Stronger Approximate Singular Value Decomposition via the Block Lanczos and Power Methods

Cameron Musco and Christopher Musco
Massachusetts Institute of Technology, EECS
Cambridge, MA 02139, USA
Email: \{cnmusco,cpmusco\}@mit.edu

April 22, 2015

Abstract

We re-analyze Simultaneous Power Iteration and the Block Lanczos method, two classical iterative algorithms for the singular value decomposition (SVD). We are interested in convergence bounds that do not depend on properties of the input matrix (e.g. singular value gaps).

Simultaneous Iteration is known to give a low rank approximation within $(1 + \epsilon)$ of optimal for spectral norm error in $\tilde{O}(1/\epsilon)$ iterations. We strengthen this result, proving that it finds approximate principal components very close in quality to those given by an exact SVD. Our work bridges a divide between classical analysis, which can give similar bounds but depends critically on singular value gaps, and more recent work, which only focuses on low rank approximation.

Furthermore, we extend our bounds to the Block Lanczos method, which we show obtains the same approximation guarantees in just $\tilde{O}(1/\sqrt{\epsilon})$ iterations, giving the fastest known algorithm for spectral norm low rank approximation and principal component approximation. Despite their popularity, Krylov subspace methods like Block Lanczos previously seemed more difficult to analyze and did not come with rigorous gap-independent guarantees.

Finally, we give insight beyond the worst case, justifying why Simultaneous Power Iteration and Block Lanczos can run much faster in practice than predicted. We clarify how simple techniques can potentially accelerate both algorithms significantly.
1 Introduction

Any matrix $A \in \mathbb{R}^{n \times d}$ with rank $r$ can be written using a singular value decomposition (SVD) as $A = U \Sigma V^\top$. $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{d \times r}$ have orthonormal columns ($A$’s left and right singular vectors) and $\Sigma \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing $A$’s singular values: $\sigma_1 \geq \ldots \geq \sigma_r$.

Among countless applications, the SVD is often used in machine learning and dimensionality reduction to provide an optimal low rank approximation to $A$. Specifically, the Eckart-Young-Mirsky theorem guarantees that $A$’s partial SVD can be used to construct a rank $k$ approximation $A_k$ such that both $\|A - A_k\|_F$ and $\|A - A_k\|_2$ are as small as possible. $A_k$ is simply equal to $A$ projected onto the space spanned by its top $k$ singular vectors. That is, $A_k = U_k U_k^\top A$.

The SVD is also used for principal component analysis (PCA). $A$’s top singular vector $u_1$ provides a top principal component, which describes the direction of greatest variance within $A$. The $i^{th}$ singular vector $u_i$ provides the $i^{th}$ principal component, which is the direction of greatest variance orthogonal to all higher principal components. In other words,

$$u_i^\top A A^\top u_i = \sigma_i^2 = \max_{x: \|x\|_2 = 1, x \perp u_j, j < i} x^\top A A^\top x,$$

where $A A^\top$ is the covariance matrix of $A$ and $\sigma_i$ is its $i^{th}$ singular value.

Since classical methods for computing a partial SVD are expensive, substantial research has focused on fast, randomized approximation algorithms that seek nearly optimal low rank approximation and PCA [Sar06, MRT06, RST09, HMT11, CW13]. These algorithms have proven to be very fast in practice, especially for large data problems [HMST11, SKT14, IBM14, Tul14].

Ideally, an approximate partial SVD algorithm will provide good theoretical guarantees for both $k$ rank approximation and PCA. For a specified error $\epsilon$, we hope to return a rank $k$ matrix $Z$ with orthonormal columns $z_1, \ldots, z_k$ satisfying:

Frobenius Norm Error: \( \|A - ZZ^\top A\|_F^2 \leq (1 + \epsilon)\|A - A_k\|_F^2 \) \hspace{1cm} (1)

Spectral Norm Error: \( \|A - ZZ^\top A\|_2^2 \leq (1 + \epsilon)\|A - A_k\|_2^2 \) \hspace{1cm} (2)

Per Vector Error: \( \forall i, \|u_i^\top A A^\top u_i - z_i^\top A A^\top z_i\| \leq \epsilon \sigma_{k+1}^2 \) \hspace{1cm} (3)

Furthermore, we seek runtime bounds that do not depend on properties of $A$. While substantial literature exists on the convergence of iterative and approximate SVD algorithms, nearly all runtime guarantees depend on the gaps between $A$'s singular values and become useless when these gaps are small. This limitation is due to a focus on how quickly approximate singular vectors converge to the actual singular vectors of $A$. When two singular vectors have nearly identical values they are difficult to distinguish, so convergence inherently depends on singular value gaps.

Only recently has a shift in approximation goal allowed for algorithms that avoid this dependence, and thus run provably fast for any matrix. For low rank approximation and PCA, we are only concerned with finding a subspace that captures nearly as much information in $A$ as its top singular vectors – distinguishing between two close singular values is overkill.

1.1 Comparing Guarantees

The Frobenius norm guarantee (1) is well studied and there now exist algorithms achieving $(1 + \epsilon)$ error in $O(\text{nnz}(A))$ time, plus lower order terms depending on $\epsilon$, where $\text{nnz}(A)$ is the number of non-zero entries in $A$ [CW13]. However, as highlighted in prior work [RST09, HMT11, SKT14] Frobenius
norm error is often insufficient, especially in applications to data analysis and machine learning. When $A$ has a “heavy-tail” of singular values, as is common for noisy data, $\|A - A_k\|_2^2 = \sum_{i=k+1}^{r} \sigma_i^2$ can be huge, potentially larger than even $A$’s largest singular value. This renders (1) meaningless since $Z$ does not need to align with any large singular vectors to obtain good multiplicative error.

