Single-pass randomized QLP decomposition for low-rank approximation

Huan Ren1 · Guiyun Xiao1 · Zheng-Jian Bai1

Received: 7 July 2022 / Revised: 16 October 2022 / Accepted: 24 October 2022 / Published online: 16 November 2022
© The Author(s) under exclusive licence to Istituto di Informatica e Telematica (IIT) 2022

Abstract
As a special UTV decomposition, the QLP decomposition is an effective alternative of the singular value decomposition (SVD) for the low-rank approximation. In this paper, we propose a single-pass randomized QLP decomposition algorithm for computing a low-rank matrix approximation. Compared with the randomized QLP decomposition, the complexity of the proposed algorithm does not increase significantly and the data matrix needs to be accessed only once. Therefore, our algorithm is suitable for a large matrix stored outside of memory or generated by streaming data. In the error analysis, we give the matrix approximation error analysis. We also provide singular value approximation error bounds, which can track the target largest singular values of the data matrix with high probability. Numerical experiments are also reported to verify our results.

Keywords QLP decomposition · Randomized algorithm · Single-pass · Singular value · Low-rank approximation

Mathematics Subject Classification 65F55 · 65F99 · 68W20

1 Introduction
Let $A \in \mathbb{R}^{m \times n}$ be a data matrix. The low-rank approximation of $A$ is to compute two low-rank matrices $E \in \mathbb{R}^{m \times k}$ and $F \in \mathbb{R}^{k \times n}$ such that
where \( \text{rank}(E) = \text{rank}(F) = k \), and \( k \ll \min\{m, n\} \), the rank \( k \) is an a priori estimate.

In the era of big data, a large amount of data needs to be processed. Thus the scale of the data matrix can be very large. In this case, the low-rank approximation in the form of (1.1) can greatly reduce the storage of the data matrix \( A \) (i.e., we only need to store \( E \) and \( F \) instead of \( A \)). Low-rank approximation is one of the essential tools in scientific computing, including principal component analysis [1–4], data analysis [5–7], fast approximate algorithms for PDEs [8–10], discrete ill-posed problems [11, 12], and hierarchical matrix computations [13–17].

For a data matrix \( A \in \mathbb{R}^{m \times n} \), we usually consider a rank-\( k \) approximate SVD, i.e.,

\[
A \approx U_k \Sigma_k V_k^T, \tag{1.2}
\]

where \( U_k = [u_1, u_2, \ldots, u_k] \), \( \Sigma_k = \text{diag}(\sigma_1(A), \sigma_2(A), \ldots, \sigma_k(A)) \), and \( V_k = [v_1, v_2, \ldots, v_k] \) with \( \sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_k(A) \geq 0 \), \( u_j \) and \( v_j \) are the left and right singular vectors corresponding to the singular value \( \sigma_j(A) \), respectively.

It was showed that the rank-\( k \) truncated SVD provides the smallest error for the rank-\( k \) approximation of a matrix in the spectral norm or Frobenius norm [18, 19]. Therefore, the truncated SVD is the best low-rank approximation with a given fixed rank. However, the computation of a SVD of a large matrix is very costly. Therefore, we wish to find an algorithm for computing a low-rank approximation to a large matrix. As expected, we hope the proposed algorithm is close to the quality that the SVD provides but needs much lower cost.

The QLP decomposition was proposed by Stewart in 1999 [20], which may be regarded as an economical method for computing a low-rank approximation to the SVD. To find a low-rank approximation to a data matrix, it is natural to use the SVD to determine the gap in the singular values and the corresponding singular subspaces while the computation cost is very high. As a special UTV decomposition [18], the QLP decomposition can provide high-quality low-rank approximation and it is more cheaper than the SVD. Compared with other general matrix decompositions (e.g., the generalized Nyström method [21]), the QLP decomposition is a completely orthogonal decomposition, which can provide high-quality singular value, rank, range and null space information of the original matrix (see [20] and [18, §5.4.6]). In fact, the QLP decomposition is equivalent to two consecutive QR decompositions with column pivoting (QRCP). Specifically, the two-step QRCP is performed on the data matrix \( A \in \mathbb{R}^{m \times n} \) in the sense that

\[
AP_0 = Q_0 R_0 \quad \text{and} \quad R_0^T P_1 = Q_1 L_0^T, \tag{1.3}
\]

where \( P_0 \in \mathbb{R}^{n \times n} \) and \( P_1 \in \mathbb{R}^{m \times m} \) are two permutation matrices, \( Q_0 \in \mathbb{R}^{m \times m} \) and \( Q_1 \in \mathbb{R}^{m \times n} \) are two orthogonal matrices, \( R_0 \in \mathbb{R}^{m \times m} \) is an upper triangular matrix, and \( L \in \mathbb{R}^{m \times n} \) is a lower triangular matrix. From (1.3) we have

\[
A = Q_0 P_1 L Q_1^T P_0^T = Q L P_1^T, \quad Q = Q_0 P_1, \quad P = P_0 Q_1.
\]
Huckaby and Chan [22] showed that the singular values of the diagonal blocks of the $L$-factor approximate the singular values of the original matrix $A$. The truncated QLP decomposition of $A$ can be expressed as follows:

$$A \approx Q_k L_k P_k^T \equiv A_k,$$

(1.4)

where both $Q_k = Q(\cdot, 1 : k) \in \mathbb{R}^{m \times k}$ and $P_k = P(\cdot, 1 : k) \in \mathbb{R}^{n \times k}$ have orthonormal column vectors and $L_k = L(1 : k, 1 : k) \in \mathbb{R}^{k \times k}$ is lower triangular. Here, $A(\cdot, 1 : k)$ denotes a matrix composed of Columns 1 to $k$ of a matrix $A$. The truncated QLP decomposition (1.4) can also be regarded as a low-rank approximation of $A$.

As early as 2000, randomized algorithms received much attention. Specifically, Papadimitriou et al. [23] proposed a Monte Carlo algorithm for low-rank approximation. Liberty et al. [24] gave a randomized algorithm for low-rank approximation. Halko et al. [25] presented probabilistic algorithms for low-rank approximation. In recent years, randomized algorithms for low-rank approximation have attracted considerable attention [26–30], including randomized QRCP algorithms [31–34]. Compared with deterministic algorithms, randomized algorithms for low-rank approximation have the advantages of low complexity, fast running speed and easy implementation. However, as noted in [35, 36], it is vital to reduce the communication costs including bandwidth and latency costs since the gap between computation and the communication costs grows exponentially. We point out that these randomized algorithms need to access the original matrix $A$ at least twice leading to at least two data communications, which is expensive for the large data matrix stored outside of core memory or generated by stream data.

In order to reduce the cost of data communication, some single-pass algorithms have been proposed [25, 37–42]. As noted in [43], in computing, a single-pass algorithm or one-pass algorithm is a streaming algorithm which reads its input exactly once. In this paper, based on the idea of single-pass and inspired by the work of Wu and Xiang [44], we propose a single-pass randomized QLP decomposition algorithm for computing a low-rank approximation of a data matrix. We also give the bounds of matrix approximation error and singular value approximation error for the proposed randomized algorithm, which hold with high probability. Finally, some numerical experiments are reported to illustrate the efficiency of the proposed algorithm.

The main contributions of this paper are summarized as follows. (i) By randomized row and column sampling simultaneously, our method can be used for a data matrix stored outside of memory or generated by stream data, where the data matrix can be accessed only once. (ii) We give the matrix approximation error analysis and the singular value approximation error analysis. Different from the error bounds between the singular values of the $L$-factor as a whole and the singular values of the data matrix in the expectation sense in the algorithm in [44], we provide the relative error bounds between the singular values of diagonal blocks of the $L$-factor and the singular values of the data matrix, which hold with high probability. Our theoretical results confirm that our algorithm can effectively track the target largest singular values of the data matrix.
Throughout the paper, we use the following notations. Let $\mathbb{R}^{m \times n}$ be the set of all $m \times n$ real matrices. For any $A \in \mathbb{R}^{m \times n}$, let $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_q(A) \geq 0$ denote the singular values of $A$, where $q = \min\{m, n\}$. The superscripts ‘$\cdot^T$’ and ‘$\cdot^\dagger$’ stand for the transpose and the Moore–Penrose inverse of a matrix, respectively. In addition, $\mathbb{E}(\cdot)$ denotes the expectation of a random variable and $\mathbb{P}(\cdot)$ denotes the probability of a random event.

The rest of this paper is organized as follows. In Sect. 2 we give some preliminary results related to subgaussian random matrices and some basic QLP decomposition algorithms. In Sect. 3 we propose a single-pass randomized QLP decomposition algorithm and present the corresponding complexity analysis. In Sect. 4 we derive the bounds of matrix approximation error and singular value approximation error for the proposed randomized algorithm. Finally, some numerical examples and concluding remarks are given in Sects. 5 and 6, respectively.

2  Preliminaries

In this section, we review some preliminary results on Gaussian random matrices and randomized QLP decomposition.

2.1 Gaussian random matrix

In this subsection, we recall some preliminary definitions and lemmas on Gaussian random matrices.

**Definition 2.1** [45, p.272] A random matrix $\Omega$ is called Gaussian if all its entries are independent standard Gaussian/normal random variables.

The following lemma gives a large deviation bound of the spectral norm of a Moore–Penrose inverse of a Gaussian random matrix.

