and can take full advantage of the computational platforms for parallel computing.[18] Therefore, they can greatly reduce the computational time and still yield good approximate solutions.

Except the randomized algorithms for the TLS and TTLS solutions, one can also design certain deterministic algorithms to reduce the computational costs. We will introduce some for the comparison with the randomized algorithms. One strategy is to improve the efficiency by computing the SVD of $[A,\ b]$ in step 1 "partially." Correspondingly, the algorithm named partial total least squares (PTLS) is given in the work of Van Huffel and Vandewalle.[1(p118)] The main difference between TTLS and PTLS lies in the first step: One uses the classical complete SVD, whereas the other applies the PSVD.[1(p106)] Van Huffel and Vandewalle[1] reported that PTLS is two times faster than TTLS while the same accuracy can be maintained. Moreover, the relative efficiency of PSVD increases when the dimension of the desired singular subspace is relatively smaller than the dimension n. However, this approach will lose the sparsity or structure of the coefficient matrix in the first step of bidiagonal reduction for large-scale sparse or structured matrices (e.g., Toeplitz matrices arising from the discretization of many convolution problems), since the initial reduction of $[A,\ b]$ to bidiagonal form in PSVD by Householder transformation will destroy the sparsity or structure of the matrix in the first step of reduction (see section 4 in the work of Fierro et al.[13]). An alternative approach LTTLS based on the Lanczos procedure will be considered later.

3.1 | Medium-sized problem

The essential step of this traditional algorithm TLS is the SVD of $C = [A,\ b]$. However, SVD can be very costly, or even prohibitive. How to reduce the computational cost and still ensure the accuracy of the approximate solution is our main concern. In the following, we introduce the Nyström method (NTLS) for computing the TLS solution, together with the Lanczos method (LTLS) for comparison.

3.1.1 | NTLS

From expression (4), we know that the key point is to find the right singular vector associated with the smallest singular value. However, the randomized SVD[18] usually approximates well the largest singular values and the corresponding singular vectors. Denote $B = (C^{\mathrm{T}}C)^{-1}$. Using the SVD above, we have $B = V\,\text{diag}(\sigma_{n+1}^{-2}, \ldots, \sigma_1^{-2})V^{\mathrm{T}}$. Hence, we can see that σ_{n+1}^{-2} becomes the largest diagonal element, and the corresponding singular vector v_{n+1} is what we need. Therefore, we can apply the randomized algorithm to approximate this value and determine its corresponding approximate right singular vector v_{n+1}. Since matrix B is symmetric positive definite and the Nyström method is known to be much more accurate for the positive semidefinite matrices,[18,30] we present a randomized algorithm that applies the Nyström technique for computing an approximate eigenvalue decomposition (NTLS for short).

Algorithm NTLS: TLS solution via Nyström scheme

Inputs: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and l ($\ll n$).

Output: $x_{\text{NTLS}} \in \mathbb{R}^n$.

1: Solve $(C^{\mathrm{T}}C)X = \Omega$, where $C = [A,\ b] \in \mathbb{R}^{m\times(n+1)}$, and Ω is an $(n+1)\times l$ Gaussian random matrix.
2: Compute the $(n+1)\times l$ orthonormal matrix Q via QR factorization $X = QR$.
3: Solve $(C^{\mathrm{T}}C)Y = Q$, and form the $l\times l$ matrix $Z = Q^{\mathrm{T}}Y$.
4: Perform the Cholesky factorization $Z = G^{\mathrm{T}}G$ and seek K by solving $KG = Y$.
5: Compute the SVD: $K = V\Sigma U^{\mathrm{T}}$, and form the solution $x_{\text{NTLS}} = -v(1:n)/v(n+1)$, where $v = V(:,1)$.

The Nyström technique in NTLS develops a rank-l approximation as

$$\begin{aligned} B &\approx (BQ)(Q^{\mathrm{T}}BQ)^{-1}(BQ)^{\mathrm{T}} = \left[(BQ)(Q^{\mathrm{T}}BQ)^{-1/2}\right]\left[(BQ)(Q^{\mathrm{T}}BQ)^{-1/2}\right]^{\mathrm{T}} \\ &= KK^{\mathrm{T}}. \end{aligned}$$

In the algorithm NTLS, matrix K approximates the Cholesky factor of B. According to lemma 4 in the work of Drineas and Mahoney,[30] the Nyström approximation error is smaller than $\|B - QQ^{\mathrm{T}}B\|_2$. For more details, see the work of Drineas and Mahoney[30] and the references therein, as well as algorithm 5.5 in the work of Halko et al.[18]

Note that l is a prespecified parameter. In the work of Halko et al.,[18] the index l is usually selected in the form $l = k + p$, where p is an oversampling parameter, and k corresponds to the rank k specified in advance for the best rank-k approximation of A. In step 1, we obtain $X = B\Omega$ to extract the column information, which is further represented by an orthonormal matrix Q in step 2. After step 3, the problem is reduced to a smaller symmetric positive definite matrix $Z = Q^{\mathrm{T}}BQ$. In fact, instead of step 4 and step 5 in algorithm NTLS, here, eigenvalue decomposition can be applied to this small matrix $Z = W\Sigma W^{\mathrm{T}}$, where W is orthogonal, and $W(:,1)$ is the eigenvector corresponding to the largest eigenvalue. This leads to an approximation

$$B \approx QQ^{\mathrm{T}}BQQ^{\mathrm{T}} = QZQ^{\mathrm{T}} = J\Sigma J^{\mathrm{T}},$$

where $J = QW$. The first column of J is the approximate eigenvector of B corresponding to σ_{n+1}^{-2} and hence approximates the right singular vector corresponding to the smallest singular value σ_{n+1}. We then use this approximation to seek the approximate TLS solution. This procedure is based on algorithm 5.3 in the work of Halko et al.[18] However, as pointed out in the aforementioned work,[18] the Nyström scheme can be used to improve the quality of standard factorizations with no additional cost when the input matrix B is symmetric positive definite. Therefore, we first decompose $Q^{\mathrm{T}}BQ$ into its Cholesky factors and then compute the factor $K = YG^{-1}$ in step 4. The approximate solution can be formed finally in step 5 by computing the SVD of K.

The linear system involving $C^{\mathrm{T}}C$ in steps 1 and 3 can be solved by direct methods or subspace iterative methods depending on the conditioning of $C^{\mathrm{T}}C$. When the problem is well conditioned, it can be solved quite efficiently. Using a preceding QR factorization, that is, $C = QR$, where Q is an orthonormal matrix and R is an upper triangular matrix, we have $C^{\mathrm{T}}C = R^{\mathrm{T}}R$ and $B = (C^{\mathrm{T}}C)^{-1} = R^{-1}R^{-\mathrm{T}}$, then the linear systems can be solved by simple forward or backward substitutions. In practice, there exist cases where matrix A is, in fact, implicit. Such cases may arise in optimization algorithms in large-scale partial differential equation–constrained inverse problems. Concretely speaking, in steps 1 and 3, we can solve the corresponding system by Krylov subspace methods, where we just need to process the multiplications of $C^{\mathrm{T}}C$ with some vectors.

3.1.2 | LTLS

To compare NTLS with the classical method, a convincing way is to consider the symmetric Lanczos method applied to the inverse of C^TC, that is, $B = (C^{\mathrm{T}}C)^{-1}$. This yields that

$$BQ_l = Q_lT_l + \beta_{l+1}q_{l+1}e_l^{\mathrm{T}}, \tag{8}$$

where $Q_l = [q_1, \ldots, q_l]$ satisfying $Q_l^{\mathrm{T}}Q_l = I$ and T_l is the symmetric tridiagonal matrix given by

$$T_l = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_l \\ & & \beta_l & \alpha_l \end{bmatrix}.$$

For the implementation of the symmetric Lanczos process, we refer to the work of Golub and Van Loan[27] and omit a detailed description here. A typical step is as follows:

$$\beta_{k+1}q_{k+1} = Bq_k - \alpha_k q_k - \beta_k q_{k-1},$$

where $k = 1, \ldots, l$ with $q_0 = 0$ and $\beta_1 = 0$. Note that in each Lanczos iteration, we need to solve a linear system with coefficient matrix $C^{\mathrm{T}}C$. If we perform the QR factorization on C as a preprocessing step to obtain the R-factor, then we just need to solve a linear equation involving $R^{\mathrm{T}}R$ instead.