To address this shortcoming, a number of papers [Sar06, Woo14, SKT14] suggest targeting spectral norm error (2). When looking for a rank $k$ approximation, $A$’s top $k$ singular vectors are usually considered data and the remaining tail is noise. Intuitively, a spectral norm guarantee is large, so (1 + $\epsilon$) multiplicative Frobenius error is meaningless as well. For example, any $Z$ obtains $\|A - ZZ^T A\|_2^2 \leq (1.01)\|A - A_k\|_2^2$.

However, while intuitively stronger for many matrices, (1 + $\epsilon$) spectral error does not imply (1 + $\epsilon$) Frobenius error. Even more concerning, we can construct matrices for which neither low rank approximation guarantee implies any sort of accuracy in $Z$. Consider an $A$ with its top $k + 1$ squared singular values all equal to 10 followed by a tail of smaller singular values (e.g. 1000$k$ at 1). $\|A - A_k\|_2^2 = 10$ but in fact $\|A - ZZ^T A\|_2^2 = 10$ for any rank $k$ matrix $Z$, leaving the spectral norm bound useless. At the same time, $\|A - A_k\|_2^2$ is large, so (1 + $\epsilon$) multiplicative Frobenius error is meaningless as well. For example, any $Z$ obtains $\|A - ZZ^T A\|_2^2 \leq (1.01)\|A - A_k\|_2^2$.

We address this concern by introducing a per vector guarantee (3) which requires each approximate singular vector $z_1, \ldots, z_k$ to capture nearly as much variance as the corresponding true singular vector. The error bound is very strong in that it depends on $\epsilon \sigma_{k+1}^2$ instead of multiplicatively like $\|u_i^T A A^T u_i - z_i^T A A^T z_i\| \leq \epsilon \sigma_i^2$, which would be weaker for $A$’s larger singular vectors. While (3) is reminiscent of the bounds sought in classical numerical analysis [Saa80], we stress that it does not require each $z_i$ to converge to $u_i$ in the presence of small singular values gaps.

\section{1.2 Our Contributions}

In this paper we re-analyze the decades old Simultaneous Power Iteration and the Block Lanczos methods, combined with simple randomized initializations, for guarantees (1), (2), and (3).

**Algorithm 1 Simultaneous Iteration**

**input:** $A \in \mathbb{R}^{n \times d}$, error $\epsilon \in (0, 1)$, rank $k \leq n, d$

**output:** $Z \in \mathbb{R}^{n \times k}$.

1: $q = \Theta(\log d/\epsilon)$, $\Pi \sim \mathcal{N}(0, 1)^{d \times k}$
2: Set $K = (AA^T)^q \Pi \Pi$
3: Orthonormalize the columns of $K$ to obtain $Q \in \mathbb{R}^{n \times k}$.
4: Compute $M = Q^T AA^T Q \in \mathbb{R}^{k \times k}$.
5: Set $\tilde{U}_k$ to the top $k$ singular vectors of $M$.
6: return $Z = Q \tilde{U}_k$.

**Algorithm 2 Block Lanczos**

**input:** $A \in \mathbb{R}^{n \times d}$, error $\epsilon \in (0, 1)$, rank $k \leq n, d$

**output:** $Z \in \mathbb{R}^{n \times k}$.

1: $q = \Theta(\log d/\sqrt{\epsilon})$, $\Pi \sim \mathcal{N}(0, 1)^{d \times k}$
2: Set $K = [AAP, (AA^T)^q AP, \ldots, (AA^T)^q AP]$.
3: Orthonormalize the columns of $K$ to obtain $Q \in \mathbb{R}^{n \times qk}$.
4: Compute $M = Q^T AA^T Q \in \mathbb{R}^{qk \times qk}$.
5: Set $\tilde{U}_k$ to the top $k$ singular vectors of $M$.
6: return $Z = Q \tilde{U}_k$.

**Theorem 1** (Main Theorem). With high probability, Algorithms 1 and 2 find approximate singular vectors $Z = [z_1, \ldots, z_k]$ satisfying low rank approximation guarantees (1) and (2), and PCA guarantee (3). For error $\epsilon$, Algorithm 1 requires $q = O(\log d/\epsilon)$ iterations while Algorithm 2 requires $q = O(\log d/\sqrt{\epsilon})$ iterations. Excluding lower order terms, both algorithms run in time $O(nmz(A)kq)$.

Our proof appears in parts as Theorems 6 and 7 (runtime) and Theorems 10, 11, and 12 (accuracy).
While Simultaneous Iteration was known to achieve (2) [Woo14], surprisingly no bounds comparable to (1) and (3) are known. In fact, our analysis is the first to show that an approximation algorithm can achieve per vector guarantees like (3) in runtime independent of singular value gaps. Perhaps of greater interest is the fact that our analysis naturally applies to Krylov subspace methods like Block Lanczos. The theory for these methods is more limited, even though they have been proposed, discussed, and tested as a potential improvements over randomized power methods [RST09, HMST11, Hal12]. As highlighted in [SKT14],

“Despite decades of research on Lanczos methods, the theory for the randomized algorithms is more complete and provides strong guarantees of excellent accuracy, whether or not there exist any gaps between the singular values.”