**Lemma 2.2** [25, Proposition 10.4] Let $\Omega$ be an $m \times n$ Gaussian matrix, where $n - m \geq 4$. For all $\alpha \geq 1$, we have

$$\mathbb{P}\left(\|\Omega^\dagger\|_2^2 \geq \frac{e \sqrt{n}}{n - m + 1} \cdot \alpha\right) \leq \alpha^{-(n-m+1)}.$$  

The following lemma gives a large deviation inequality for the spectral norm of a Gaussian random matrix.

**Lemma 2.3** [46, Corollary 5.35] Let $\Omega$ be an $m \times n$ Gaussian matrix. Then for any $\beta \geq 0$,

$$\|\Omega\|_2 \leq \sqrt{m} + \sqrt{n} + \beta$$

with probability not less than $1 - 2 \exp(-\beta^2/2)$.
2.2 Randomized QLP decomposition

In this subsection, we recall the QLP decomposition and the randomized QLP decomposition. We first recall the QLP decomposition [20], which is described as Algorithm 1.

Algorithm 1 The QLP decomposition [20]
Input: \( A \in \mathbb{R}^{m \times n} \).
Output: Matrices \( Q \), \( L \), and \( P \) such that \( A = QLP^T \), where \( Q \) and \( P \) are orthogonal and \( L \) is lower triangular.

code
function \([Q, L, P] = QLP(A)\)
1: \([Q_0, R_0, P_0]\) = qr(\( A \)). \( \triangleright \) QRCP of \( A \).
2: \([Q_1, L^T, P_1]\) = qr(\( R_0^T \)). \( \triangleright \) QRCP of \( R_0^T \).
3: \( Q = Q_0P_1 \), \( P = P_0Q_1 \), \( L = L(l: k, 1:k) \).

In [44], a randomized QLP decomposition (RQLP) was proposed based on column sampling, as shown in Algorithm 2. For a data matrix \( A \in \mathbb{R}^{m \times n} \) with the target rank \( k \) and the oversampling parameter \( p \geq 2 \), a sample matrix \( Y = A\Omega \) is formed, where \( \Omega \in \mathbb{R}^{n \times l} \) is a Gaussian random matrix \( (l = k + p) \). Then the approximate basis matrix \( V \in \mathbb{R}^{m \times l} \) of the range of \( A \) is formed by using the QR decomposition of \( Y \). However, the QLP decomposition of the \( l \times n \) matrix \( B = V^TA \) formed in the RQLP algorithm needs \( \mathcal{O}(n^2) + \mathcal{O}(n^2) \) operations. To further improve the efficiency of Algorithm 2, we will adopt row sampling.

Algorithm 2 Randomized QLP decomposition (RQLP) with column sampling [44]
Input: \( A \in \mathbb{R}^{m \times n} \), target rank: \( k \), oversampling parameter: \( p \geq 2 \), and number of columns sampled: \( l = k + p \).
Output: Matrices \( Q \), \( L \), \( P \) such that \( A \approx QLP^T \), where \( Q \) and \( P \) have orthonormal columns, \( L \) is a lower triangular matrix.

code
1: \( \Omega = \text{randn}(n, l) \).
2: \( Y = A\Omega \).
3: \([V, R] = \text{qr}(Y, 0)\). \( \triangleright \) QRCP of \( B \).
4: \( B = V^TA \).
5: \([Q_0, R_0, P_0]\) = qr(\( B \)). \( \triangleright \) QRCP of \( R_0^T \).
6: \([Q_1, L^T, P_1]\) = qr(\( R_0^T \)).
7: \( Q = Q_0P_1(:, 1:k) \), \( P = VP_0Q_1(:, 1:k) \), \( L = L(l:k, 1:k) \).

As in [25, 37], to calculate a low-rank approximation of \( A \), we first construct a low-rank matrix \( V \) with orthonormal columns such that \( A \approx VV^TA \) with \( \text{range}(V) \approx \text{range}(A) \) or \( A \approx AVV^T \) with \( \text{range}(V) \approx \text{range}(A^T) \). Instead of the column sampling in Algorithm 2, we perform row sampling \( Y = \Omega A \) on the data matrix \( A \) by a Gaussian random matrix \( \Omega \in \mathbb{R}^{n \times m} \), and calculate the QR decomposition of \( Y^T \) to get the \( Q \)-factor \( V \in \mathbb{R}^{m \times d} \). Then we have

\[
A \approx AVV^T = BV^T, \quad B := AV.
\]
Then, the QLP decomposition of the \( m \times l \) matrix \( B = AV \) needs \( \mathcal{O}(ml^2 + l^3) \) operations. This shows that the RQLP algorithm with row sampling may be more effective when \( m \approx n \).

Therefore, we can revise Algorithm 2 by using row sampling. The MATLAB pseudo-code of the RQLP algorithm with row sampling is stated as Algorithm 3.

**Algorithm 3 Randomized QLP decomposition (RQLP) with row sampling**

**Input:** \( A \in \mathbb{R}^{m \times n} \), target rank: \( k \), oversampling parameter: \( p \geq 2 \), and number of columns sampled: \( l = k + p \).

**Output:** Matrices \( Q, L, P \) such that \( A \approx QLP^T \), where \( Q \) and \( P \) have orthonormal columns, \( L \) is a lower triangular matrix.

1: \( \Omega = \text{randn}(l, m) \).
2: \( Y = \Omega A \).
3: \( [V, R] = \text{qr}(Y^T, 0) \).
4: \( B = AV \).
5: \( [Q_0, R_0, P_0] = \text{qr}(B, 0) \). \( \triangleright \) QRCP of \( B \).
6: \( [Q_1, L^T, P_1] = \text{qr}(R_0^T) \). \( \triangleright \) QRCP of \( R_0^T \).
7: \( Q = Q_0P_1(:, 1 : k) \), \( P = VP_0Q_1(:, 1 : k) \), \( L = L(l : k, 1 : k) \).

### 3 Single-pass randomized QLP decomposition

We observe from Algorithm 3 that the randomized QLP decomposition algorithm in [44] needs to access the data matrix \( A \) twice. This is costly since the data matrix is often stored outside of core memory. In particular, the input data matrix can only be accessed once when it is generated by stream data.

Inspired by the single-pass idea in [25, 37, 39–42] and the randomized QLP decomposition algorithm in [44], in this section, we propose a single-pass randomized QLP decomposition algorithm for computing a low-rank approximation to a data matrix \( A \in \mathbb{R}^{m \times n} \). This is motivated by single-pass algorithms [25, 37] and the randomized QLP decomposition [44].

The basic idea of a single-pass randomized algorithm is that each entry of the input matrix can only be accessed once. To do so, we wish replace the matrix \( B = AV \) by another expression without \( A \). We note that, for the matrix \( V \), we have \( A \approx AVV^T = BV^T \) since \( B = AV \). Postmultiplying a Gaussian random matrix \( \Omega_2 \in \mathbb{R}^{m \times l_2} \) on both sides of \( A \approx BV^T \), we get

\[
Y_2 = A\Omega_2 \approx BV^T\Omega_2.
\]

Then, the matrix \( B \) can be approximately expressed as \( B \approx Y_2(V^T\Omega_2)^T \).

Based on the above analysis, we give a single-pass randomized QLP decomposition algorithm for computing a low-rank approximation of the data matrix \( A \), which is stated as Algorithm 4.
Algorithm 4 Single-pass randomized QLP decomposition (SPRQLP)

Input: $A \in \mathbb{R}^{m \times n}$, target rank: $k$, oversampling parameter: $p \geq 2$, number of rows sampled: $l_1 = k + p$, and number of columns sampled: $l_2 \geq l_1$.  

Output: Matrices $Q, L, P$ such that $A \approx QLP^T$, where $Q$ and $P$ have orthonormal columns, $L$ is lower triangular matrix.

1. $\Omega_1 = \text{randn}(l_1, m)$, $\Omega_2 = \text{randn}(n, l_2)$.
2. $Y_1 = \Omega_1 A$, $Y_2 = A \Omega_2$.
3. $[V, R] = \text{qr}(Y_1^T, 0)$.
4. $B = Y_2 (V^T \Omega_2)^\dagger$.
5. $[Q_0, R_0, P_0] = \text{qr}(B, 0)$, where $Q_0 \in \mathbb{R}^{m \times l_1}$ and $R_0 \in \mathbb{R}^{l_1 \times l_1}$.
6. $[Q_1, R_1, P_1] = \text{qr}(R_0^\dagger)$.
7. $Q = Q_0 P_1(:, 1 : k)$, $L = R_1^T (1 : k, 1 : k)$, $P = V P_0 Q_1(:, 1 : k)$.

On Algorithm 4, we have some comments as follows.

Remark 3.1 In Step 4 of Algorithm 4, we can compute $(V^T \Omega_2)^\dagger = (V^T \Omega_2)^T (V^T \Omega_2 (V^T \Omega_2)^T)^{-1}$ since $V^T \Omega_2$ is a Gaussian random matrix (the Gaussian distribution is rotationally invariant [25]) and thus $V^T \Omega_2$ is full row rank. In addition, Algorithm 4 reduces to Algorithm 3 if $\Omega_2 = I$.

We note that the data matrix $A$ appears twice in Step 2 of Algorithm 4, but this does not indicate that the algorithm is not single-pass. In fact, we form the sampling matrices $Y_1 = \Omega_1 A$ and $Y_2 = A \Omega_2$ (i.e., both row space and column space of $A$ are randomly sampled simultaneously) in a single pass over $A$ via reading $A$ into RAM once [25, 38, 47]. Hence, the proposed algorithm is single-pass.

On the complexity of Algorithm 4, we have the following remarks.