Suppose that T_l has the eigendecomposition $T_l = W\Sigma W^{\mathrm{T}}$, where W is orthogonal and Σ is diagonal. We then have

$$B \approx Q_l T_l Q_l^{\mathrm{T}} = Q_l W \Sigma (Q_l W)^{\mathrm{T}}.$$

Let the first column $W(:, 1)$ be the eigenvector of T_l associated to its largest eigenvalue. Then, $Q_l W(:, 1)$ is the approximation of the right singular vector associated to the smallest singular value of C. That is to say, we need to seek the largest eigenvalue of T_l and the corresponding eigenvector by solving

$$T_l\, y = \mu\, y. \tag{9}$$

Then, the Ritz vector $Q_l y$ is what we need. The above process is summarized in the algorithm LTLS. Due to the presence of rounding errors, the orthogonality of Lanczos vectors can be lost unless reorthogonalization is used. One can restore the orthogonality of the generated Lanczos vectors by some extra work. For example, for full reorthogonalization, the Gram–Schmidt orthogonalization process is used, and "twice is enough." Also, selective orthogonalization can be applied, which has nearly the high accuracy as full reorthogonalization with nearly the low cost as no reorthogonalization (see section 7.5 in the work of Demmel[31]).

Algorithm LTLS: TLS solution via symmetric Lanczos

Inputs: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $l\ (\ll n)$.

Output: $x_{\mathrm{LTLS}} \in \mathbb{R}^n$.

1: Apply l-step symmetric Lanczos and obtain Q_l and T_l in (8).
2: Solve the eigenproblem (9) and find the eigenvector y associated with the largest eigenvalue.
3: Compute the Ritz vectotr $v = Q_l y$, and form the solution $x_{\mathrm{LTLS}} = -v(1 : n)/v(n + 1)$.

3.2 | Large-scale problem

The algorithms NTLS and LTLS involve the computation associated with $(C^{\mathrm{T}}C)^{-1}$. This is costly or even prohibitive for the large-scale problems to implement. What is more, the gap $\overline{\sigma}_n - \sigma_{n+1}$ can be small, which makes the TLS problem very ill-conditioned. We need to reduce the floating operations (flops) and use regularization techniques to avoid noise contaminations to obtain a meaningful approximate solution. For the ill-posed problems, there already exist several regularization strategies of the TLS solution. For example, the solution can be stabilized by truncating small singular values of $[A,\ b]$ via an iterative algorithm based on Golub–Kahan bidiagonalization.[13] The technique of TTLS described in the work of Fierro et al.[13] is similar in spirit to the truncated SVD (TSVD), where the small singular values of $[A,\ b]$ are treated as zeros, and the problem is reduced to an exactly rank-deficient one. The Tikhonov regularization strategy is used in other works,[7,32–35] where Cholesky decomposition is computed in each step in the work of Beck and Ben-Tal,[32] and the linear systems are projected onto the Krylov subspace of much smaller dimensions to reduce the problem size in the work of Lampe and Voss.[34] Regularization by an additional quadratic constraint is another choice,[36–39] which is the regularized TLS based on quadratic eigenvalue problems (QEP): adding a quadratic constraint to the TLS and then iteratively solving the QEP. Regularization properties of the TTLS with multiple observations can be found in the work of Hnětynková et al.[40] Recently, an analysis of the sensitivity and conditioning of TTLS has been given in the work of Gratton et al.[41] Some applications of the TTLS are reported[42]: astronomy, chemometrics, computer algebra, computer vision, image reconstruction, linear system theory, machine learning, modal and spectral analysis, speech and audio processing, system identification, etc.

In this subsection, for the large-scale cases, we propose the randomized truncated TLS (RTTLS) with the known or estimated rank as the regularization parameter. To compare with the classical deterministic method, we also introduce the PSVD based on the Golub–Kahan bidiagonalization process for the TTLS solution (LTTLS).

3.2.1 | RTTLS

In the following, we introduce a randomized algorithm to approximate the SVD (see the algorithm RTTLS), which will reduce the original large-scale data into a small one, and hence, the original intractable problem can be resolved. What is more, in this new randomized algorithm, most flops are spent on the matrix–matrix multiplications, which are the so-called BLAS-3 operations, and the algorithm can be realized by accessing the original large-scale matrix A only twice.

Algorithm RTTLS: Randomized algorithm for TTLS

Inputs: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $k < l\ (\ll n)$.

Output: $x_{\mathrm{RTTLS}} \in \mathbb{R}^n$.

1: Form the $m \times l$ matrix $Y = C\Omega$, where $C = [A,\ b]$, and Ω is an $(n+1)\times l$ Gaussian random matrix.
2: Apply QR decomposition to Y, i.e., $Y = QR$, where $Q \in \mathbb{R}^{m\times l}$.
3: Form the $l \times (n+1)$ matrix Z such that $Z = Q^{\mathrm{T}}C$.
4: Compute the SVD $Z = W\Sigma V^{\mathrm{T}}$, where $V \in \mathbb{R}^{(n+1)\times l}$.
5: Let $\mathbf{V_{11}} = V(1{:}n, 1{:}k)$, $\mathbf{v_{21}} = V(\mathrm{end}, 1{:}k)$, and form the solution $x_{\mathrm{RTTLS}} = \left(\mathbf{V_{11}^{\mathrm{T}}}\right)^{\dagger}\mathbf{v_{21}^{\mathrm{T}}}$.

The algorithm RTTLS can be efficiently implemented provided that the action of A and A^{T} on vectors is known. In fact, $Y = C\Omega$ in step 1 can be rewritten as $Y = A\Omega_1 + b\omega_2^{\mathrm{T}}$, where $[\Omega_1^{\mathrm{T}}, \omega_2]^{\mathrm{T}} = \Omega$ with ω_2^{T} being the last row vector. In step 3, we have $Z = Q^{\mathrm{T}}C = [Q^{\mathrm{T}}A, Q^{\mathrm{T}}b]$. Note that Ω and Q just have l columns. For some special cases, the operations $A\Omega_1$ and $A^{\mathrm{T}}Q$ can be performed fast. For example, if A is a Toeplitz matrix, then the fast Fourier transform can be applied.

After step 2, the orthonormal columns of Q approximately span the range of C. Step 3 is used to form the Z-part in the projection $QQ^{\mathrm{T}}C = QZ$, and then, its SVD is performed in step 4. After step 4, we, in fact, have

$$C \approx QQ^{\mathrm{T}}C = (QW)\Sigma V^{\mathrm{T}}.$$

Usually, the randomized algorithm cannot approximate the small singular values very well; hence, we do not prefer to use the expression $x_{\mathrm{TTLS}} = -\mathbf{V_{12}}\mathbf{v_{22}}^{\dagger}$ directly as the algorithm TTLS. Since, in the algorithm RTTLS, we can obtain a good approximation of the right singular vectors associated with the largest singular values, we use $x_{\mathrm{RTTLS}} = (\mathbf{V_{11}^{\mathrm{T}}})^{\dagger}\mathbf{v_{21}^{\mathrm{T}}}$ in step 5.

In the algorithm RTTLS, parameter l stands for the number of sampling, and the number k is the parameter for truncating ($k \le l$). A larger l will improve the reliability of the algorithm[18] but also increase the computational complexity. In practice, we choose $l \ll n$ and make a balance between reliability and computational complexity. Here, the truncation parameter k acts as the regularization parameter. It can be user specified or determined by some regularization technique if no a priori estimate is known. Without regularization, the ordinary least squares or TLS solvers yield physically meaningless solutions for discrete ill-posed problems.

For the large-scale discrete ill-conditioned problem, the choice of the regularization parameter based on a complete SVD is prohibitive. However, the SVD of a matrix can be well approximated by the randomized SVD,[18] and the regularization parameter can also be located by randomized algorithms.[24,25] Here, we obtain an estimation for this parameter by the randomized regularization techniques introduced in the work of Xiang and Zou.[24] We first perform randomized algorithms to obtain an approximate SVD of A, then a GCV function based on this approximation is used to determine the truncation parameter k for the TSVD solution of $Ax \approx b$. This procedure can be performed very fast.[24] This parameter cannot be optimal for the TLS based on the SVD of the augmented matrix $[A,\ b]$, but should be a reasonable estimate for the truncation parameter in TTLS. Other rules such as the L-curve, quasi-optimality, and the discrepancy principle can be also used for the regularization parameter choice. The detailed discussion about some important issues, such as the regularization parameter choice and the scaling of A and b (see section 3.6.2 in the work of Van Huffel and Vandewalle[1]) is beyond the scope of this paper.