Theorem 1 addresses this issue by giving the first gap independent bound for a Krylov subspace method. For guarantees (2) and (3), randomized Block Lanczos gives the fastest known algorithm, improving on the $\epsilon$ dependence of Simultaneous Iteration (substantially for small $\epsilon$).

Finally, in Section 5.3 we use our results to give a very simple alternative analysis that does depend on singular value gaps and can offer significantly faster convergence when $A$ has decaying singular values. It is possible to take further advantage of this result by running Algorithms 1 and 2 with a $\Pi$ that has $> k$ columns, a very simple modification for accelerating either method.

2 Background and Intuition

The goal of this section is to 1) provide background on algorithms for approximate singular value decomposition and 2) give intuition for Simultaneous Power Iteration and the Block Lanczos method, justifying why they can give strong gap-independent error guarantees.

2.1 Frobenius Norm Error

Progress on algorithms for Frobenius norm error low rank approximation (1) has been most considerable. Work in this direction dates back to the strong rank-revealing QR factorizations of Gu and Eisenstat [GE96]. They give deterministic algorithms that run in approximately $O(ndk)$ time, vs. $O(nd\min(n,d))$ for a full SVD, and roughly achieve constant factor Frobenius norm error.

Recently, randomization has been applied to achieve even faster algorithms with $(1+\epsilon)$ error. The paradigm is to compute a linear sketch of $A$ into very few dimensions using either a column sampling matrix or Johnson-Lindenstrauss random projection matrix $\Pi$. Typically $A\Pi$ has at most $\text{poly}(k/\epsilon)$ columns and can be used to quickly find $Z$ using a number of methods [Sar06, CEM+15].

$$A_{n \times d} \times \Pi_{d \times \text{poly}(k/\epsilon)} = AP_{n \times \text{poly}(k/\epsilon)}$$

This approach was developed and refined in several pioneering results, including [FKV04, DFK+04, DM06, DV06] for column sampling, [PTRV00, MRT06] for random projection, and definitive work by Sarlós [Sar06]. Recent work on sparse Johnson-Lindenstrauss type matrices [CW13, MM13, NN13] has brought the cost of Frobenius error low rank approximation down to

1 By the Abel-Ruffini Theorem, an exact SVD is incomputable even with exact arithmetic – see [TB97]. Accordingly, all SVD algorithm are inherently iterative. Nevertheless, classical methods such as the QR algorithm obtain superlinear convergence rates for the low rank approximation and PCA problems and in any reasonable computing environment, can be taken to run in $O(nd\min(n,d))$ time.
O(mnz(A) + n poly(k/ε)) time, where the first term is the number of non-zero entries in A and is considered to dominate since typically \( k \ll n, d \).

The sketch-and-solve method is very popular, largely because the computation of \( A\Pi \) is easily parallelized and, regardless, pass-efficient in a single processor setting. Furthermore, once a small compression of A is obtained, it can be manipulated in fast memory. This is not typically true of A itself, making it difficult to directly process the original matrix at all. Fast implementations of random projection methods are available through [ME11], [IBM14], and [SKT14].

### 2.2 Spectral Norm Error

Unfortunately, as discussed, Frobenius norm error is often insufficient when A has a heavy singular value tail. Furthermore, it seems an inherent limitation of sampling or random projection methods. The noise from A’s lower \( r - k \) singular values corrupts \( A\Pi \), making it impossible to extract a good partial SVD if the sum of these singular values (i.e. \( \| A - A_k \|_F^2 \)) is too large. In other words, any error inherently depends on the size of this tail.

This raises a natural question – is there any way to reduce this noise down to the scale of \( \sigma_{k+1} = \| A - A_k \|_2 \) and thus achieve a spectral norm bound like (2)? The answer is yes, and in fact this is exactly the intuition behind the famed power method.

Simultaneous Power Iteration (Algorithm 1), also known as subspace iteration, or orthogonal iteration, denoises A by working with the powered matrix \( A^q \) [Bau57, Rut70]. By the spectral theorem, \( A^q \) has exactly the same singular vectors as A, but its singular values are equal to the singular values of A raised to the \( q^\text{th} \) power². Powering spreads the values apart and accordingly, \( A^q \)’s lower singular values are relatively much smaller than its top singular values (see Figure 1a). This effectively reduces the noise in our problem – if we use a sketching method to find a good Z for approximating \( A^q \), even up to Frobenius error, Z will have to align very well with \( A^q \)’s large singular vectors.

Specifically, \( q = \tilde{O}(1/\epsilon) \) is sufficient to increase any singular value \( \geq (1 + \epsilon)\sigma_{k+1} \) to be significantly (i.e. poly(d) times) larger than any value \( \leq \sigma_{k+1} \). So \( \| A^q - A_k^q \|_F^2 \) is extremely small compared to the top singular values of \( A^q \). In order to achieve even rough multiplicative approximation to this error, Z must align extremely well with every singular vector with value \( \geq (1 + \epsilon)\sigma_{k+1} \). It thus provides an accurate basis for approximating A up to small spectral norm error.