- Step 1: Generating random matrices $\Omega_1, \Omega_2$ takes $O(ml_1 + nl_2)$ operations;
- Step 2: Computing $Y_1 = \Omega_1 A$ and $Y_2 = A \Omega_2$ takes $O(mn(l_1 + l_2))$ operations;
- Step 3: Computing unpivoted QR decomposition of $Y_1^T$ of size $n \times l_1$ takes $O(nl_1^2)$ operations;
- Step 4: Computing $V^T \Omega_2$ takes $O(ml_1 l_2)$ operations, computing the Moore–Penrose inverse $(V^T \Omega_2)^\dagger = (V^T \Omega_2)^T (V^T \Omega_2 (V^T \Omega_2)^T)^{-1}$ takes $O(l_1^2 l_2 + l_1^3)$ operations, and forming $B = Y_2 (V^T \Omega_2)^\dagger$ takes $O(ml_1 l_2)$ operations;
- Step 5: Computing the QRCP of $B$ of size $m \times l_1$ takes $O(ml_1^2)$ operations;
- Step 6: Computing the QRCP of $R_0^T$ of size $l_1 \times l_1$ takes $O(l_1^3)$ operations;
- Step 7: Form the $Q, L, P$ factors takes $O(ml_1^2 + nl_1^2 + l_1^3)$ operations.

The total complexity of Algorithm 4 is

$$C_{SPRQLP} = O(ml_1 + nl_2 + mn(l_1 + l_2) + (m + n)l_1 l_2 + (m + n)l_1^2 + l_1^2 l_2 + l_1^3).$$

It is easy to see that the total complexity of Algorithm 3 is

$$C_{RQLP} = O(ml + mnl + nl_1^2 + l_1^3).$$
We can take \( l_1 = l \) and we also see that \( k < l_1 < l_2 \leq \min\{m,n\} \). Then it seems that Algorithm 3 has slightly lower computational complexity than Algorithm 4, where ‘\( \mathcal{O} \)’ is the worst-case scenario growth rate function. However, as stated in [35, 36], the current technology trends indicate an exponential growth in the gap between the communication cost and computation. In consideration of the cost of data communication, the single-pass algorithm is much cheaper than the multi-pass algorithm, especially when the data matrix is generated in the form of stream data. As we know, it is more important to minimize the cost of data communication than to reduce arithmetic operations. In particular, when the data is stored outside the core memory, the cost of data communication is much higher than the algorithm itself. Especially when the data matrix is generated by stream data, only Algorithm 4 can be used since the data can only be accessed once. We note that Algorithm 3 needs to access the data matrix \( A \) at least twice. Therefore, Algorithm 4 may be more efficient than Algorithm 3 in terms of the total cost.

### 4 Error analysis

In this section, we evaluate the performance of the proposed single-pass randomized QLP decomposition algorithm in terms of matrix approximation error and singular value approximation error.

#### 4.1 Matrix approximation error analysis

The following theorem provides some bounds for the matrix approximation error of Algorithm 4 in the spectral norm.

**Theorem 4.1** Let \( A \in \mathbb{R}^{m \times n} \) and \( k \) be the target rank. Suppose \( l_2 \geq l_1 + 4 \). Let \( Q, L, \) and \( P \) be generated by Algorithm 4. Then we have

\[
\|A - QLP^T\|_2 \leq \left(1 + \frac{ae\sqrt{l_2}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{\beta})\right) \sqrt{\frac{k\sigma_1^2(A)\sigma_{k+1}^2(A)C_1^2}{\sigma_{k+1}^2(A)C_1^2 + \sigma_1^2(A)} + \sigma_{k+1}^2(A)}
\]

with probability not less than \( 1 - \Delta_1 - \Delta_2 \) and

\[
\|A - QLP^T\|_F \leq \left(1 + \frac{ae\sqrt{l_1}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{\beta})\right) \sqrt{\frac{k\sigma_1^2(A)\sigma_{k+1}^2(A)C_1^2}{\sigma_{k+1}^2(A)C_1^2 + \sigma_1^2(A)} + \sum_{i=k+1}^{n} \sigma_i^2(A)}
\]

with probability not less than \( 1 - \Delta_1 - \Delta_2 \), where \( \Delta_1 = \alpha^{-(p+1)} + 2 \exp(-\beta^2/2) \), \( \Delta_2 = \alpha^{-(l_2-l_1+1)} + 2 \exp(-\beta^2/2) \) for arbitrary \( \alpha \geq 1 \) and \( \beta \geq 0 \) and

\[
C_1 = \frac{ae\sqrt{l_1}}{p + 1} (\sqrt{m-k} + \sqrt{l_1} + \beta)
\]

for the oversampling parameter \( p = l_1 - k \).
Proof. Firstly, we derive the matrix approximation error bound in the spectral norm. From Algorithm 4 we have

\[ \|A - QLP^T\|_2 = \|A - BV^T\|_2 = \|A - A\Omega_2(V^T\Omega_2)^\dagger V^T\|_2 \]

\[
\leq \|A - AVV^T\|_2 + \|AVV^T - A\Omega_2(V^T\Omega_2)^\dagger V^T\|_2. \tag{4.1}
\]

The second term of the right hand side of (4.1) is reduced to

\[
\|AVV^T - A\Omega_2(V^T\Omega_2)^\dagger V^T\|_2 = \|AVV^T\Omega_2(V^T\Omega_2)^\dagger V^T - A\Omega_2(V^T\Omega_2)^\dagger V^T\|_2 \leq \|A - AVV^T\|_2 \Omega_2(V^T\Omega_2)^\dagger V^T\|_2,
\]

where the first equality follows from the fact that \(V^T\Omega_2\) has full row rank (with the probability of 1), thus \(V^T\Omega_2(V^T\Omega_2)^\dagger = I\) since \(V^T\Omega_2 \in \mathbb{R}^{l \times l_2}\) is a Gaussian random matrix. Thus,

\[
\|A - QLP^T\|_2 \leq (1 + \|\Omega_2(V^T\Omega_2)^\dagger V^T\|_2)\|A - AVV^T\|_2. \tag{4.2}
\]

Let \(A\) admit the SVD defined by \((A.1)\) and \(U^T_m\Omega_1 = [\hat{\Omega}_1^T, \hat{\Omega}_2^T]^T\) with \(\hat{\Omega}_1 \in \mathbb{R}^{k \times l_1}\). For the first term of the right hand side of (4.1), by Lemma A.1 we have

\[
\|A - AVV^T\|_2 = \|A^T - VV^TA^T\|_2 \leq \sqrt{\frac{k\sigma_1^2(A)\sigma_{l_1-p+1}(A)\|\hat{\Omega}_2\|_2^2\|\hat{\Omega}_1\|_2^2}{\sigma_{l_1-p+1}(A)\|\hat{\Omega}_2\|_2^2\|\hat{\Omega}_1\|_2^2 + \sigma_1^2(A)} + \sigma_{k+1}^2(A}). \tag{4.3}
\]

By Lemmas 2.2–2.3 we obtain

\[
\|\hat{\Omega}_2\|_2\|\hat{\Omega}_1\|_2 \leq \frac{\alpha e\sqrt{l_1}}{p + 1}(\sqrt{m - k} + \sqrt{l_1} + \beta) \equiv C_1 \tag{4.4}
\]

with probability not less than \(1 - \alpha^{-(p+1)} - 2 \exp(-\beta^2/2) \equiv 1 - \Delta_1\). Using (4.3) and (4.4) we have

\[
\|A - AVV^T\|_2 \leq \sqrt{\frac{k\sigma_1^2(A)\sigma_{k+1}(A)C_1^2}{\sigma_{k+1}(A)C_1^2 + \sigma_1^2(A)} + \sigma_{k+1}^2(A)} \tag{4.5}
\]

with probability not less than \(1 - \Delta_1\). Furthermore,

\[
\|\Omega_2(V^T\Omega_2)^\dagger V^T\|_2 = \|\Omega_2(V^T\Omega_2)^\dagger\|_2 \leq \|V^T\Omega_2\|_2 \Omega_2\|_2. \tag{4.6}
\]

We already know that a Gaussian matrix is full rank (in the probability of 1) and \(V^T\Omega_2 \in \mathbb{R}^{l \times l_2}\) is a Gaussian random matrix since the Gaussian distribution is rotationally invariant [25]. Suppose that \(l_2 \geq l_1 + 4\). Thus, by Lemma 2.2 we have

\[
\|V^T\Omega_2\|_2 \leq \frac{e\sqrt{l_2}}{l_2 - l_1 + 1}\alpha \tag{4.7}
\]

with probability not less than \(1 - \alpha^{-(l_2-l_1+1)}\). By Lemma 2.3 we have
\[ \| \Omega_2 \|_2 \leq \sqrt{n} + \sqrt{l_2} + \beta \] (4.8)

with probability not less than \( 1 - 2 \exp(-\beta^2/2) \). Substituting (4.7) and (4.8) into (4.6) yields

\[ \| \Omega_2 (V^T \Omega_2)^\dagger V^T \|_2 = \| \Omega_2 (V^T \Omega_2)^\dagger \|_2 \leq \frac{ae \sqrt{l_2}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{l_2} + \beta) \] (4.9)

with probability not less than \( 1 - \alpha^{-1} (l_2 - l_1 + 1) - 2 \exp(-\beta^2/2) \equiv 1 - \Delta_2 \). Plugging (4.5) and (4.9) into (4.2) gives rise to

\[ \| A - QLP^T \|_2 \leq \left( 1 + \frac{ae \sqrt{l_2}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{l_2} + \beta) \right) \sqrt{\frac{k \sigma_1^2(A) \sigma_{k+1}^2(A) C_1^2}{\sigma_{k+1}^2(A) C_1^2 + \sigma_1^2(A)} + \sigma_{k+1}^2(A)} \] (4.10)

with probability not less than \( 1 - \Delta_1 - \Delta_2 \).