The randomized algorithms discussed above are used to solve the fixed-rank problems.[18] In practical applications, the target rank is sometimes not known in advance. We do not need to determine it accurately. In the adaptive approach in section 4.4 in the work of Halko et al.,[18] the number of samples is increased until the desired tolerance is achieved. We will consider this adaptive approach in a future paper.

3.2.2 | LTTLS

Methods based on Golub–Kahan bidiagonalization can be used to compute good approximations to the singular triplets associated with several largest singular values of a matrix (see the works of O'Leary and Simmons[14] and Björck[15]). In the work of Fierro et al.,[13] the authors proposed the method based on Golub–Kahan bidiagonalization for computing TTLS

solutions. Golub–Kahan bidiagonalization is applied to matrix A rather than the augmented matrix $[A,\ b]$. However, a fairly straightforward approach would be to choose some integer l and perform l Lanczos iterations applied to the matrix $C = [A,\ b]$. In the following, we will use this approach to find a PSVD approximation of C, which will be used to define a TTLS solution. We denote this algorithm as LTTLS.

The Golub–Kahan bidiagonalization process can start with a random vector u_1 and produce two sets of orthonormal vectors U_{l+1} and V_l and a lower bidiagonal matrix $\underline{B}_l$ with the following definitions:

$$U_{l+1} = (u_1, \ldots, u_{l+1}) \in \mathbb{R}^{m\times(l+1)}, \qquad V_l = (v_1, \ldots, v_l) \in \mathbb{R}^{(n+1)\times l}, \qquad \underline{B}_l = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_l \\ & & & \beta_{l+1} \end{bmatrix} \in \mathbb{R}^{(l+1)\times l}.$$

We can write the Golub–Kahan bidiagonalization procedure in a compact matrix form as

$$CV_l = U_{l+1}\underline{B}_l, \qquad C^{\mathrm{T}}U_{l+1} = V_l\underline{B}_l^{\mathrm{T}} + \alpha_{l+1}v_{l+1}e_{l+1}^{\mathrm{T}}. \tag{10}$$

Then, we can easily check that

$$U_{l+1}^{\mathrm{T}}CV_l = \underline{B}_l.$$

Suppose that we have the SVD: $\underline{B}_l = U\Sigma W^{\mathrm{T}}$, where $U^{\mathrm{T}}U = I$ and $W^{\mathrm{T}}W = I$. We then have

$$C \approx U_{l+1}U\Sigma(V_lW)^{\mathrm{T}}. \tag{11}$$

We regard (11) as the PSVD of C. Suppose that C has k largest singular values and the others are small. Define $V \equiv V_lW \in \mathbb{R}^{(n+1)\times l}$ and partition it as follows:

$$V = \begin{array}{c} \\ \left[\begin{array}{cc} \mathbf{V_{11}} & \mathbf{V_{12}} \\ \mathbf{v_{21}} & \mathbf{v_{22}} \end{array}\right] \\ \begin{array}{cc} k & l-k \end{array} \end{array} \begin{array}{l} n \\ 1 \end{array}, \tag{12}$$

where $\mathbf{V_{11}} \in \mathbb{R}^{n\times k}$, $\mathbf{V_{12}} \in \mathbb{R}^{n\times(l-k)}$, $\mathbf{v_{21}} \in \mathbb{R}^{1\times k}$, and $\mathbf{v_{22}} \in \mathbb{R}^{1\times(l-k)}$. It is well known that Golub–Kahan bidiagonalization can be used to compute good approximations to the singular triplets associated with the largest singular values of a matrix. The first k columns of V approximate the right singular vectors corresponding to the k largest singular values. By neglecting the small singular values in (11) derived from the Golub–Kahan bidiagonalization, we obtain a TTLS solution.

Please note that with slight abuse of the notations, it has the same form of the minimum-norm TLS solution x_{TTLS} given under the condition $\overline{\sigma}_k > \sigma_{k+1} = \cdots = \sigma_{n+1}$. The computation procedures are summarized in the algorithm LTTLS.

Algorithm LTTLS: Truncated total least squares based on Golub-Kahan bidiagonalization

Inputs: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $k < l\ (\ll n)$.

Output: $x_{\mathrm{LTTLS}} \in \mathbb{R}^n$.

1: Use Golub-Kahan bidiagonalization to obtain (10).
2: Compute the SVD: $\underline{B}_l = U\Sigma W^T$, and set $V = V_lW$.
3: Partition the matrix V as (12), and form the truncated TLS solution $x_{\mathrm{LTTLS}} = \left(\mathbf{V}_{\mathbf{11}}^{\mathrm{T}}\right)^{\dagger}\mathbf{v}_{\mathbf{21}}^{\mathrm{T}}$.

The Lanczos procedure is an efficient method for the SVD approximation. It does not alter matrix A. Hence, the sparsity can be still kept. Golub–Kahan bidiagonalization mainly uses the BLAS-2 operations, that is, the matrix–vector multiplications, and needs to access the coefficient matrix l times, whereas the randomized algorithm RTTLS mainly uses BLAS-3 operations and can be realized by accessing the original large-scale matrix A only twice.

4 | ALGORITHM ANALYSIS

4.1 | Computational complexity

We shall say a few words about the computational complexity of the randomized algorithms. The cost of each step of the algorithm is listed in Table 1. In TLS and TTLS, to obtain the singular values and the right singular vectors of the matrix $C = [A,\ b] \in \mathbb{R}^{m\times(n+1)}$, the flop count of the classical SVD based on R-bidiagonalization (R-SVD) is about $2mn^2 + 11n^3$.[27(p493)]

TABLE 1 Computational complexity (l-step Lanczos procedures in LTLS and LTTLS)

Step	TLS	NTLS	LTLS	TTLS	RTTLS	LTTLS
0		$2mn^2 - \frac{2}{3}n^3$	$2mn^2 - \frac{2}{3}n^3$			
1	$2mn^2 + 11n^3$	$2n^2l$	$2n^2l$	$2mn^2 + 11n^3$	$2mnl$	$4mnl$
2	n	$2nl^2$	$6l^3$	$2n(n-k)$	$2ml^2$	$13l^3$
3		$2n^2l + 2nl^2$	$2nl$		$2mnl$	$2nk^2 + 11k^3$
4		$\frac{1}{3}l^3 + nl^2$			$4n^2l + 13l^3$	
5		$2nl^2 + 11l^3$			$2nk^2 + 11k^3$	

Note that step 0 is the preprocessing step for NTLS and LTLS. Here, we use the Householder QR factorization on C and return the R-factor, which costs about $2mn^2 - \frac{2}{3}n^3$. Then, $(R^TR)X = \Omega$ can be solved by forward and backward substitution, which costs about $2n^2l$. Please also note that such preprocessing based on QR factorization is just one possible choice to facilitate the solution of a linear system involving C^TC. In step 2 of NTLS, the modified Gram–Schmidt is used to get the Q-factor, and it costs $2nl^2$ flops. In step 4 of NTLS, the symmetric QR algorithm is applied on Z, and it costs about $9l^3$ flops; if R-SVD is used, it costs about $13l^3$. In step 4, the Cholesky factorization of Z costs $\frac{1}{3}l^3$, and the linear solver needs nl^2 flops. Finally, the R-SVD of $K \in \mathbb{R}^{(n+1)\times l}$ in step 5 costs about $2nl^2 + 11l^3$.

In LTLS, the main cost of the l-step Lanczos process is the matrix–vector involving $(C^TC)^{-1}$ in each step, which can be achieved by solving a linear system with coefficient matrix R^TR. Hence, the l-step Lanczos process needs about $2n^2l$ flops. If we use full reorthogonalization, where the Gram–Schmidt process is used twice, then the extra work for reorthogonalization is about $4nl^2$. In the numerical tests, we find that the Lanczos without reorthogonalization works well, since we seek the largest eigenvalue and its corresponding eigenvector of $(C^TC)^{-1}$, whereas such extreme eigenvalue usually can be well approximated by the Krylov subspace methods. Besides, in step 2 of LTLS, we need to solve a symmetric tridiagonal eigensystem. The tridiagonal QR algorithm (two steps per eigenvalue assumed) needs an estimated $6l^3$ flops to converge.