Computing \( A^q \) directly is costly, so \( A^q\Pi \) is computed iteratively. We start with a random \( \Pi \) and repeatedly multiply by A on the left. Since even a rough Frobenius norm approximation for \( A^q \) suffices, \( \Pi \) is often chosen to have just k columns. Each iteration thus takes \( O(mnz(A)k) \) time. After \( A^q\Pi \) is computed, Z can simply be set to a basis for its column span. Note that per vector guarantees will require a more careful choice of this basis.

To the best of our knowledge, this approach to analyzing Simultaneous Iteration without dependence on singular value gaps began with [RST09]. The technique was popularized in [HMT11] and its analysis improved in [WC14] and [BDM14]. [Woo14] gives the first bound that directly achieves (2), showing that \( O(\log d/\epsilon) \) power iterations is sufficient for \((1 + \epsilon)\) error. All of these papers rely on an improved understanding of the benefits of starting with a randomized \( \Pi \), which has developed from work on the sketch-and-solve paradigm.

² For nonsymmetric matrices, we will work with \((AA^\top)^qA\).
A's singular values compared to those of $A^q$, rescaled to match on $\sigma_1$. Notice the significantly reduced tail after $\sigma_8$.

(b) A $\sqrt{q}$-degree Chebyshev polynomial, $T_{\sqrt{q}}(x)$, pushes low values nearly as close to zero as $x^q$ while spreading higher values less significantly.

Figure 1: Replacing $A$ with a matrix polynomial facilitates higher accuracy approximation.

### 2.3 Beating Simultaneous Iteration with Lanczos

Numerous papers hint at the possibility of beating Simultaneous Iteration with the Block Lanczos method [CD74, GU77, GLO81], a well studied variant of Lanczos iteration [Lan50], which is the canonical Krylov subspace method for large singular value problems. In particular, [RST09], [HMST11] and [Hal12] suggest and experimentally confirm the potential of randomized Block Lanczos (Algorithm 2) for beating Simultaneous Iteration for low rank approximation. [ME11] also notes the difficulty of beating state-of-the-art Lanczos implementations [Lar01, Lar05] with Simultaneous Iteration.

The intuition behind Block Lanczos matches that of many accelerated iterative methods. Simply put, there are better polynomials than $A^q$ for denoising tail singular values. In particular, we can use lower degree polynomials, allowing us to compute fewer powers of $A$ and thus leading to an algorithm with fewer iterations. For example, an appropriately shifted $\sqrt{q}$ degree Chebyshev polynomial can push the tail of $A$ nearly as close to zero as $A^q$, even if the long run growth of the polynomial is much lower (see Figure 1b). Specifically, $q = \tilde{O}(1/\sqrt{\epsilon})$ will increase any singular value $\geq (1 + \epsilon)\sigma_{k+1}$ to be significantly larger than any singular value below $\sigma_{k+1}$ – enough to achieve near optimal spectral norm error using a sketching method.

Block Lanczos takes advantage of such polynomials by working with the block Krylov subspace,

$$K = [\Pi \quad A\Pi \quad A^2\Pi \quad A^3\Pi \ldots \quad A^{\sqrt{q}}\Pi],$$

from which we can construct $p_{\sqrt{q}}(A)\Pi$ for any polynomial $p_{\sqrt{q}}(\cdot)$ of degree $\sqrt{q}$. Since an effective polynomial for denoising $A$ must be scaled and shifted based on the value of $\sigma_{k+1}$, we cannot easily compute $p_{\sqrt{q}}(A)\Pi$ directly. Instead, we argue that the best $k$ rank approximation to $A$ lying in the span of $K$ at least matches the approximation achieved by projecting onto the span of $p_{\sqrt{q}}(A)\Pi$. Finding this best approximation will therefore give a nearly optimal low rank approximation to $A$.

Unfortunately, there’s a catch. Perhaps surprisingly, it is not clear how to efficiently compute the best spectral norm error low rank approximation to $A$ lying in a specific subspace (e.g. $K$’s span)
aligns very well with the singular vectors corresponding to these large singular values. The total loss compared to optimal is at worst $\epsilon$ for the true top $k$ Frobenius norm error and the per vector guarantee (3.1 Singular Value Decomposition and Low Rank Approximation). polynomial approximation, and randomized low rank approximation.

Before proceeding to the full technical analysis, we overview required results from linear algebra, preliminary results from linear algebra, polynomial approximation, and randomized low rank approximation.

### 2.4 Per Vector Error

Achieving the per vector guarantee of (3) requires a more nuanced understanding of how Simultaneous Iteration and Block Lanczos denoise the spectrum of $A$. The analysis for spectral norm low rank approximation relies on the fact that $A^3$ and $p, \sigma(A)$ blow up any singular value $\sigma_{k+1}$ to much larger than any singular value $\leq \sigma_{k+1}$. This ensures that the $Z$ outputted by both algorithms aligns very well with the singular vectors corresponding to these large singular values.

If $\sigma_k \geq (1 + \epsilon)\sigma_{k+1}$, then $Z$ aligns well with all top $k$ singular vectors of $A$ and we get good Frobenius norm error and the per vector guarantee (3). Unfortunately, when there is a small gap between $\sigma_k$ and $\sigma_{k+1}$, $Z$ could miss intermediate singular vectors whose values lie between $\sigma_{k+1}$ and $(1 + \epsilon)\sigma_{k+1}$. This is the case where gap dependent guarantees of classical analysis break down.