Next, we give the matrix approximation error bound in the Frobenius norm. By using the arguments similar to those for the spectral norm, we obtain

\[ \| A - QLP^T \|_F = \| A - BV^T \|_F = \| A - A\Omega_2 (V^T \Omega_2)^\dagger V^T \|_F \leq \| A - AVV^T \|_F + \| AVV^T - A\Omega_2 (V^T \Omega_2)^\dagger V^T \|_F. \] (4.11)

The second term of the right hand side of (4.11) is reduced to

\[ \| AVV^T - A\Omega_2 (V^T \Omega_2)^\dagger V^T \|_F = \| AVV^T \Omega_2 (V^T \Omega_2)^\dagger V^T - A\Omega_2 (V^T \Omega_2)^\dagger V^T \|_2 \leq \| \Omega_2 (V^T \Omega_2)^\dagger V^T \|_2 \| A - AVV^T \|_F, \]

where the first equality follows from the fact that the Gaussian matrix \( V^T \Omega_2 \) has full row rank and thus \( V^T \Omega_2 (V^T \Omega_2)^\dagger = I \). Thus,

\[ \| A - QLP^T \|_F \leq (1 + \| \Omega_2 (V^T \Omega_2)^\dagger V^T \|_2) \| A - AVV^T \|_F. \] (4.12)

By using the arguments similar to those of (4.5) we have

\[ \| A - AVV^T \|_F = \| A^T - VV^T A \|_F \leq \sqrt{\frac{k \sigma_1^2(A) \sigma_{k+1}^2(A) C_1^2}{\sigma_{k+1}^2(A) C_1^2 + \sigma_1^2(A)} + \sum_{i=k+1}^n \sigma_i^2(A)} \] (4.13)

with probability not less than \( 1 - \Delta_1 \). Substituting (4.13) and (4.9) into (4.12) yields

\[ \| A - QLP^T \|_F \leq \left( 1 + \frac{ae \sqrt{l_2}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{l_2} + \beta) \right) \sqrt{\frac{k \sigma_1^2(A) \sigma_{k+1}^2(A) C_1^2}{\sigma_{k+1}^2(A) C_1^2 + \sigma_1^2(A)} + \sum_{i=k+1}^n \sigma_i^2(A)} \]

with probability not less than \( 1 - \Delta_1 - \Delta_2 \). \( \square \)

On the error bounds in Theorem 4.1, we have the following comments. For the error bound in the spectral norm, we can find that the upper bound is of the order
of $\sigma_{k+1}(A)$ with high probability, where the constant is related to the parameters for the smallest singular spectral distribution of the random matrices $\Omega_1$ and $\Omega_2$. This shows that Algorithm 4 provides a low-rank approximation to $A$. We have the similar comments for the error bound in the Frobenius norm.

### 4.2 Singular value approximation error analysis

In [44], Wu and Xiang gave the lower bounds on the singular values of the matrix $L$ as a whole to track the singular values of the data matrix $A$ approximately. In this subsection, we not only provide a lower bound and an upper bound on the singular values of the matrix $L$ to approximately track the singular values of the data matrix $A$ but also give the error analysis between the singular values of the diagonal blocks of $L$ and that of $A$, which hold with high probability.

We first give some lemmas on the error bounds for singular values of the QLP decomposition for a data matrix. As noted in [20], for the QLP decomposition, the pivoting in the first QR decomposition is essential while the pivoting of the second QR decomposition is only necessary to avoid “certain contrived counterexamples”. Therefore, in order to simplify the analysis, we assume that there is no pivoting in the second QR decomposition of the QLP decomposition for the matrix $B$ generated by Algorithm 4.

The following lemma gives a bound for the maximum singular value approximation error.

**Lemma 4.2** [22, Theorem 3.3] Let $A \in \mathbb{R}^{m \times n}$ and $\sigma_1(A) > \sigma_2(A)$. Let $R_0$ be the $R$-factor in the pivoted QR factorization of $A$, $AP_0 = Q_0R_0$ and let $L^T$ be the $R$-factor in the unpivoted QR factorization of $R_0^T$, $R_0^T = Q_1L^T$. Partition $R_0$ and $L$ as

$$R_0 = \begin{bmatrix} r_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix},$$

where $r_{11}, l_{11} \in \mathbb{R}$. If $\|R_{22}\|_2 \leq \sqrt{2(n-1)}\sigma_2(A)$ and $\frac{\|R_{22}\|_2}{|r_{11}|} < 1$, then

$$|l_{11}^{-1} - \sigma_1^{-1}(A)| \leq \frac{\sigma_2^2(A)}{\sigma_1^3(A)} \left( \frac{n^2 \|R_{12}\|_2^2}{(1 - r_{11}^2)|l_{11}|^2} \right),$$

where $\rho_1 = \frac{\|L_{22}\|_2}{|l_{11}|}$.

The following lemma provides some bounds for the interior singular value approximation errors.

**Lemma 4.3** [22, Theorem 3.4] Let $A \in \mathbb{R}^{m \times n}$ and $\sigma_k(A) > \sigma_{k+1}(A)$. Let $R_0$ be the $R$-factor in the pivoted QR factorization of $A$, $AP_0 = Q_0R_0$ and let $L^T$ be the $R$-factor in the unpivoted QR factorization of $R_0^T$, $R_0^T = Q_1L^T$. Partition $R_0$ and $L$ as
\[ R_0 = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \]

where \( R_{11}, L_{11} \in \mathbb{R}^{k \times k} \). If \( \|R_{22}\|_2 \leq \sqrt{(k+1)(n-k)} \cdot \sigma_{k+1}(A) \), \( \sigma_k(R_{11}) \geq \frac{\sigma_k(A)}{\sqrt{k(n-k+1)}} \), and \( \frac{\|R_{22}\|}{\sigma_k(R_{11})} < 1 \), then for \( j = 1, \ldots, n-k \),

\[
\frac{\sigma_j(L_{22}) - \sigma_{k+j}(A)}{\sigma_{k+j}(A)} \leq \left( \frac{\sigma_{k+j}(A)}{\sigma_k(A)} \right)^2 \mathcal{O} \left( \frac{n^2 \|R_{12}\|^2}{1 - \rho_1^2} \sigma_k^2(L_{11}) \right)
\]

and for \( j = 1, \ldots, k \),

\[
\frac{\sigma_j^{-1}(L_{11}) - \sigma_j^{-1}(A)}{\sigma_j^{-1}(A)} \leq \left( \frac{\sigma_{k+j}(A)}{\sigma_k(A)} \right)^2 \mathcal{O} \left( \frac{n^2 \|R_{12}\|^2}{1 - \rho_1^2} \sigma_k^2(L_{11}) \right)
\]

where \( \rho_1 = \frac{\|R_{22}\|}{\sigma_k(R_{11})} \).

Now we are ready to state our first theorem on the singular value approximation error of Algorithm 4.

Let \( B \) be generated by Algorithm 4. Then we can rewrite \( B \) as

\[
B = AV + AV = B - AV + AV.
\]

It follows from Lemma A.2 in Appendix that for any \( 1 \leq j \leq l_1 \),

\[
|\sigma_j(B) - \sigma_j(AV)| \leq \sigma_1(B - AV).
\]

Thus, for any \( 1 \leq j \leq l_1 \),

\[
\sigma_j(AV) - \sigma_1(B - AV) \leq \sigma_j(B) \leq \sigma_j(AV) + \sigma_1(B - AV).
\]

It is easy to see that \( \sigma_j(B) = \sigma_j(L) \) for all \( j = 1, \ldots, l_1 \). Thus, for any \( 1 \leq j \leq l_1 \),

\[
\sigma_j(AV) - \sigma_1(B - AV) \leq \sigma_j(L) \leq \sigma_j(AV) + \sigma_1(B - AV).
\]  \hfill (4.14)

The following singular value approximation error analysis only applies to the situation where the numerical rank of \( A \) is clearly defined, i.e., there exists a clear gap in the singular values of \( A \). This is reasonable since the data matrix often exhibits the nature of numerical low rank in data science and scientific computing \[48\].

On the singular value approximation error of Algorithm 4, we have the following result.

**Theorem 4.4** With the notation of Algorithm 4, calculate the pivoted QLP decomposition of \( B \) such that \( R_0 \) is the R-factor in the pivoted QR factorization of \( B \), \( BP_0 = Q_0 R_0 \) and \( L^T \) is the R-factor in the unpivoted QR factorization of \( R_0^T \), \( R_0^T = Q_1 L^T \). Then
\[ \sigma_j(L) \leq \sigma_j(A) + C \]

with probability not less than \(1 - \Delta_1 - \Delta_2\) for all \(j = 1, \ldots, k\) and

\[ \sigma_j(L) \geq \frac{\sigma_j(A)}{\delta_j} - C \]

with probability not less than \(1 - 2\Delta_1 - \Delta_2\) for all \(j = 1, \ldots, k\), where

\[
C = \frac{\alpha e \sqrt{\frac{L}{2}}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{L_2} + \beta) \sqrt{\frac{k\sigma_1^2(A)\sigma_{k+1}^2(A)C_1^2}{\sigma_{k+1}^2(A)C_1^2 + \sigma_1^2(A)} + \sigma_{k+1}^2(A)},
\]

\[
\delta_j = \sqrt{1 + C_1^2 \left( \frac{\sigma_{k+1}(A)}{\sigma_j(A)} \right)^2}
\]

with \(\alpha \geq 1, \beta \geq 0\) and \(C_1, \Delta_1, \Delta_2\) being defined as in Theorem 4.1.