In LTTLS, we use the l-step Golub–Kahan bidiagonalization to compute the PSVD in step 1. The main cost per step of Golub–Kahan bidiagonalization comes from the two matrix–vector multiplications involving C and C^T, respectively, and each needs about $2mn$ flops. Hence, l-step costs about $4mnl$ regardless of the sparsity. The SVD in step 2 of LTTLS needs about $13l^3$ flops. In RTTLS, step 4 is equivalent to $Z^T = V\Sigma W^T$, and we need the left singular vectors of $Z^T \in \mathbb{R}^{(n+1)\times l}$, and hence, R-SVD needs about $4n^2l + 13l^3$ flops. In the last step of LTTLS and RTTLS, we have to form the solution by exploiting the singular vectors corresponding to the largest singular values, that is, $(\mathbf{V_{11}^T})^\dagger \mathbf{v_{21}^T}$. The Moore–Penrose inverse via SVD needs about $2nk^2 + 11k^3$.

For the cases where singular values decay rapidly, we can choose a small parameter l. For most cases, $m \gtrsim n \gg l$ and $k \leq l$. Under the condition that $l \ll n \approx m$, we can see from Table 1 that the flop count of TLS and TTLS is $O(n^3)$, and steps 1–5 of NTLS and LTLS count for $O(n^2l)$ flops, whereas their preprocessing step of QR factorization needs about $O(n^3)$ flops, respectively. However, both LTTLS and RTTLS only need $O(n^2l)$ flops. Although the computational cost of LTTLS is of the same order as RTTLS, usually, the randomized algorithms can be essentially faster than the traditional counterpart. The most flops in RTTLS are performed by very efficient BLAS-3 operations, whereas LTTLS just carries out BLAS-2 operations. The advantage of RTTLS can be more obvious than just what the flop account tells.

4.2 | Error estimates

We will analyze the accuracy of the algorithm RTTLS in this part. Before the main results, we introduce an important estimate in the work of Halko et al.[18]

Lemma 1. (See corollary 10.9 in the work of Halko et al[18])
Suppose that $A \in \mathbb{R}^{m\times n}$ has singular values $\sigma_1 \geq \sigma_2 \geq \cdots$. Choose a target rank $k \geq 2$ and an oversampling parameter $p \geq 4$, where $k + p = l \leq \min\{m, n\}$. Draw an $n \times l$ standard Gaussian matrix Ω, and let Q be an orthonormal matrix whose columns form a basis for the range of the sampled matrix $A\Omega$. Then, we have

$$\|A - QQ^TA\|_2 \leq \left(1 + 9\sqrt{l}\sqrt{\min\{m, n\}}\right)\sigma_{k+1},$$

with a failure probability of, at most, $3p^{-p}$.

This error bound is a simplification of the error bound in corollary 10.9 in the work of Halko et al.[18] and given in (1.9) of the aforementioned work.[18] From the process of the algorithm RTTLS, we see that $U\Sigma V^{\mathrm{T}} = QW\Sigma V^{\mathrm{T}} = QQ^{\mathrm{T}}C$, where we denote $U = QW$ and $C = [A,\ b]$. Hence, we have $\|C - U\Sigma V^{\mathrm{T}}\|_2 = \|C - QQ^{\mathrm{T}}C\|_2$. Moreover, we obtain a good SVD approximation for C with high probability, especially when C can be well approximated by a low-rank matrix.

For the perturbation of the TTLS solution, we slightly improve the perturbation bound given in theorem 4.1 in the work of Wei[29] and present it in the following. It will be used in the later analysis of the accuracy of the randomized algorithms.

Lemma 2. *Consider the* TTLS *problem. Let the* SVD*s for A and $[A,\ b]$ be given as in the preliminaries. Assume that for some $q \le n$, $\overline{\sigma}_q > \sigma_{q+1}$. Partition V as in the algorithm* TTLS *and define $\widehat{A} \in \mathbb{R}^{m\times n}$, $\widehat{b} \in \mathbb{R}^m$ by $[\widehat{A},\ \widehat{b}] = [A,\ b]+[\delta A,\ \delta b]$ with $\|[\delta A,\ \delta b]\|_2 \le \frac{1}{6}(\overline{\sigma}_q - \sigma_{q+1})$. Let the* SVD *for $[\widehat{A},\ \widehat{b}]$ be*

$$\widehat{U}^{\mathrm{T}}\ [\widehat{A},\ \widehat{b}]\ \widehat{V}^{\mathrm{T}} = \widehat{\Sigma}.$$

Partition $\widehat{V}$ conformally with V and accordingly define $\widehat{V}_{11},\ \widehat{V}_{12},\ \widehat{v}_{21},\ \widehat{v}_{22}$ such as $V_{11}, V_{12}, v_{21}, v_{22}$ in the algorithm TTLS. *Define $\widehat{x}_{\mathrm{TTLS}} = (\widehat{V}_{11}^{\mathrm{T}})^{\dagger}\widehat{v}_{21}^{\mathrm{T}}$ and $x_{\mathrm{TTLS}} = (V_{11}^{\mathrm{T}})^{\dagger}v_{21}^{\mathrm{T}}$. When $x_{\mathrm{TTLS}} \ne \mathbf{0}$ and $\|b\|_2 > \sigma_{q+1}$, the following estimate holds:*

$$\frac{\left\|x_{\mathrm{TTLS}} - \widehat{x}_{\mathrm{TTLS}}\right\|_2}{\|x_{\mathrm{TTLS}}\|_2} \le \left(\frac{6\sigma_1}{\|b\|_2 - \sigma_{q+1}} + 2\right)\frac{\left(\|[\delta A,\ \delta b]\|_2 + \sigma_{q+1}\right)}{\overline{\sigma}_q - \sigma_{q+1}}.$$

Proof. Denote $[A', b'] = [A, b] - U_2\Sigma_2[V_{12}^{\mathrm{T}}, v_{22}^{\mathrm{T}}]$. Here, U_2 and Σ_2 are the submatrices of U and Σ, which are partitioned conformally with V in the algorithm TTLS. $\widehat{U}$ and $\widehat{\Sigma}$ are partitioned conformally with $\widehat{V}$, and the submatrices are denoted by $\widehat{U}_i$ and $\widehat{\Sigma}_i$ $(i = 1, 2)$. For the perturbed cases, we define similar notations $[\widehat{A}', \widehat{b}'] = [\widehat{A}, \widehat{b}] - \widehat{U}_2\widehat{\Sigma}_2[\widehat{V}_{12}^{\mathrm{T}}, \widehat{v}_{22}^{\mathrm{T}}]$. From the proof of theorem 4.1 in the work of Wei,[29] we know that $x_{\mathrm{TTLS}} - \widehat{x}_{\mathrm{TTLS}}$ can be rewritten as

$$\begin{aligned} x_{\mathrm{TTLS}} - \widehat{x}_{\mathrm{TTLS}} &= (\widehat{A}')^{\dagger}(\widehat{A}' - A')\,x_{\mathrm{TTLS}} + \left(I - (\widehat{A}')^{\dagger}\widehat{A}'\right)x_{\mathrm{TTLS}} + (\widehat{A}')^{\dagger}(b' - \widehat{b}') \\ &= (\widehat{A}')^{\dagger}\left[\widehat{A}' - A', \widehat{b}' - b'\right]\begin{bmatrix} x_{\mathrm{TTLS}} \\ -1 \end{bmatrix} + \left(I - (\widehat{A}')^{\dagger}\widehat{A}'\right)x_{\mathrm{TTLS}}. \end{aligned} \tag{13}$$

Note that

$$\begin{aligned} \left\|\left[\widehat{A}' - A', \widehat{b}' - b'\right]\right\|_2 &\le \|[\delta A,\ \delta b]\|_2 + \left\|U_2\Sigma_2\left[V_{12}^{\mathrm{T}}, V_{22}^{\mathrm{T}}\right]\right\|_2 + \left\|\widehat{U}_2\widehat{\Sigma}_2\left[\widehat{V}_{12}^{\mathrm{T}}, \widehat{V}_{22}^{\mathrm{T}}\right]\right\|_2 \\ &\le 2\left(\|[\delta A,\ \delta b]\|_2 + \sigma_{q+1}\right). \end{aligned}$$