Nevertheless, we can argue that $A^3$ or, for Block Lanczos, another $\sqrt{t}$-degree polynomial in our Krylov subspace, significantly separates singular values $> \sigma_{k+1}$ from those $< (1 - \epsilon)\sigma_{k+1}$. Thus, each column of $Z$ will align with $A$ at least nearly as well as $u_{k+1}$. There may be a large subspace of singular vectors with values in the intermediate range $[(1 - \epsilon)\sigma_{k+1}, (1 + \epsilon)\sigma_{k+1}]$. Our polynomial cannot spread apart these values significantly, so we cannot characterize how $Z$ aligns with this space. However, as long as it avoids singular values below this range, we can guarantee (3).

For Frobenius norm low rank approximation, we can argue that the degree to which $Z$ falls outside of the space spanned by the top $k$ singular vectors depends on the number of intermediate singular values between $\sigma_{k+1}$ and $(1 - \epsilon)\sigma_{k+1}$. These are the singular values that may be ‘swapped in’ for the true top $k$ singular values. Since their weight counts towards $A$’s tail, we can show that the total loss compared to optimal is at worst $\epsilon \|A - A_k\|_F^2$.

### 3 Preliminaries

Before proceeding to the full technical analysis, we overview required results from linear algebra, polynomial approximation, and randomized low rank approximation.

#### 3.1 Singular Value Decomposition and Low Rank Approximation

As mentioned, the singular value decomposition can be used to write any $A \in \mathbb{R}^{n \times d}$ as $A = U \Sigma V^T$, where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{d \times r}$ have orthonormal columns and $\Sigma \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing the singular values of $A$: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$. The singular value decomposition exists for all matrices. The pseudoinverse of $A$ is given by $A^+ = V \Sigma^{-1} U^T$. Additionally, for any polynomial $p(x)$, we define $p(A) = U p(\Sigma) V^T$. Note that, since singular values are always take to be non-negative, $p(A)$’s singular values are given by $|p(\Sigma)|$.

Let $\Sigma_k$ be $\Sigma$ with all but its largest $k$ singular values zeroed out. Let $U_k$ and $V_k$ be $U$ and $V$ with all but their first $k$ columns zeroed out. For any $k$, $A_k = U \Sigma_k V^T = U_k \Sigma_k V_k^T$ is the closest rank $k$ approximation to $A$ for any unitarily invariant norm, including the Frobenius norm and spectral norm [Mir60]. The squared Frobenius norm is given by $\|A\|_F^2 = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T) = \sum_{i,j} A_{i,j}^2 = \text{tr}(AA^T)$.
The spectral norm is given by \( \|A\|_2 = \sigma_1 \).

\[
\|A - A_k\|_F = \min_{B: \text{rank}(B) = k} \|A - B\|_F \quad \text{and} \quad \|A - A_k\|_2 = \min_{B: \text{rank}(B) = k} \|A - B\|_2.
\]

We often work with the remainder matrix \( A - A_k \) and label it \( A_{r,k} \). Its singular value decomposition is given by \( A_{r,k} = U_{r,k} \Sigma_{r,k} V_{r,k}^T \) where \( U_{r,k}, \Sigma_{r,k}, \) and \( V_{r,k}^T \) have their first \( k \) columns zeroed.

While the SVD gives a globally optimal rank \( k \) approximation for \( A \), both Simultaneous Iteration and Block Lanczos will return the best \( k \) rank approximation falling within some fixed subspace spanned by a basis \( Q \) (with rank \( \geq k \)). For the Frobenius norm, this simply requires projecting \( A \) to \( Q \) and taking the best rank \( k \) approximation of the resulting matrix using an SVD.

**Lemma 2** (Lemma 4.1 of [Woo14]). Given \( A \in \mathbb{R}^{n \times d} \) and \( Q \in \mathbb{R}^{m \times n} \) with orthonormal columns,

\[
\|A - (QQ^T A)_k\|_F = \|A - Q (Q^T A)_k\|_F = \min_{C: \text{rank}(C) = k} \|A - QC\|_F.
\]

This low rank approximation can be obtained using an SVD (equivalently, eigendecomposition) of the \( m \times m \) matrix \( M = Q^T (AA^T) Q \). Specifically, letting \( M = U \Sigma^2 U^T \), then:

\[
(Q \bar{U}_k) (Q \bar{U}_k)^T A = Q (Q^T A)_k.
\]

**Proof.** The first fact is well known in the literature and is given as Lemma 4.1 of [Woo14]. The second fact just follows from noting that, if the SVD of \( Q^T A \) is given by \( Q^T A = \bar{U} \Sigma \bar{V}^T \) then \( M = Q^T (AA^T) Q = U \Sigma^2 U^T \). So \( Q (Q^T A)_k = Q \bar{U}_k \Sigma_k \bar{V}_k^T = Q (\bar{U}_k \bar{U}_k^T) U \Sigma V^T = Q \bar{U}_k \bar{U}_k^T Q^T A \).

Note that \( Q \bar{U}_k \) has orthonormal columns since \( \bar{U}_k^T \bar{U}_k = I_k \).