**Proof** By Lemma A.3 we obtain

\[ \sigma_j(AV) = \sigma_j(V^TA^T) \leq \sigma_j(A^T) = \sigma_j(A), \quad \forall 1 \leq j \leq k. \quad (4.15) \]

Using (4.5) and (4.9) we have

\[
\sigma_1(B - AV) = \|A\Omega_2(V^T\Omega_2)^\dagger - AVV^T\Omega_2(V^T\Omega_2)^\dagger\|_2 \leq \|\Omega_2(V^T\Omega_2)^\dagger\|_2 \|A - AVV^T\|_2
\]

\[
\leq \frac{\alpha e \sqrt{\frac{L}{2}}}{l_2 - l_1 + 1} (\sqrt{n} + \sqrt{L_2} + \beta) \sqrt{\frac{k\sigma_1^2(A)\sigma_{k+1}^2(A)C_1^2}{\sigma_{k+1}^2(A)C_1^2 + \sigma_1^2(A)} + \sigma_{k+1}^2(A)} \equiv C
\]

(4.16)

with probability not less than \(1 - \Delta_1 - \Delta_2\). From (4.14), (4.15), and (4.16) we have

\[ \sigma_j(L) \leq \sigma_j(AV) + \sigma_1(B - AV) \leq \sigma_j(A) + C \]

with probability not less than \(1 - \Delta_1 - \Delta_2\) for all \(j = 1, \ldots, k\).

On the other hand, let \(\hat{B} = AV\) and let \(A\) admit the SVD defined by (A.1). Let \(U_m^T\Omega_1^T = [\hat{\Omega}_1^T, \hat{\Omega}_2^T]^T\), where \(\hat{\Omega}_1 \in \mathbb{R}^{k \times l}\) and \(\hat{\Omega}_2 \in \mathbb{R}^{(m-k) \times l}\). According to Lemma A.4, we obtain

\[ \sigma_j(AV) = \sigma_j(\hat{B}) \geq \frac{\sigma_j(A)}{\sqrt{1 + ||\hat{\Omega}_2||_2^2 ||\hat{\Omega}_1||_2^2 \left( \frac{\sigma_{k+1}(A)}{\sigma_j(A)} \right)^2}}, \]

for all \(j = 1, \ldots, k\). Using (4.4) we have \(||\hat{\Omega}_2||_2 \leq C_2^2 ||\hat{\Omega}_1||_2^2 \leq C^2 \) with probability not less than \(1 - \Delta_1\). Then we have \(\sigma_j(AV) \geq \sigma_j(A)/\delta_j \) with probability not less than \(1 - \Delta_1\).

This, together with (4.14) and (4.16), yields \(\sigma_j(L) \geq \sigma_j(A)/\delta_j - C\) with probability not less than \(1 - 2\Delta_1 - \Delta_2\) for all \(j = 1, \ldots, k\). \(\square\)
Theorem 4.4 show that, if there exists a gap between $\sigma_k(A)$ and $\sigma_{k+1}(A)$, then with high probability, the singular values of the matrix $L$ generated by Algorithm 4 can approximate the first $k$ largest singular values of $A$ with a high accuracy.

Next, we give the following theorem on the maximum singular value approximation error of Algorithm 4.

Theorem 4.5 With the notation of Algorithm 4, calculate the pivoted QLP decomposition of $B$ such that $R_0$ is the $R$-factor in the pivoted QR factorization of $B$, $BP_0 = Q_0R_0$ and $L^T$ is the $R$-factor in the unpivoted QR factorization of $R_0^T$, $R_0^T = Q_1L^T$. Partition $R_0$ and $L$ as

$$R_0 = \begin{bmatrix} r_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix},$$

where $r_{11}, l_{11} \in \mathbb{R}$. Suppose $\sigma_1(B) > \sigma_2(B)$, $\|R_{22}\|_2 \leq \sqrt{2(l_1 - 1)\sigma_2(B)}$, and $\frac{\|R_{22}\|_2}{|r_{11}|} < 1$. Then

$$|l_{11}|^{-1} - \sigma_1^{-1}(A) \leq \sigma_1^{-1}(B) \left(1 - \frac{1}{\delta_1} \cdot \frac{1}{1 + C\sigma_1^{-1}(B)}\right) + \frac{\sigma_2^2(B)}{\sigma_1^3(B)} \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2)|l_{11}|^2} \quad (4.17)$$

with probability not less than $1 - 2\Delta_1 - \Delta_2$, where $\rho_1 = \frac{\|L_{12}\|_2}{|l_{11}|}$, $C$ and $\delta_1$ are defined as in Theorem 4.4 with $\alpha \geq 1$, $\beta \geq 0$ and $C_1, \Delta_1, \Delta_2$ being defined as in Theorem 4.1.

Proof By Theorem 4.4 we have

$$\sigma_1(B) = \sigma_1(L) \geq \frac{\sigma_1(A)}{\delta_1} - C,$$

i.e.,

$$\frac{1}{\sigma_1(A)} \geq \frac{1}{\delta_1\sigma_1(B) + C\delta_1} \quad (4.17)$$

with probability not less than $1 - 2\Delta_1 - \Delta_2$. By Lemma 4.2 we obtain

$$|l_{11}|^{-1} \leq \sigma_1^{-1}(B) + \frac{\sigma_2^2(B)}{\sigma_1^3(B)} \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2)|l_{11}|^2} \cdot$$

Subtracting $\sigma_1^{-1}(A)$ from both sides of the above inequality gives rise to
\[
\left| l_{11} \right|^{-1} - \sigma_1^{-1}(A) \leq \sigma_1^{-1}(B) - \frac{1}{\sigma_1(A)} + \frac{\sigma_2^2(B)}{\sigma_1^2(B)} \mathcal{O} \left( \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2) |l_{11}|^2} \right)
\]

\[
\leq \sigma_1^{-1}(B) - \frac{1}{\delta_1 \sigma_1(B) + C\delta_1} + \frac{\sigma_2^2(B)}{\sigma_1^2(B)} \mathcal{O} \left( \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2) |l_{11}|^2} \right)
\]

\[
= \sigma_1^{-1}(B) \left( 1 - \frac{1}{\delta_1} \cdot \frac{1}{1 + C\sigma_1^{-1}(B)} \right) + \frac{\sigma_2^2(B)}{\sigma_1^2(B)} \mathcal{O} \left( \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2) |l_{11}|^2} \right)
\]

with probability not less than \(1 - 2\Delta_1 - \Delta_2\), where the second inequality follows from (4.17).

Finally, on the interior singular value approximation errors of Algorithm 4, we have the following theorem.

**Theorem 4.6** With the notation of Algorithm 4, calculate the pivoted QLP decomposition of B such that \(R_0\) is the R-factor in the pivoted QR factorization of B, \(B_{P_0} = Q_0 R_0\) and \(L^T\) is the R-factor in the unpivoted QR factorization of \(R_0^T\), \(R_0^T = Q_1 L^T\). Suppose \(\sigma_s(B) > \sigma_{s+1}(B)\) for some \(1 \leq s < k\). Partition \(R_0\) and \(L\) as

\[
R_0 = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix},
\]

where \(R_{11}, L_{11} \in \mathbb{R}^{s \times s}\). Assume that \(\|R_{22}\|_2 \leq \sqrt{(s + 1)(1 - s)}\sigma_{s+1}(B)\), \(\sigma_s(R_{11}) \geq \frac{\sigma_s(B)}{\sqrt{s(l_{11} - s + 1)}}, \) and \(\frac{\|R_{22}\|_2}{\sigma_s(L_{11})} < 1\). Then

\[
\frac{\sigma_j^{-1}(L_{11}) - \sigma_j^{-1}(A)}{\sigma_j^{-1}(A)} \leq \delta_j - 1 + \frac{C\delta_j}{\sigma_j(B)} + \left( \frac{C\delta_j}{\sigma_j(B)} + \delta_j \right) \left( \frac{\sigma_{s+1}(B)}{\sigma_j(B)} \right)^2 \mathcal{O} \left( \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2)\sigma_j^2(L_{11})} \right)
\]

with probability not less than \(1 - 2\Delta_1 - \Delta_2\) for all \(j = 1, \ldots, s\),

\[
\frac{\sigma_j(L_{22}) - \sigma_{s+j}(A)}{\sigma_{s+j}(A)} \leq \frac{C}{\sigma_{s+j}(B) - C} \left( \frac{\sigma_{s+1}(B)}{\sigma_j(B)} \right)^2 \mathcal{O} \left( \frac{l_1^2 \|R_{12}\|_2^2}{(1 - \rho_1^2)\sigma_j^2(L_{11})} \right)
\]

with probability not less than \(1 - \Delta_1 - \Delta_2\) for all \(j = 1, \ldots, k - s\), where \(\rho_1 = \frac{\|L_{22}\|_2}{\sigma_j(L_{11})}\), \(C\) and \(\delta_j\) are defined as in Theorem 4.4 with \(\alpha \geq 1, \beta \geq 0\) and \(C_1, \Delta_1, \Delta_2\) being defined as in Theorem 4.1.
Proof From Theorem 4.4, we know that $\sigma_j(B) \leq \sigma_j(A) + C$, i.e., $1 / \sigma_j(A) \leq 1 / (\sigma_j(B) - C)$ with probability not less than $1 - \Delta_1 - \Delta_2$ for all $j = 1, \ldots, k$. Similarly, we have $\sigma_j(B) \geq \sigma_j(A) / \delta_j - C$, i.e., $\sigma_j(A) \leq C \delta_j + \delta_j \sigma_j(B)$ with probability not less than $1 - 2 \Delta_1 - \Delta_2$ for all $j = 1, \ldots, k$. Using Lemma 4.3 we have, for $j = 1, \ldots, k - s$,

$$\sigma_j(L_{22}) - \sigma_{s+j}(A) \leq \sigma_{s+j}(B) - \sigma_{s+j}(A) + \frac{\sigma_{s+1}^2(B) \sigma_{s+j}(B)}{\sigma_s^2(B)} \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right).$$