Taking norms on both sides of (13) and using inequalities (4.7) and (4.8) in the work of Wei,[29] we get

$$\left\|x_{\mathrm{TTLS}} - \widehat{x}_{\mathrm{TTLS}}\right\|_2 \le \left(3\sqrt{1 + \|x_{\mathrm{TTLS}}\|_2^2} + 2\|x_{\mathrm{TTLS}}\|_2\right)\frac{\left(\|[\delta A,\ \delta b]\|_2 + \sigma_{q+1}\right)}{\overline{\sigma}_q - \sigma_{q+1}}. \tag{14}$$

The same techniques exploited in the proof of theorem 4.1 in the work of Wei[29] can be applied, and it follows that $\sqrt{1 + \|x_{\mathrm{TTLS}}\|_2^2}/\|x_{\mathrm{TTLS}}\|_2 = 1/\|v_{21}\|_2 \le 2\sigma_1/(\|b\|_2 - \sigma_{q+1})$. Substituting it into (14) yields the conclusion. □

Note that the bound in (14) is smaller than the corresponding bound in theorem 4.1 in the work of Wei.[29] Consequently, our result derived in Lemma 2 is slightly better. The derived perturbation bounds reflect the sensitivity of the problem and can be useful for the analysis of numerical algorithms for the TLS and TTLS solutions. In fact, we can check this bound easily. Consider an example from Hansen's Regularization Tools,[43] say BAART, with the matrix size $n = 1,000$ and $q = 9$. Perturb $[A,\ b]$ with random noise such that $\|[\delta A,\ \delta b]\|_2 = \frac{1}{8}(\overline{\sigma}_q - \sigma_{q+1})$. We obtain that the relative error is 1.4, whereas the upper bound given by Lemma 2 is 1.6.

Using Lemmas 1 and 2, we get the estimate below.

Theorem 1. *Assume $m \ge n + 1$. Assume that $[A,\ b]$ has singular values $\sigma_1, \dots, \sigma_{n+1}$ and A has singular values $\overline{\sigma}_1 \ge \cdots \ge \overline{\sigma}_n$. Moreover, assume $\overline{\sigma}_q > \sigma_{q+1}$ with $q \le n$, and let k be the target rank of $[A,\ b]$ and x_{RTTLS} be the approximate* TTLS *solution by performing the algorithm* RTTLS *with the Gaussian random matrix $\Omega \in \mathbb{R}^{n\times l}$. If $x_{\mathrm{TTLS}} \ne \mathbf{0}$, $\|b\|_2 > \sigma_{q+1}$, and $\sigma_{k+1} \le (\overline{\sigma}_q - \sigma_{q+1})/(6 + 54\sqrt{l(n+1)})$, then we have*

$$\frac{\|x_{\mathrm{TTLS}} - x_{\mathrm{RTTLS}}\|_2}{\|x_{\mathrm{TTLS}}\|_2} \le \frac{(6\sigma_1 + 2\|b\|_2 - 2\sigma_{q+1})}{(\overline{\sigma}_q - \sigma_{q+1})(\|b\|_2 - \sigma_{q+1})}\left[\left(1 + 9\sqrt{l(n+1)}\right)\sigma_{k+1} + \sigma_{q+1}\right] \tag{15}$$

with failure probability of, at most, $3p^{-p}(p = l - k)$. *More specifically, if* $q = k$ *is the numerical rank of* $[A,\ b]$, *we get the bound below with probability not less than* $1 - 3p^{-p}$, *that is,*

$$\frac{\|x_{\mathrm{TTLS}} - x_{\mathrm{RTTLS}}\|_2}{\|x_{\mathrm{TTLS}}\|_2} \leq \frac{(6\sigma_1 + 2\|b\|_2)}{\overline{\sigma}_k \|b\|_2} \left(2 + 9\sqrt{l(n+1)}\right) \sigma_{k+1} + \mathcal{O}\left(\sigma_{k+1}^2\right). \tag{16}$$

Proof. Denote that $C = [A,\ b]$ and $\widehat{C} = QQ^{\mathrm{T}}C$. From Lemma 1 and the assumption, we know that

$$\|C - \widehat{C}\|_2 = \|C - QQ^{\mathrm{T}}C\|_2 \leq \left(1 + 9\sqrt{l(n+1)}\right) \sigma_{k+1} \leq \frac{1}{6}\left(\overline{\sigma}_q - \sigma_{q+1}\right)$$

with probability not less than $1 - 3p^{-p}$. Then, applying Lemma 2 and using the upper bound involving σ_{k+1}, we obtain (15).

For the specific cases, if the numerical rank of $[A,\ b]$ is $k = q$, it means that σ_{k+1} is very close to zero. The bound in (15) can be simplified to

$$\frac{\|x_{\mathrm{TTLS}} - x_{\mathrm{RTTLS}}\|_2}{\|x_{\mathrm{TTLS}}\|_2} \leq \frac{(6\sigma_1 + 2\|b\|_2 - 2\sigma_{k+1})}{(\overline{\sigma}_k - \sigma_{k+1})(\|b\|_2 - \sigma_{k+1})} \left(2 + 9\sqrt{l(n+1)}\right) \sigma_{k+1}.$$

Consider the Taylor expansion for the function $f(x) = 1/[(\overline{\sigma}_k - x)(\|b\|_2 - x)]$ at $x = 0$, we obtain that

$$f(\sigma_{k+1}) = \frac{1}{\overline{\sigma}_k \|b\|_2} + \left(\frac{1}{\overline{\sigma}_k^2 \|b\|_2} + \frac{1}{\overline{\sigma}_k \|b\|_2^2}\right) \sigma_{k+1} + \mathcal{O}\left(\sigma_{k+1}^2\right).$$

Substituting this equation into the above inequality directly, we can get (16). □

We point out that the assumption for σ_{k+1} usually holds for the ill-conditioned cases, where $k = q$ is the numerical rank, and we treat the other smaller singular values as zeros. For these cases, the upper bound (16) is of order $\mathcal{O}(\sigma_{k+1})$, and hence, the relative error of the solution for RTTLS and the solution for TTLS is small.

5 | NUMERICAL EXAMPLES

In this section, we give numerical examples to check our randomized TLS algorithms (NTLS and RTTLS). The following numerical tests are performed via MATLAB R2015a in a laptop with Intel Core i7 by using double precision. The tic-toc MATLAB function is used to measure the wall-clock time, which contains certain useful practical information and should be worth using for reference.

Example 1. In this example (see example 1 in the work of Baboulin and Gratton[44]), we consider the TLS problem $Ax \approx b$, where $[A,\ b]$ is defined by

$$[A,\ b] = Y \begin{bmatrix} \Lambda \\ \mathbf{0} \end{bmatrix} Z^{\mathrm{T}} \in \mathbb{R}^{m\times(n+1)}, Y = I_m - 2yy^{\mathrm{T}}, Z = I_{n+1} - 2zz^{\mathrm{T}},$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^{n+1}$ are random unit vectors, and $\Lambda = \mathrm{diag}(n, n-1, \ldots, 1, 1-\epsilon_p)$ for a given parameter ϵ_p. The quantity $\overline{\sigma}_n - \sigma_{n+1}$ measures the distance of our problem to nongenericity, and due to the interlacing property, we have in exact arithmetic

$$\overline{\sigma}_n - \sigma_{n+1} \leq \sigma_n - \sigma_{n+1} = \epsilon_p.$$

Example Prony. The TLS approach is a promising method in the field of signal processing. Rahman and Yu[45] presented a method for frequency estimation using TLS for solving the linear prediction equation. The problem here is taken from the work of Majda et al.[46] We first consider a set of linear prediction equations. Assume $a_j = [y_{j-1}, \ldots, y_{j+m-2}]^{\mathrm{T}}$, where $y_l = \sum_{j=1}^{p} c_j z_j^l,\ z_j = \exp(\lambda_j T),\ j = 1, \ldots, p$. The λ_j's and c_j's are to be determined. Furthermore, assume that c_j and z_j are nonzeros and z_j's are distinct for $j = 1, \ldots, p$. Let $A_n = [a_1, \ldots, a_n]$, $b_n = -a_{n+1}$ and consider the linear system

$$A_n x = b_n. \tag{17}$$

Assume $m \geq n, m \geq p$. It is known[47] that $\mathrm{rank}(A_n) = \min\{n, p\}$. If $n \geq p$, then (17) is compatible. For any solution $x = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})^{\mathrm{T}}$, construct a polynomial

$$P_n(z) = z^n + \alpha_{n-1} z^{n-1} + \cdots + \alpha_1 z + \alpha_0,$$

then we know that P_n has zeros $z_1, \ldots, z_p$.