In general, this rank \( k \) approximation does not give the best spectral norm approximation to \( A \) falling within \( Q \) [BDM14]. A closed form solution for the best spectral norm approximation can be obtained using the results of [SR10], which are related to Parrott’s theorem. However we do not know an efficient way to compute this solution without essentially performing an SVD of \( A \). It is simple to show at least that, given a rank \( k \) basis, the optimal spectral norm approximation for \( A \) spanned by that basis is obtained by projecting \( A \) to the basis:

**Lemma 3** (Lemma 4.14 of [Woo14]). For \( A \in \mathbb{R}^{n \times d} \) and \( Q \in \mathbb{R}^{n \times k} \) with orthonormal columns,

\[
\|A - Q Q^T A\|_2 = \min_{C} \|A - QC\|_2.
\]

### 3.2 Other Linear Algebra Tools

Throughout this paper we will use \( \text{span}(M) \) to denote the column span of the matrix \( M \) and say that a matrix \( Q \) is an orthonormal basis for the column span of \( M \) if \( Q \) has orthonormal columns and \( QQ^T M = M \). That is, projecting the columns of \( M \) to \( Q \) fully recovers those columns. \( QQ^T \) is the orthogonal projection matrix onto the span of \( Q \). \( (QQ^T)(QQ^T) = QIQ^T = QQ^T \).

If two matrices \( M \) and \( N \) have the same dimensions and \( MN^T = 0 \) then \( \|M + N\|_F^2 = \|M\|_F^2 + \|N\|_F^2 \). This matrix Pythagorean theorem follows from the fact that \( \|M + N\|_F^2 = \text{tr}((M + N)(M + N)^T) \). As an example, note that for any orthogonal projection \( QQ^T \), \( A^T(I - QQ^T)QQ^T A = 0 \) so:

\[
\|A - QQ^T A\|_F^2 = \|A\|_F^2 - \|QQ^T A\|_F^2.
\]

This implies for example, that since \( A_k = U_k U_k^T A \) minimizes \( \|A - A_k\|_F^2 \) over all rank \( k \) matrices, \( U_k U_k^T A \) maximizes \( \|U_k U_k^T A\|_F^2 \) over all rank \( k \) orthogonal projections.
3.3 Randomized Low Rank Approximation

As mentioned, our proofs build on well known sketch-based algorithms for low rank approximation with Frobenius norm error. A short proof of the following Lemma is in Appendix A:

**Lemma 4 (Frobenius Norm Low Rank Approximation).** Take any \( A \in \mathbb{R}^{n \times d} \) and \( \Pi \in \mathbb{R}^{d \times k} \) where the entries of \( \Pi \) are independent Gaussians drawn from \( \mathcal{N}(0, 1) \). If we let \( Z \) be an orthonormal basis for \( \text{span}(A\Pi) \), then with probability at least \( 99/100 \), for some fixed constant \( c \),

\[
\|A - ZZ^\top A\|_F^2 \leq c \cdot dk \|A - A_k\|_F^2.
\]

3.4 Chebyshev Polynomials

As outlined in Section 2.3, our proof also requires polynomials to more effectively denoise the tail of \( A \). As is standard for Krylov subspace methods, we use a variation on the Chebyshev polynomials. The proof of the following Lemma is relegated to Appendix A.

**Lemma 5 (Chebyshev Minimizing Polynomial).** Given a specified value \( \alpha > 0 \), gap \( \gamma \in (0, 1] \), and degree \( q \geq 1 \), there exists a degree \( q \) polynomial \( p(x) \) such that:

1. \( p((1 + \gamma)\alpha) = (1 + \gamma)\alpha \)
2. \( p(x) \geq x \) for all \( x \geq (1 + \gamma)\alpha \)
3. \( p(x) \leq \frac{\alpha}{2^{\lfloor q/2 \rfloor + 1}} \) for all \( x \in [0, \alpha] \)

4 Implementation and Runtimes

In this section we briefly discuss runtime and implementation considerations for Algorithms 1 and 2, our randomized variants of Simultaneous Power Iteration and the Block Lanczos methods.

4.1 Simultaneous Iteration

Algorithm 1 can be modified in a number of ways. \( \Pi \) can be replaced by a random sign matrix, or any matrix achieving the guarantee of Lemma 4. \( \Pi \) may also be chosen with \( p > k \) columns. We will discuss in detail how this approach can give improved accuracy in Section 5.3.

In our implementation we set \( Z = Q\overline{U}_k \). This ensures that, for all \( l \leq k \), \( Z_l \) gives the best rank \( l \) Frobenius norm approximation to \( A \) within the span of \( K \) (See Lemma 2). This is necessary for achieving per vector guarantees for approximate PCA. However, if we are only interested in computing a near optimal low rank approximation, we can simply set \( Z = Q \). Projecting \( A \) to \( Q\overline{U}_k \) is equivalent to projecting to \( Q \) as these two matrices have the same spans.

**Theorem 6 (Simultaneous Iteration Runtime).** Algorithm 1 runs in time

\[
O(\text{nnz}(A)k \log d/\epsilon + nk^2).
\]

**Proof.** Computing \( K \) requires first multiplying \( A \) by \( \Pi \), which takes \( O(\text{nnz}(A)k) \) time. Computing \( (AA^\top)^i \Pi \) given \( (AA^\top)^{i-1} \Pi \) then takes \( O(\text{nnz}(A)k) \) time to first multiply our \( (n \times k) \) matrix by \( A^\top \) and then by \( A \). This gives a total runtime of \( O(\text{nnz}(A)kq) \) for computing \( K \).
Finding $Q$ via Gram-Schmidt orthogonalization or Householder reflections takes $O(nk^2)$ time. Computing $M$ by multiplying from left to right requires $O(nnz(A)k + nk^2)$ time. $M$’s SVD then requires $O(k^3)$ time using classical techniques (e.g. the QR algorithm). Finally, multiplying $\bar{U}_k$ by $Q$ takes time $O(nk^2)$. Since we set $q = \Theta(\log d/\epsilon)$, our total runtime is $O\left(nnz(A)\frac{k\log d}{\epsilon} + nk^2\right)$.