Dividing $\sigma_{s+j}(A)$ on both sides of the above inequality yields

$$\frac{\sigma_j(L_{22}) - \sigma_{s+j}(A)}{\sigma_{s+j}(A)} \leq \frac{\sigma_{s+j}(B) - \sigma_{s+j}(A)}{\sigma_{s+j}(A)} + \frac{\sigma_{s+1}^2(B) \sigma_{s+j}(B)}{(\sigma_{s+j}(B) - C) \sigma_s^2(B)} \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right)$$

$$\leq \frac{C}{\sigma_{s+j}(B) - C} + \left( 1 + \frac{C}{\sigma_{s+j}(B) - C} \right)^2 \left( \frac{\sigma_{s+1}(B)}{\sigma_s(B)} \right)^2 \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right)$$

with probability not less than $1 - \Delta_1 - \Delta_2$ for all $j = 1, \ldots, k - s$.

By Lemma 4.3 again we have, for $j = 1, \ldots, s$,

$$\sigma_j^{-1}(L_{11}) - \sigma_j^{-1}(A) \leq \sigma_j^{-1}(B) - \sigma_j^{-1}(A) + \frac{\sigma_{s+1}^2(B)}{\sigma_s^2(B) \sigma_j(B)} \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right).$$

Dividing $\sigma_j^{-1}(A)$ on both sides of the above inequality yields

$$\frac{\sigma_j^{-1}(L_{11}) - \sigma_j^{-1}(A)}{\sigma_j^{-1}(A)} \leq \frac{\sigma_j^{-1}(B) - \sigma_j^{-1}(A)}{\sigma_j^{-1}(A)} + \frac{\theta_j(A) \sigma_{s+1}^2(B)}{\sigma_j^2(B) \sigma_j(B)} \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right)$$

$$\leq \frac{\sigma_j^{-1}(B) - (\delta_j \sigma_j(B) + C \delta_j)^{-1}}{(\delta_j \sigma_j(B) + C \delta_j)^{-1}} + \frac{(\delta_j \sigma_j(B) + C \delta_j) \sigma_{s+1}^2(B)}{\sigma_s^2(B) \sigma_j(B)} \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right)$$

$$= \frac{\delta_j \sigma_j(B) + C \delta_j}{\sigma_j(B)} - 1 + \frac{\delta_j \sigma_j(B) + C \delta_j}{\sigma_j(B)} \left( \frac{\sigma_{s+1}(B)}{\sigma_s(B)} \right)^2 \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right)$$

$$= \delta_j - 1 + \frac{C \delta_j}{\sigma_j(B)} + \left( \frac{C \delta_j}{\sigma_j(B)} + \delta_j \right)^2 \left( \frac{\sigma_{s+1}(B)}{\sigma_s(B)} \right)^2 \Theta \left( \frac{l_1^5 \| R_{12} \|_2^2}{(1 - \rho_1^2) \sigma_s^2(L_{11})} \right)$$
with probability not less than \(1 - 2\Delta_1 - \Delta_2\) for all \(j = 1, \ldots, s\).

**Corollary 4.7** Under the same assumptions of Theorem 4.6, if \(s > k\), then

\[
\frac{\sigma_j^{-1}(L_{11}) - \sigma_j^{-1}(A)}{\sigma_j^{-1}(A)} \leq \delta_j - 1 + \frac{C\delta_j}{\sigma_k(B)}
\]

\[
+ \left( \frac{C\delta_j}{\sigma_k(B)} + \delta_j \right) \left( \frac{\sigma_{s+1}(B)}{\sigma_s(B)} \right)^2 \mathcal{O}\left( \frac{\|R_{12}\|_2}{(1 - \rho_1^2)\sigma_k^2(L_{11})} \right)
\]

with probability not less than \(1 - 2\Delta_1 - \Delta_2\) for all \(j = 1, \ldots, s\). In particular, if \(s = k\), then

\[
\frac{\sigma_j^{-1}(L_{11}) - \sigma_j^{-1}(A)}{\sigma_j^{-1}(A)} \leq \delta_j - 1 + \frac{C\delta_j}{\sigma_k(B)}
\]

\[
+ \left( \frac{C\delta_j}{\sigma_k(B)} + \delta_j \right) \left( \frac{\sigma_{k+1}(B)}{\sigma_k(B)} \right)^2 \mathcal{O}\left( \frac{\|R_{12}\|_2}{(1 - \rho_1^2)\sigma_k^2(L_{11})} \right)
\]

with probability not less than \(1 - 2\Delta_1 - \Delta_2\) for all \(j = 1, \ldots, k\).

Theorem 4.5 shows that with high probability, the (1, 1) entry of the matrix \(L\) generated by Algorithm 4 provides a desired approximation to \(\sigma_1(A)\) if there exists a gap between \(\sigma_k(A)\) and \(\sigma_{k+1}(A)\) and \(\sigma_1(B) > \sigma_2(B)\). Theorem 4.6 and Corollary 4.7 show that, instead of the individual diagonal element approximating, the singular values of the diagonal blocks of \(L\) can accurately approximate the \(k\) largest singular values of \(A\) with high probability when there is a gap between \(\sigma_k(A)\) and \(\sigma_{k+1}(A)\). We must point out that these singular value error bounds are established under the assumption that there exist a gap in the singular values of \(A\). However, the later numerical experiments show that the proposed algorithm works well even the gap is not clear.

## 5 Numerical experiments

In this section, we give some numerical examples to illustrate the effectiveness of Algorithm 4. We also compare our algorithm with Algorithms 2–3. All experiment are performed by using MATLAB 2019b on a personal laptop with an Intel(R) CPU i5-10210U of 1.6 GHz and 8 GB of RAM.

**Example 5.1** [40](synthetic input matrix) Let \(A = U_n \Sigma V_n^T \in \mathbb{R}^{n \times n}\), where \(U_n, V_n \in \mathbb{R}^{n \times n}\) are orthogonal matrices generated by using the built-in functions `rand` and `orth` and \(\Sigma\) is given as follows:
• Polynomially decaying spectrum (pds):
  \[ \Sigma = \text{diag}(1, \ldots, 1, 2^{-s}, 3^{-s}, \ldots, (n-t+1)^{-s}) ; \]

• Exponentially decaying spectrum (eds):
  \[ \Sigma = \text{diag}(1, \ldots, 1, 2^{-r}, 2^{-2s}, \ldots, 2^{-(n-r)s}) . \]

Here, the constants \( t, s > 0 \) control the rank of the significant part of the matrix and the rate of decay, respectively. We report our numerical results for \( n = 2000 \).

Example 5.2 [49]: (ill-conditioned matrix) The ill-conditioned matrix \( A \) is generated by discretization of the Fredholm integral equation of the first kind with a square integrable kernel:

\[
\int_{z_1}^{z_2} K(y, z)f(z)dz = g(y), \quad y_1 \leq y \leq y_2,
\]

where \( y_1, y_2, z_1, \) and \( z_2 \) are some constants, from the regularization tool package\(^1\). We use the MATLAB functions heat and deriv2 to generate two ill-conditioned matrices, which are respectively discretized by means of simple quadrature (midpoint rule) and the Galerkin method with orthonormal box function as default. On how far the discrete problem deviates from the continuous one, one may refer to [50, 51]. The integral kernel \( K \) of heat is \( K(y, z) = K(y - z) \) with \( K(z) = \frac{e^{-z^2/2}}{2\sqrt{\pi}} \exp(1/(4\kappa^2 z)) \), where the parameter \( \kappa \) controls the ill-conditioning of the discrete matrix \( A \). The integral kernel \( K \) of deriv2 is \( K(y, z) = y(z - 1) \) if \( y < z \) and \( K(y, z) = z(y - 1) \) if \( y \geq z \). We take \( \kappa = 1 \) and the size of the test matrices is set to be \( m = n = 2000 \).

In Algorithms 2–4, the output low-rank representation of \( A \) is \( \hat{A} = QLP^T \) with the target rank \( k \), where \( Q, L \) and \( P \) are the output matrices of Algorithms 2–4, respectively. Then the matrix approximation error is given by

\[
E_F = \frac{\|A - \hat{A}\|_F}{\|A\|_F}.
\]

and the absolute and relative singular value approximation errors can be measured by

\[
AE_{\sigma_j} = |\sigma_j(A) - |l_{jj}|| \quad \text{and} \quad RE_{\sigma_j} = \frac{|\sigma_j(A) - |l_{jj}||}{\sigma_j(A)} \quad \forall 1 \leq j \leq k.
\]

The parameters used in Algorithms 2–4 are listed in Table 1.