TABLE 2 Tests of LTLS and NTLS ($l = 10$) on Example I. $\text{Time}_{\text{LTLS}}$ and $\text{Time}_{\text{NTLS}}$ are the computational times (in seconds) for the algorithms LTLS and NTLS, respectively. The relative errors are $\text{Err}_{\text{LTLS}} = \|x_{\text{TLS}} - x_{\text{LTLS}}\|_\infty / \|x_{\text{TLS}}\|_\infty$ and $\text{Err}_{\text{NTLS}} = \|x_{\text{TLS}} - x_{\text{NTLS}}\|_\infty / \|x_{\text{TLS}}\|_\infty$

Matrix size	Cond(A)	Cond($[A, b]$)	$\text{Time}_{\text{LTLS}}$	$\text{Time}_{\text{NTLS}}$	Err_{LTLS}	Err_{NTLS}
$m = 500$	2.00E+2	8.34E+6	0.0114	0.0039	1.17E−12	4.56E−13
$m = 1,000$	4.00E+2	1.67E+7	0.0216	0.0094	2.65E−13	1.56E−12
$m = 5,000$	2.00E+3	8.34E+7	0.9232	0.3738	5.83E−11	3.19E−10

Examples from Regularization Tools. Our ill-conditioned cases are taken from Hansen's Regularization Tools.[43] For example, the case SHAW is generated by the command $[\overline{A},\ \overline{b}] = \text{SHAW}(m)$, where m is the matrix size. Then, noise is added to $\overline{A}$ and $\overline{b}$. Suppose that δ is the relative noise level. We define

$$b = \overline{b} + \delta\|\overline{b}\|_2 \frac{\zeta}{\|\zeta\|_2}, \qquad A = \overline{A} + \delta\|\overline{A}\|_F \frac{Z}{\|Z\|_F},$$

where ζ is a random vector, $\zeta = 2 * \text{rand}(m, 1) - 1$, and Z is a random matrix, $Z = 2 * \text{rand}(m) - 1$ in MATLAB notations. It is easy to verify that $\|b - \overline{b}\|_2 / \|\overline{b}\|_2 = \|A - \overline{A}\|_F / \|\overline{A}\|_F = \delta$. Then, we will seek the TLS solution of $Ax \approx b$ and compare it with the corresponding true solution x_{true} in the sense of time and accuracy.

We apply the algorithms NTLS and RTTLS to Example I, Example Prony, and the ill-conditioned cases in Hansen's Regularization Tools.[43] We will compare the computational time and solution accuracy of our new randomized TLS algorithms with the traditional algorithms. For Example I, we compare our randomized algorithm NTLS with LTLS in which the symmetric Lanczos method is applied to the inverse of $C^{\text{T}}C$. For Example Prony and the ill-conditioned cases from Regularization Tools, we compare our randomized algorithm RTTLS with the traditional TTLS based on SVD and the LTTLS based on Golub–Kahan bidiagonalization.

5.1 | Algorithm NTLS

For the case Example I in Table 2, we choose $\epsilon_p = 9.99976031\text{E}{-1}$, and set $n = \frac{2}{5}m$. The solution x_{TLS} is computed by (4), whereas x_{LTLS} and x_{NTLS} are obtained by the algorithms LTLS and NTLS, respectively. Denote the relative errors $\text{Err}_{\text{LTLS}} = \|x_{\text{TLS}} - x_{\text{LTLS}}\|_\infty / \|x_{\text{TLS}}\|_\infty$ and $\text{Err}_{\text{NTLS}} = \|x_{\text{TLS}} - x_{\text{NTLS}}\|_\infty / \|x_{\text{TLS}}\|_\infty$ as well as the corresponding execution times $\text{Time}_{\text{LTLS}}$ and $\text{Time}_{\text{NTLS}}$. From Table 2, we can see that our algorithm NTLS on large matrices outperforms the algorithm LTLS according to computational time, whereas the accuracy of solutions of the three methods is comparable. For the small matrices, the advantage of NTLS will not be so obvious. Our randomized algorithm usually is effective for matrices that have excellent low-rank approximations, for example, the case where there exists an obvious gap in its singular values. In fact, Example I is somewhat a well-conditioned case, not a rank-deficient one. Even for this case, the randomized algorithm NTLS can still give an approximate solution with good accuracy, which again sheds some light on the robustness of the randomized algorithms.

5.2 | Algorithm RTTLS

For the example based on TLS-Prony modeling, we choose λ_j and c_j as in Table 3. In this example, the parameters $T = 0.2$, $m = 2,000, p = 12$, and $n = 1,000$ are used, and we compare the TTLS with the RTTLS, where the sampling size is chosen as $l = p + 1$. Here, p is the parameter k in the algorithm RTTLS. The plots for the solution to (17) are shown in Figure 1, and the results for time cost and relative errors are given in Table 4. Since we do not know the exact solution of this problem, we define the relative errors by $\text{Err}_{\text{LTTLS}} = \|x_{\text{TTLS}} - x_{\text{LTTLS}}\|_\infty / \|x_{\text{TTLS}}\|_\infty$ and $\text{Err}_{\text{RTTLS}} = \|x_{\text{TTLS}} - x_{\text{RTTLS}}\|_\infty / \|x_{\text{TTLS}}\|_\infty$, respectively. We find that RTTLS needs much less time while keeping almost the same accuracy.

For the ill-conditioned cases from the Regularization Tools, we first set the parameter for sampling size $l = 20$ for all cases. The results are given in Table 5. It is well known that l should be sufficiently large for randomized methods. Even though carefully choosing the parameter l can further improve the performance of the randomized algorithm, we set l to be the same value for the sake of simplicity. The truncation parameter k is estimated by the randomized algorithm with GCV and TSVD.[24] Since the true solutions of these problems are known in advance, we compute the relative errors by $\|x_{\text{comp}} - x_{\text{true}}\|_\infty / \|x_{\text{true}}\|_\infty$, where x_{comp}'s are the computed solutions through different algorithms. For the traditional method TTLS based on the complete SVD of $[A,\ b]$, it is very costly for large-scale problems and, therefore, meaningless to record its time cost. We just list its relative errors in the fourth column and do not display the computational time. For

TABLE 3 Six pairs of poles and residues

λ_j	c_j
$-0.082 \pm 0.926i$	1
$-0.147 \pm 2.874i$	1
$-0.188 \pm 4.835i$	1
$-0.220 \pm 6.800i$	1
$-0.247 \pm 8.767i$	1
$-0.270 \pm 10.733i$	1

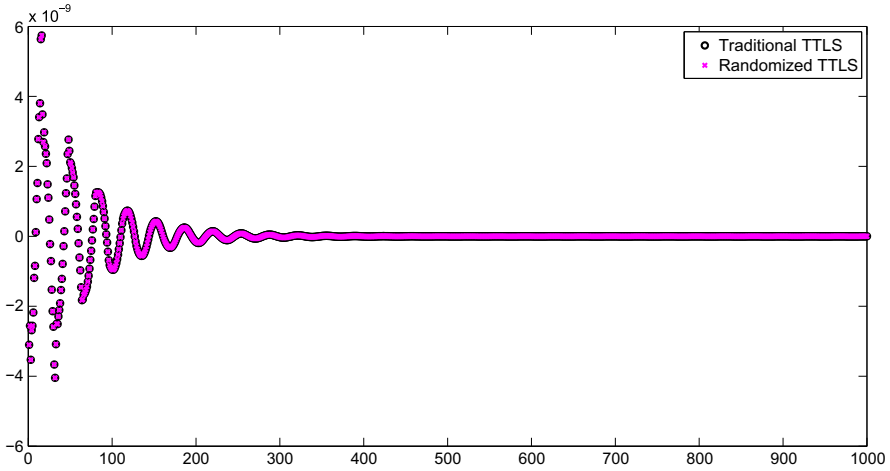

FIGURE 1 Computed TTLS and RTTLS solutions of Prony (17)

TABLE 4 Test of Prony on RTTLS ($l = p + 1$). $\text{Time}_{\text{LTTLS}}$ and $\text{Time}_{\text{RTTLS}}$ are the computational times (in seconds) for the algorithms LTTLS and RTTLS, respectively. The relative errors are $\text{Err}_{\text{LTTLS}} = \|x_{\text{TTLS}} - x_{\text{LTTLS}}\|_\infty / \|x_{\text{TTLS}}\|_\infty$ and $\text{Err}_{\text{RTTLS}} = \|x_{\text{TTLS}} - x_{\text{RTTLS}}\|_\infty / \|x_{\text{TTLS}}\|_\infty$