### 4.2 Block Lanczos

As with Simultaneous Iteration, we can replace $\Pi$ with any matrix achieving the guarantee of Lemma 4 and can use $p > k$ columns to improve accuracy (see Section 5.3). Additionally, $Q$ can be computed in a number of ways. In the traditional Block Lanczos algorithm, one starts by computing an orthonormal basis for $A\Pi$, the first block in the Krylov subspace. Bases for subsequent blocks are computed from previous blocks using a three term recurrence that ensures $Q^\top AA^\top Q$ is block tridiagonal, with $k \times k$ sized blocks [GU77]. This technique can be useful if $qk$ is large, since it is faster to compute the top singular vectors of a block tridiagonal matrix. However, computing $Q$ using a recurrence can introduce a number of stability issues, and additional steps may be required to ensure that the matrix remains orthogonal [GVL96].

An alternative is to compute $K$ explicitly and then compute $Q$ using a QR decomposition. This method is used in [RST09] and [HMST11]. It does not guarantee that $Q^\top AA^\top Q$ is block tridiagonal, but helps avoid a number of stability issues. Furthermore, if $qk$ is small, taking the SVD of $Q^\top AA^\top Q$ will still be fast and typically dominated by the cost of computing $K$.

**Theorem 7 (Block Lanczos Runtime).** Algorithm 2 runs in time

$$O\left(nnz(A)\frac{k\log d}{\sqrt{\epsilon}} + n\frac{k^2 \log^2 d}{\epsilon} + \frac{k^3 \log^3 d}{\epsilon^{3/2}}\right).$$

**Proof.** Computing $K$ requires $O(nnz(A)kq)$ time just like computing $K$ for Simultaneous Iteration (see Theorem 6). The remaining steps are analogous to those in Simultaneous Iteration except somewhat more costly as we work an $k \cdot q$ dimensional rather than $k$ dimensional subspace. Finding $Q$ takes $O(n(kq)^2)$ time. Computing $M$ take $O(nnz(A)(kq) + n(kq)^2)$ time and its SVD then requires $O((kq)^3)$ time. Finally, multiplying $\bar{U}_k$ by $Q$ takes time $O(nk(kq))$. Plugging in $q = \Theta(\log d/\sqrt{\epsilon})$ gives the claimed runtime. □

### 5 Error Bounds

We now prove that both Algorithms 1 and 2 return a basis $Z$ that gives relative error Frobenius (1) and spectral norm (2) low rank approximation error as well as the per vector guarantees (3).

#### 5.1 Main Approximation Lemma

We first prove a general approximation lemma, which gives three guarantees formalizing the intuition given in Section 2. All other proofs follow nearly immediately from this lemma.

For simplicity we assume throughout that $k \leq r \leq n, d$. However, if $k$ is greater than $r = \text{rank}(A)$ it can be seen that both algorithms still return a basis satisfying the proven guarantees. We start with a definition:
Definition 8. For a given matrix $\mathbf{Z} \in \mathbb{R}^{n \times k}$ with orthonormal columns, letting $\mathbf{Z}_l \in \mathbb{R}^{n \times l}$ be the first $l$ columns of $\mathbf{Z}$, we define the error function:

$$
\mathcal{E}(\mathbf{Z}_l, \mathbf{A}) = \|\mathbf{A}_l\|_F^2 - \|\mathbf{Z}_l \mathbf{Z}_l^\top \mathbf{A}\|_F^2 \\
= \|\mathbf{A} - \mathbf{Z}_l \mathbf{Z}_l^\top \mathbf{A}\|_F^2 - \|\mathbf{A} - \mathbf{A}_l\|_F^2.
$$

Recall that $\mathbf{A}_l$ is the best rank $l$ approximation to $\mathbf{A}$. This error function measures how well $\mathbf{Z}_l \mathbf{Z}_l^\top \mathbf{A}$ approximates $\mathbf{A}$ in comparison to the optimal.

Lemma 9 (Main Approximation Lemma). Let $m$ be the number of singular values $\sigma_i$ of $\mathbf{A}$ with $\sigma_i \geq (1+\epsilon/2)\sigma_{k+1}$. Let $w$ be the number of singular values with $1/(1+\epsilon/2)\sigma_k \leq \sigma_i < \sigma_k$. With probability $99/100$ Algorithms 1 and 2 return $\mathbf{Z}$ satisfying:

1. $\forall l \leq m$, $\mathcal{E}(\mathbf{Z}_l, \mathbf{A}) \leq (\epsilon/2) \cdot \sigma_{k+1}^2$.
2. $\forall l \leq k$, $\mathcal{E}(\mathbf{Z}_l, \mathbf{A}) \leq \mathcal{E}(\mathbf{Z}_{l-1}, \mathbf{A}) + 3\epsilon \cdot \sigma_{k+1}^2$.
3. $\forall l \leq k$, $\mathcal{E}(\mathbf{Z}_l, \mathbf{A}) \leq (w + 1) \cdot 3\epsilon \cdot \sigma_{k+1}^2$.