\(^1\) http://www.imm.dtu.dk/~pcha/Regutools/.

Springer
5.1 Comparison of running time

Figure 1 shows the running time of Algorithms 2–4 with different target ranks for Examples 5.1–5.2. We can find from Fig. 1 that Algorithms 3–4 are always faster than Algorithm 2 and Algorithm 4 is almost as efficient as Algorithm 3. Due to the limitation of conditions, we cannot measure the data communication time and thus the running time in Fig. 1 does not include the data communication time. However, under current technology trends, the cost of data communication is the main factor restricting the efficiency of an algorithm. Therefore, when the problem involves large matrices, the vast majority of the running time is spent on the data communication. When we take the data communication into account, Algorithm 4 may be
much more efficient than Algorithm 3. In particular, among the three algorithms, only our algorithm can be used to the case of stream data.

5.2 Comparison of matrix approximation error

Figure 2 shows the trend of matrix approximation error of Algorithms 2–4 with different target rank for Examples 5.1–5.2. We observe from Fig. 2 that, in terms of matrix approximation error, Algorithms 2–4 show good performance for four numerical examples, where the error decreases with the increase of matrix rank. This is consistent with our theoretical analysis. We also see that Algorithms 2–3 behavior slightly better than Algorithm 4 for most cases. This is reasonable since the row-column sampling is employed in Algorithm 4.

Compared with Algorithms 2–3, Algorithm 4 only needs to access the data matrix once. In spite of its slight loss of matrix approximation quality, Algorithm 4 can greatly reduces the total cost if the data communication cost is dominant and it is more suitable for the case that the data matrix is stored outside the core memory or is generated in the form of streaming data.

Fig. 2 Matrix approximation error for different test matrices
5.3 Comparison of singular value approximation error

In Figs. 3, 4, we plot the curves of singular value absolute error, singular value relative error of Algorithms 2–4 for Examples 5.1–5.2 accordingly, where the target rank is set to be \( k = 860 \). We point out that, in Fig. 4b, taking the index \( j = 250 \) as the dividing line, the singular values at the tail of the data matrix \( A \) are less than the machine precision. This makes the singular values relative error increase sharply and the second half of the figure can not display other information. Therefore, we do some processing on the figure, i.e., we truncate the x-axis at index \( j = 250 \).

We see from Figs. 3, 4 that Algorithm 4 works more effectively in terms of absolute and relative singular value error. For most cases, the singular value approximation error of Algorithm 4 is close to Algorithms 2–3, and even better for some cases. In particular, in Figs. 3a, d and 4a, d, we see that Algorithm 4 performs much better than Algorithm 3 for rapidly decayed singular values without a clear gap.

The above experiments on the singular value approximation error show that Algorithm 4 provides a very effective, fast and inexpensive way for the singular value approximation. This also shows that Algorithm 4 is an efficient method for
the numerical rank estimation since the numerical rank estimation depends on the estimation of singular values.

5.4 Application to 2D integral kernels

To further illustrate the efficiency of the proposed algorithm, we consider the following 2D integral kernels.

**Example 5.3** [52] We consider the 2D nuclear magnetic resonance relaxometry problem, which is modeled by the following Fredholm integral equation of the first kind

$$\int_0^{\tilde{T}_2} \int_0^{\tilde{T}_1} K(\tau^1, \tau^2, T^1, T^2)f(T^1, T^2)dT^1dT^2 = g(\tau^1, \tau^2),$$

where $g(\tau^1, \tau^2)$ is the noiseless signal as a function of experiment times $(\tau^1, \tau^2)$, and $f(T^1, T^2)$ is the density distribution function. The kernel is given by
By using the midpoint quadrature rule with logarithmically equispaced nodes from the IR Tool package\(^2\) as the na49 package, we get the discrete ill-posed problem \(A\tilde{x} \approx \tilde{b}\) via \([A, b, \tilde{x}, \text{ProbInfo}] = \text{PRnmr}(n)\), where \(n\) is the size of the relaxation time distribution to be recovered, and the coefficient matrix \(A\) is ill-conditioned matrix and all its singular values gradually decay to zero, \(\tilde{x}\) is the true solution, and \(\tilde{b}\) is the measured data. For simplicity, we set \(n = 48\) and the size of \(A\) is \(9216 \times 2304\).

Figure 5 shows the approximation error for Example 5.3 by computing a low-rank approximation to \(A\) via Algorithms 2–4. We see from Fig. 5 that Algorithm 4 works effectively in terms of both the matrix approximation error and the singular value approximation error.

Figure 5 shows the approximation error for Example 5.3 by computing a low-rank approximation to \(A\) via Algorithms 2–4. We see from Fig. 5 that Algorithm 4 works effectively in terms of both the matrix approximation error and the singular value approximation error.

Figure 6 shows the computed solution to \(QLP^T x \approx b\) in the least squares sense (i.e., \(x = PL^{-1}Q^T b\)), where the low-rank approximation \(QLP^T\) is obtained by

\(^2\) http://www.netlib.org/nmralgo/.
Algorithm 4 for different target rank $k$. We observe from Fig. 6 that the computed solution become closer to the true solution as $k$ increases. This shows that our algorithm is competitive if the target rank is chosen appropriately.

5.5 Selection of parameter $l_2$

Figure 7 displays the approximation error for Example 5.1 $\text{pds}(t = 30, s = 2)$ with different values of $l_2$. We find from Fig. 7 that the approximation accuracy becomes lower as $l_2$ tends to $l_1$. This shows that the parameter $l_2$ is of great importance to the approximation accuracy. Figure 7 indicates that it is acceptable to choose $l_2 = 1.5k$. For more discussion on sampling parameters of single-pass algorithms, one may refer to [39].
6 Conclusions

In this paper, we have proposed a single-pass randomized QLP decomposition algorithm, which provides a low-rank approximation of a data matrix. We also derive the bounds for the matrix approximation error and the singular value approximation errors, which hold with high probability. Numerical experiments are reported to confirm our theoretical results.

Appendix A

In this appendix, we give some preliminary results on Gaussian random matrices. In what follows, we assume that all Gaussian random matrices have full rank. We first recall the following lemmas.

Lemma A.1 [29, Theorem 4.4] Let $A \in \mathbb{R}^{m \times n}$, whose SVD is given by

$$A = U_m \Sigma V_n^T,$$

where $\Sigma = \text{diag}(\sigma_1(A), \ldots, \sigma_q(A)) \in \mathbb{R}^{m \times n}$, $U_m \in \mathbb{R}^{m \times m}$ and $V_n \in \mathbb{R}^{n \times n}$ are two orthogonal matrices. Suppose $k, l$, and $p$ are three positive integers such that $k \leq l$ and $0 \leq p \leq l - k$. Let $\Omega \in \mathbb{R}^{m \times l}$ be a standard Gaussian random matrix and the reduced QR decomposition of $A \Omega$ be given by $A \Omega = Q_\Omega R$, where $Q \in \mathbb{R}^{m \times l}$ has orthonormal columns and $R \in \mathbb{R}^{l \times l}$ is upper triangular. Let $V_n^T \Omega = [\hat{\Omega}_1^T, \hat{\Omega}_2^T]$, $\hat{\Omega}_1 \in \mathbb{R}^{k \times l}$. If $\hat{\Omega}_1$ has full row rank, then

$$\|A - QQ^T A\|_F \leq \sqrt{\frac{k \sigma_1^2(A) \sigma_{l-p+1}(A) \|\hat{\Omega}_2\|_2^2 \|\hat{\Omega}_1^\dagger\|_2^2}{\sigma_{l-p+1}(A) \|\hat{\Omega}_2\|_2^2 \|\hat{\Omega}_1^\dagger\|_2^2 + \sigma_1^2(A)}} + \sum_{i=k+1}^n \sigma_i^2(A),$$

Fig. 7 Approximation errors for different choices of oversampling parameter $l_2$
The following lemmas provide some inequalities about singular values, which are helpful to prove the main theorem on the singular value approximation error of our algorithm.

**Lemma A.2** [53, Theorem 3.3.16] Let $G, \Delta G \in \mathbb{R}^{m \times n}$. Then we have $|\sigma_i(G + \Delta G) - \sigma_i(G)| \leq \sigma_1(\Delta G)$ for $i = 1, \ldots, q = \min\{m, n\}$.

**Lemma A.3** [18] Let $A \in \mathbb{R}^{m \times n}$. Suppose $Q \in \mathbb{R}^{m \times l}$ is a matrix with orthonormal columns. Then $\sigma_j(A) \geq \sigma_j(Q^T A)$ for $j = 1, \ldots, l$.

**Lemma A.4** [29, Theorem 4.3] Let $A \in \mathbb{R}^{m \times n}$ admit the SVD as in (A.1). Suppose $k, l, p$ are three positive integers such that $0 \leq p \leq l - k$. Let $\Omega \in \mathbb{R}^{n \times l}$ be a standard Gaussian matrix. Partition $V_n^T \Omega = [\hat{\Omega}_1^T, \hat{\Omega}_2^T]^T$ with $\hat{\Omega}_1 \in \mathbb{R}^{k \times l}$. Let $B = Q^T A$ with $Q \in \mathbb{R}^{m \times l}$ being a matrix with orthonormal columns. If the matrix $\hat{\Omega}_1$ has full row rank, then

$$\sigma_j(Q^T A) = \sigma_j(B) \geq \frac{\sigma_j(A)}{\sqrt{1 + \|\hat{\Omega}_2\|^2_2 \|\hat{\Omega}_1\|^2_2 \left( \frac{\sigma_{l-p+1}(A)}{\sigma_l(A)} \right)^2}}$$

for $j = 1, \ldots, l$.