Matrix size	$\text{Time}_{\text{LTTLS}}$	$\text{Time}_{\text{RTTLS}}$	$\text{Err}_{\text{LTTLS}}$	$\text{Err}_{\text{RTTLS}}$
$m = 2,000$	0.2777	0.0096	4.10E−8	4.10E−8

TABLE 5 Algorithm RTTLS on ill-conditioned cases. RTTLS needs less computational time than LTTLS based on the Lanczos procedure

			TTLS	LTTLS		RTTLS	
Matrix	Size m	k	Error	Time	Error	Time	Error
BAART	1,000	9	7.76E−2	0.1832	7.76E−2	0.0070	7.76E−2
	5,000	7	1.06E−1	1.5913	1.06E−1	0.1584	1.06E−1
DERIV2	1,000	17	9.83E−1	0.1927	9.83E−1	0.0080	9.83E−1
	5,000	18	9.96E−1	1.1918	9.96E−1	0.0730	9.96E−1
FOXGOOD	1,000	6	3.06E−3	0.1191	3.06E−3	0.0054	3.06E−3
	5,000	7	2.00E−3	0.6481	2.00E−3	0.0627	2.02E−3
GRAVITY	1,000	16	5.20E−3	0.1794	5.21E−3	0.0073	5.26E−3
	5,000	16	5.44E−3	1.1615	5.44E−3	0.1009	5.44E−3
HEAT	1,000	17	8.12E−2	0.1814	9.12E−2	0.0073	8.47E−2
	5,000	17	8.19E−2	1.1312	8.19E−2	0.0970	8.21E−2
I_LAPLACE	1,000	19	5.98E−1	0.2533	5.98E−1	0.0096	5.93E−1
	5,000	19	8.37E−1	1.3123	8.37E−1	0.1164	8.35E−1
PHILLIPS	1,000	11	8.13E−3	0.1574	8.13E−3	0.0071	8.10E−3
	5,000	12	5.08E−3	0.8494	5.09E−3	0.0743	7.00E−3
SHAW	1,000	19	2.06E−3	0.2487	7.96E−3	0.0075	8.01E−4
	5,000	19	4.63E−4	2.1058	3.85E−2	0.0873	1.42E−3

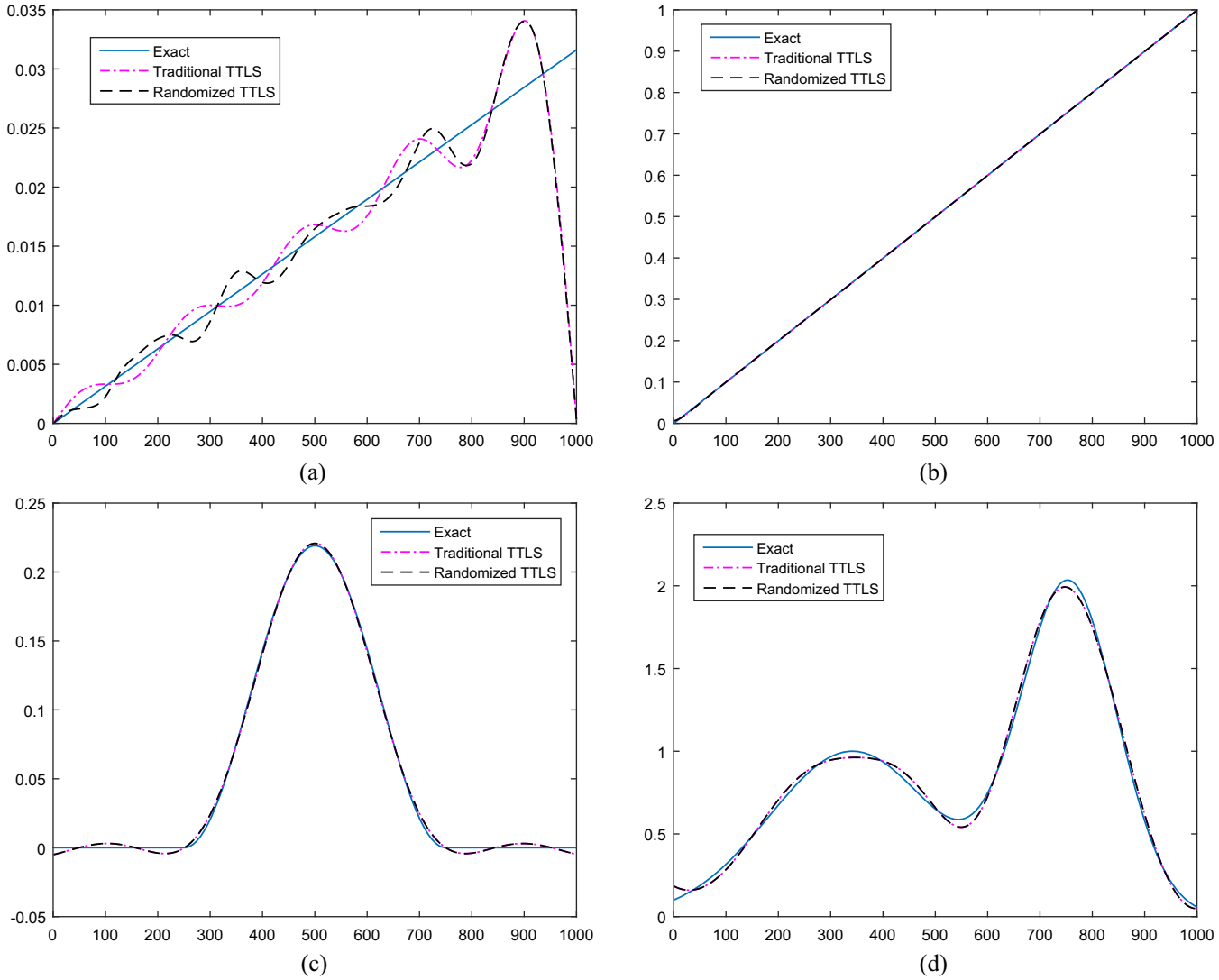

FIGURE 2 RTTLS for ill-conditioned cases of size $m = 1,000$. (a) DERIV2. (b) FOXGOOD. (c) PHILLIPS. (d) SHAW

LTTLS and RTTLS, we incorporate both the time cost and relative errors into Table 5 for comparison. We find that RTTLS can give accuracy of almost the same magnitude but cost much less time compared with LTTLS and TTLS.

We call the subroutine lansvd in PROPACK[48] to implement LTTLS. The algorithm LTTLS can show overwhelming advantages over the TTLS when the size of the matrix becomes larger. However, it is still not as efficient as the randomized one because of its BLAS-2 operations. The randomized strategy can greatly speed up the classical algorithm TTLS. The advantage of our algorithm RTTLS is more obvious when we test the larger matrices. To intuitively illustrate the performance of these algorithms, we plot the computed solutions of four chosen cases in Figure 2.

6 | CONCLUSION

In this paper, for the numerical solutions of medium-sized or large-scale TLS problems, we have proposed randomized algorithms NTLS and RTTLS, respectively. Specifically, the algorithm NTLS is compared with the traditional method LTLS based on the symmetric Lanczos method, whereas the performance of the algorithm RTTLS is illustrated through the comparison with the method LTTLS based on Golub–Kahan bidiagonalization. These randomized algorithms can greatly reduce the computational time and still yield solutions with good accuracy. The regularization parameter in RTTLS can be estimated by the truncation parameter of the TSVD solution of $Ax \approx b$ based on a fast randomized SVD of A,[24] and then, a randomized SVD of $[A, b]$ together with this truncation parameter can be used to approximate the TTLS solution for

large-scale ill-conditioned cases. Further research on other regularization parameter choices and other techniques such as Tikhonov regularization is needed. Another question is how to design adaptive randomized algorithms for the cases where the numerical rank is not known or cannot be estimated in advance. These questions will be the topics for future research.

ACKNOWLEDGEMENTS

The authors sincerely thank the editor and two anonymous referees for their comments and suggestions, which led to great improvements of this paper. The work of P. Xie was supported by the National Natural Science Foundation of China under Grants 11271084, 11801534, and 11701538 and the Fundamental Research Funds for the Central Universities under Grant 201562012. Part of the work was done when the author visited Wuhan University in 2015 and 2016. The work of H. Xiang was supported by the National Natural Science Foundation of China under Grants 11571265, 11471253, and NSFC-RGC(China-Hong Kong) 11661161017. Part of the work was completed when the author visited Fudan University in 2015 and 2016 and The Chinese University of Hong Kong in 2014. The work of Y. Wei was supported by the National Natural Science Foundation of China under Grant 11771099 and the International Cooperation Project of the Shanghai Municipal Science and Technology Commission under Grant 16510711200.

ORCID

Hua Xiang http://orcid.org/0000-0002-5094-2376
Yimin Wei http://orcid.org/0000-0001-6192-0546

REFERENCES

1. Van Huffel S, Vandewalle J. The total least squares problem: computational aspects and analysis. Philadelphia, PA: SIAM; 1991.
2. Golub GH, Van Loan CF. An analysis of the total least squares problem. SIAM J Numer Anal. 1980;17(6):883–893.
3. Wei M. Algebraic relations between the total least squares and least squares problems with more than one solution. Numer Math. 1992;62(1):123–148.
4. Mastronardi N, Lemmerling P, Kalsi A, O'Leary DP, Van Huffel S. Implementation of the regularized structured total least squares algorithms for blind image deblurring. Linear Algebra Appl. 2004;391:203–221.
5. Van Huffel S, editor. Recent advances in total least squares techniques and errors-in-variables modeling. Philadelphia, PA: SIAM; 1997.
6. Hansen PC, Nagy JG, O'leary DP. Deblurring images: matrices, spectra and filtering. Vol. 3. Philadelphia, PA: SIAM; 2006.
7. Lee G, Fu H, Barlow JL. Fast high-resolution image reconstruction using Tikhonov regularization based total least squares. SIAM J Sci Comput. 2013;35(1):B275–B290.
8. Hnětynková I, Plešinger M, Sima DM, Strakoš Z, Van Huffel S. The total least squares problem in $AX \approx B$: a new classification with the relationship to the classical works. SIAM J Matrix Anal Appl. 2011;32(3):748–770.
9. Hnětynková I, Plešinger M, Strakoš Z. The core problem within a linear approximation problem $AX \approx B$ with multiple right-hand sides. SIAM J Matrix Anal Appl. 2013;34:917–931.
10. Hnětynková I, Plešsinger M, Strakoš Z. Band generalization of the Golub-Kahan bidiagonalization, generalized Jacobi matrices, and the core problem. SIAM J Matrix Anal Appl. 2015;36:417–434.
11. Hnětynková I, Plešinger M, Sima DM. Solvability of the core problem with multiple right-hand sides in TLS sense. SIAM J Matrix Anal Appl. 2016;37:861–876.
12. Van Huffel S. Partial singular value decomposition algorithm. J Comput Appl Math. 1990;33(1):105–112.
13. Fierro RD, Golub GH, Hansen PC, O'leary DP. Regularization by truncated total least squares. SIAM J Sci Comput. 1997;18(4):1223–1241.
14. O'Leary DP, Simmons JA. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. SIAM J Sci Comput. 1981;2(4):474–489.
15. Björck Å. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. BIT Numer Math. 1988;28(3):659–670.
16. Golub GH, Kahan W. Calculating the singular values and pseudo-inverse of a matrix. J Soc Indust Appl Math Ser B Numer Anal. 1965;2:205–224.
17. Liesen J, Strakoš Z. Krylov subspace methods Principles and analysis. Oxford, UK: Oxford University Press; 2013. Numerical mathematics and scientific computation.
18. Halko N, Martinsson PG, Tropp JA. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. 2011;53(2):217–288.
19. Drineas P, Mahoney MW, Muthukrishnan S, Sarlós T. Faster least squares approximation. Numer Math. 2011;117(2):219–249.
20. Rokhlin V, Tygert M. A fast randomized algorithm for overdetermined linear least-squares regression. Proc Natl Acad Sci. 2008;105(36):13212–13217.

21. Avron H, Maymounkov P, Toledo S. Blendenpik: supercharging LAPACK's least-squares solver. SIAM J Sci Comput. 2010;32(3):1217–1236.
22. Coakley ES, Rokhlin V, Tygert M. A fast randomized algorithm for orthogonal projection. SIAM J Sci Comput. 2011;33(2):849–868.
23. Meng X, Saunders MA, Mahoney MW. LSRN: a parallel iterative solver for strongly over- or underdetermined systems. SIAM J Sci Comput. 2014;36(2):C95–C118.
24. Xiang H, Zou J. Regularization with randomized SVD for large-scale discrete inverse problems. Inverse Probl. 2013;29(8):085008.
25. Xiang H, Zou J. Randomized algorithms for large-scale inverse problems with general Tikhonov regularizations. Inverse Probl. 2015;31:085008.
26. Wei Y, Xie P, Zhang L. Tikhonov regularization and randomized GSVD. SIAM J Matrix Anal Appl. 2016;37(2):649–675.
27. Golub GH, Van Loan CF. Matrix computations. 4th ed. Baltimore, MD: Johns Hopkins University Press; 2013. Johns Hopkins studies in the mathematical sciences.
28. Paige CC, Strakoš Z. Core problems in linear algebraic systems. SIAM J Matrix Anal Appl. 2005;27(3):861–875.
29. Wei M. The analysis for the total least squares problem with more than one solution. SIAM J Matrix Anal Appl. 1992;13(3):746–763.
30. Drineas P, Mahoney MW. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. J Mach Learn Res. 2005;6:2153–2175.
31. Demmel JW. Applied numerical linear algebra. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.
32. Beck A, Ben-Tal A. On the solution of the Tikhonov regularization of the total least squares problem. SIAM J Optim. 2006;17(1):98–118.
33. Golub GH, Hansen PC, O'leary DP. Tikhonov regularization and total least squares. SIAM J Matrix Anal Appl. 1999;21(1):185–194.
34. Lampe J, Voss H. Large-scale Tikhonov regularization of total least squares. J Comput Appl Math. 2013;238:95–108.
35. Lu S, Pereverzev SV, Tautenhahn U. Regularized total least squares: computational aspects and error bounds. SIAM J Matrix Anal Appl. 2009;31(3):918–941.
36. Beck A, Ben-Tal A, Teboulle M. Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares. SIAM J Matrix Anal Appl. 2006;28(2):425–445.
37. Lampe J, Voss H. Global convergence of RTLSQEP: a solver of regularized total least squares problems via quadratic eigenproblems. Math Model Anal. 2008;13(1):55–66.
38. Renaut RA, Guo H. Efficient algorithms for solution of regularized total least squares. SIAM J Matrix Anal Appl. 2004;26(2):457–476.
39. Sima DM, Van Huffel S, Golub GH. Regularized total least squares based on quadratic eigenvalue problem solvers. BIT. 2004;44(4):793–812.
40. Hnětynková I, Plešinger M, Žáková J. Filter factors of truncated TLS regularization with multiple observations. Appl Math. 2017;62(2):105–120.
41. Gratton S, Titley-Peloquin D, Ilunga JT. Sensitivity and conditioning of the truncated total least squares solution. SIAM J Matrix Anal Appl. 2013;34(3):1257–1276.
42. Markovsky I, Van Huffel S. Overview of total least-squares methods. Signal Process. 2007;87(10):2283–2302.
43. Hansen PC. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. Numer Algorithms. 1994;6(1–2):1–35.
44. Baboulin M, Gratton S. A contribution to the conditioning of the total least-squares problem. SIAM J Matrix Anal Appl. 2011;32(3):685–699.
45. Rahman MD, Yu K-B. Total least squares approach for frequency estimation using linear prediction. IEEE Trans Acoust Speech Signal Process. 1987;35(10):1440–1454.
46. Majda G, Strauss WA, Wei M. Computation of exponentials in transient data. IEEE Trans Antennas Propag. 1989;37(10):1284–1290.
47. Wei M, Majda G. A new theoretical approach for Prony's method. Linear Algebra Appl. 1990;136:119–132.
48. Larsen RM. Lanczos bidiagonalization with partial reorthogonalization (DAIMI Report Series). Aarhus, Denmark: Department of Computer Science, Aarhus University. 1998. http://sun.stanford.edu/~rmunk/PROPACK/

How to cite this article: Xie P, Xiang H, Wei Y. Randomized algorithms for total least squares problems. *Numer Linear Algebra Appl*. 2019;26:e2219. https://doi.org/10.1002/nla.2219