**Acknowledgements** The authors would like to thank the editor and the referees for their valuable comments.

**Data Availability** The datasets analysed during the current study were derived from the following public resources: [http://www.imm.dtu.dk/~pcha/Regutools/](http://www.imm.dtu.dk/~pcha/Regutools/); [http://www.netlib.org/numeralgo/](http://www.netlib.org/numeralgo/).

** Declarations**

**Conflict of Interest Statement** The authors declare that they have no conflict of interest.

**References**

1. Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag, New York (1986)
2. Rokhlin, V., Szlam, A., Tygert, M.: A randomized algorithm for principal component analysis. SIAM J. Matrix Anal. Appl. 31, 1100–1124 (2009)
3. Halko, N., Martinsson, P.-G., Shkolnisky, Y., Tygert, M.: An algorithm for the principal component analysis of large data sets. SIAM J. Sci. Comput. 33, 2580–2594 (2011)
4. Feng, X., Xie, Y., Song, M., Yu, W., Tang, J.: Fast randomized PCA for sparse data. In Proc. ACML 95, 710–725 (2018)
5. M. Mahoney, Randomized algorithms for matrices and data, preprint, arXiv:1104.5557, (2011)
6. Drineas, P.: RandNLA: randomized numerical linear algebra. Commun. ACM 59, 80–90 (2016)
7. Li, K., Wu, G.: A randomized generalized low rank approximations of matrices algorithm for high dimensionality reduction and image compression. Numer. Linear Algebra Appl. 28, e2338 (2021)
8. Martinsson, P.-G.: A fast randomized algorithm for computing a hierarchically semi-separable representation of a matrix. SIAM J. Matrix Anal. Appl. 32, 1251–1274 (2011)
9. Ghysels, P., Li, X., Rouet, F., Williams, S., Napov, A.: An efficient multicore implementation of a novel HSS-structured multifrontal solver using randomized sampling. SIAM J. Sci. Comput. 38, S358–S384 (2016)
10. Xia, J., Gu, M.: Robust approximate Cholesky factorization of rank-structured symmetric positive definite matrices. SIAM J. Matrix Anal. Appl. 31, 2899–2920 (2010)
11. Xiang, H., Zou, J.: Randomized algorithms for large-scale inverse problems with general Tikhonov regularizations. Inverse Problems 31, 085008 (2015)
12. Zhang, L., Wei, Y.: Randomized core reduction for discrete ill-posed problem. J. Comput. Appl. Math. 375, 112797 (2020)
13. Börm, S., Grasedyck, L., Hackbusch, W.: Introduction to hierarchical matrices with applications. Eng. Anal. Bound. Elem. 27, 405–422 (2003)
14. Chandrasekaran, S., Gu, M., Pals, T.: A fast ULV decomposition solver for hierarchically semi-separable representations. SIAM J. Matrix Anal. Appl. 28, 603–622 (2006)
15. Cai, D., Chow, E., Erlandson, L., Saad, Y., Xi, Y.: SMASH: Structured matrix approximation by separation and hierarchy. Numer. Linear Algebra Appl. 25, e2204 (2018)
16. Boukaram, W., Lucchesi, M., Turkıyayh, G., Maître, O.L., Knio, O., Keyes, D.: Hierarchical matrix approximations for space-fractional diffusion equations. Comput. Methods Appl. Mech. Engrg. 369, 113191 (2020)
17. Erlandson, L., Cai, D., Chow, E., Xi, Y.: Accelerating parallel hierarchical matrix-vector products via data driven sampling. In: IEEE International Parallel and Distributed Processing Symposium, pp. 749–758 (2020)
18. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
19. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika 1, 211–218 (1936)
20. Stewart, G.W.: The QLP approximation to the singular value decomposition. SIAM J. Sci. Comput. 20, 1336–1348 (1999)
21. Nakatsukasa, Y.: Fast and stable randomized low-rank matrix approximation, preprint, arXiv:2009.11392, (2020)
22. Huckaby, D.A., Chan, T.F.: On the convergence of Stewart’s QLP algorithm for approximating the SVD. Numer. Algorithms 32, 287–316 (2003)
23. Papadimitriou, C.H., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: a probabilistic analysis. J. Comput. Syst. Sci. 61, 217–235 (2000)
24. Liberty, E., Woolfe, F., Martinsson, P.-G., Rokhlin, V., Tygert, M.: Randomized algorithms for the low-rank approximation of matrices. Proc. Natl. Acad. Sci. 104, 20167–20172 (2007)
25. Halko, N., Martinsson, P.-G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. 53, 217–288 (2011)
26. Kaloorazi, M.F., de Lamare, R.C.: Subspace-orbit randomized decomposition for low-rank matrix approximations. IEEE Trans. Singnal Process. 66, 4409–4424 (2018)
27. Shabat, G., Shmueli, Y., Aizenbud, Y., Averbuch, A.: Randomized LU decomposition. Appl. Comput. Harmon. Anal. 44, 246–272 (2018)
28. Aizenbud, Y., Shabat, G., Averbuch, A.: Randomized LU decomposition using sparse projections. Comput. Math. Appl. 72, 2525–2534 (2016)
29. Gu, M.: Subspace iteration randomization and singular value problems. SIAM J. Sci. Comput. 37, A1139–A1173 (2015)
30. Aizenbud, Y., Averbuch, A.: Matrix decompositions using sub-Gaussian random matrices. Inf. Inference 8, 445–469 (2019)
31. Duersch, J.A., Gu, M.: Randomized QR with column pivoting. SIAM J. Sci. Comput. 39, C263–C291 (2017)
32. Xiao, J., Gu, M., Langou, J.: Fast parallel randomized QR with column pivoting algorithms for reliable low-rank matrix approximations. In: 2017 IEEE 24th International Conference on High Performance Computing, pp. 233–242 (2017)
33. Duersch, J.A., Gu, M.: Randomized projection for rank-revealing matrix factorizations and low-rank approximations. SIAM Rev. 62, 661–682 (2020)
34. Martinsson, P.-G., Ortí, G.Q., Heavner, N., van de Geijn, R.: Householder QR factorization with randomization for column pivoting (HQRRP). SIAM J. Sci. Comput. 39, C96–C115 (2017)
35. Demmel, J., Hoemmen, M., Mohiyuddin, M., Yelick, K.: Avoiding communication in computing Krylov subspaces, Technical report UCB/EECS-2007-123. EECS Department, University of California, Berkeley (2007)
36. Demmel, J., Hoemmen, M., Mohiyuddin, M., Yelick, K.: Avoiding communication in sparse matrix computations. In: Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, pp. 1–12 (2008)
37. Li, H., Yin, S.: Single-pass randomized algorithms for LU decomposition. Linear Algebra Appl. 595, 101–122 (2020)
38. Yu, W., Gu, Y., Li, J., Liu, S., Li Y.: Single-pass PCA of large high-dimensional data, preprint, arXiv:1704.07669, (2017)
39. Bjarkason, E.K.: Pass-efficient randomized algorithms for low-rank matrix approximation using any number of views. SIAM J. Sci. Comput. 41, A2355–A2383 (2019)
40. Tropp, J.A., Yurtsever, A., Udell, M., Cevher, V.: Practical sketching algorithms for low-rank matrix approximation. SIAM J. Matrix Anal. Appl. 38, 1454–1485 (2017)
41. Tropp, J.A., Yurtsever, A., Udell, M., Cevher, V.: Streaming low-rank matrix approximation with an application to scientific simulation. SIAM J. Sci. Comput. 41, A2430-2463 (2019)
42. Woodruff, D.P.: Sketching as a tool for numerical linear algebra. Found. Trends Theor. Comput. Sci. 10, 1–157 (2014)
43. Schweikardt, N.: One-pass algorithm. In: Encyclopedia of database systems. Springer, Boston (2009)
44. Wu, N., Xiang, H.: Randomized QLP decomposition. Linear Algebra Appl. 599, 18–35 (2020)
45. Foucart, S., Rauhut, H.: A mathematical introduction to compressive sensing. Springer, New York (2013)
46. Eldar, Y.C., Kutyniok, G.: Compressed Sensing: Theory and Applications. Cambridge University Press, Cambridge (2012)
47. Martinsson, P.-G.: Randomized methods for matrix computations, The Mathematics of Data, AMS, IAS/Park City Math. Ser. vol. 25, Amer. Math. Soc., Providence RI, 187–229, (2018)
48. Meier, M., Nakatsukasa, Y.: Fast randomized numerical rank estimation, preprint, arXiv:2105.07388 (2021)
49. Hansen, P.C.: Regularization tools: a MATLAB package for analysis and solution of discrete ill-posed problems (version 4.1 for MATLAB 7.3), Numer. Algorithms, 46 189–194, (2007)
50. Cai, D., Vassilevski, P.: Eigenvalue problems for exponential type kernels. Comput. Methods Appl. Math. 20, 61–78 (2020)
51. Atkinson, K., Chien, D., Hansen, O.: Multivariate quadrature of a singular integrand. Adv. Comput. Math. 47, 44 (2021)
52. Gazzola, S., Hansen, P.C., Nagy, J.G., Tools, I.R.: a MATLAB package of iterative regularization methods and large-scale test problems. Numer. Algorithms 81, 773–811 (2019)
53. Hron, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1991)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.