Property 1 captures the intuition given in Section 2.2. Both algorithms return $\mathbf{Z}$ with $\mathbf{Z}_l$ equal to the best Frobenius norm low rank approximation in $\text{span}(\mathbf{K})$. Since $\sigma_1 \geq \ldots \geq \sigma_m \geq (1+\epsilon/2)\sigma_{k+1}$ and our polynomials separate any values above this threshold from anything below $\sigma_{k+1}$, $\mathbf{Z}$ must align very well with $\mathbf{A}$’s top $m$ singular vectors. Thus $\mathcal{E}(\mathbf{Z}_l, \mathbf{A})$ is very small for all $l \leq m$.

Property 2 captures the intuition of Section 2.4 – outside of the largest $m$ singular values, $\mathbf{Z}$ still performs well. We may fail to distinguish between vectors with values between $\frac{1}{1+\epsilon/2} \sigma_k$ and $(1+\epsilon/2)\sigma_{k+1}$. However, aligning with the smaller vectors in this range rather than the larger vectors can incur a cost of at most $O(\epsilon)\sigma_{k+1}^2$. Since every column of $\mathbf{Z}$ outside of the first $m$ may incur such a cost, there is a linear accumulation as characterized by Property 2.

Finally, Property 3 captures the intuition that the total error in $\mathbf{Z}$ is bounded by the number of singular values falling in the range $\frac{1}{1+\epsilon/2} \sigma_k \leq \sigma_i < \sigma_k$. This is the total number of singular vectors that aren’t necessarily separated from and can thus be ‘swapped in’ for any of the $(k - m)$ true top vectors with singular value $<(1+\epsilon/2)\sigma_{k+1}$. Property 3 is critical in achieving near optimal Frobenius norm low rank approximation.

Proof. Proof of Property 1

Assume $m \geq 1$. If $m = 0$ then Property 1 trivially holds. We will prove the statement for Algorithm 2, since this is the more complex case, and then explain how the proof extends to Algorithm 1.

Let $p_1$ be the polynomial from Lemma 5 with $\alpha = \sigma_{k+1}$, $\gamma = \epsilon/2$, and $q \geq c \log(d/\epsilon)/\sqrt{\epsilon}$ for some fixed constant $c$. We can assume $1/\epsilon = O(\text{poly } d)$ and thus $q = O(\log d/\sqrt{\epsilon})$. Otherwise our Krylov subspace would have as many columns as $\mathbf{A}$ and we may as well use a classical algorithm to compute $\mathbf{A}$’s partial SVD directly. Let $\mathbf{Y}_1 \in \mathbb{R}^{n \times k}$ be an orthonormal basis for the span of $p_1(\mathbf{A})\mathbf{P}$. Recall that we defined $p_1(\mathbf{A}) = \mathbf{U} p_1(\Sigma) \mathbf{V}^\top$. As long as we choose $q$ to be odd, by the recursive definition of the Chebyshev polynomials, $p_1(\mathbf{A})$ only contains odd powers of $\mathbf{A}$. Any odd power $i$ can be evaluated as $(\mathbf{A} \mathbf{A}^\top)^{(i-1)/2} \mathbf{A}$. Accordingly, $p_1(\mathbf{A})\mathbf{P}$ and thus $\mathbf{Y}_1$ have columns falling within the span of the Krylov subspace from Algorithm 2 (and hence its column basis $\mathbf{Q}$).
By Lemma 4 we have with probability 99/100:

\[ \| p_1(A) - Y_1Y_1^T p_1(A) \|_F^2 \leq cdk \| p_1(A) - p_1(A)_k \|_F^2. \tag{4} \]

Furthermore, one possible rank \( k \) approximation of \( p_1(A) \) is \( p_1(A)_k \). By the optimality of \( p_1(A)_k \),

\[ \| p_1(A) - p_1(A)_k \|_F^2 \leq \| p_1(A) - p_1(A) \|_F^2 \leq \sum_{i=k+1}^{d} p_1(\sigma_i)^2 \leq d \cdot \left( \frac{\sigma_{k+1}^2}{2^{2q\sqrt{\epsilon}/2-2}} \right) \leq O \left( \frac{\epsilon}{2d^2\sigma_{k+1}^2} \right). \]

The last inequalities follow from setting \( q = \Theta(\log(d/\epsilon)/\sqrt{\epsilon}) \) and from the fact that \( \sigma_i \leq \sigma_{k+1} = \alpha \) for all \( i \geq k + 1 \) and thus by property 3 of Lemma 5, \( p_1(\sigma_i) \leq \frac{\sigma_{k+1}}{2\sqrt{\epsilon}/2-1} \). Noting that \( k \leq d \), we can plug this bound into (4) to get

\[ \| p_1(A) - Y_1Y_1^T p_1(A) \|_F^2 \leq \frac{\epsilon}{2}\sigma_{k+1}^2. \tag{5} \]

Applying the Pythagorean theorem and the invariance of the Frobenius norm under rotation gives

\[ \| p_1(\Sigma) \|_F^2 - \frac{\epsilon\sigma_{k+1}^2}{2} \leq \| Y_1Y_1^T U p_1(\Sigma) \|_F^2. \]

\( Y_1 \) falls within \( A \)’s column span, and therefore \( U \)’s column span. So we can write \( Y_1 = UC \) for some \( C \in \mathbb{R}^{r \times k} \). Since \( Y_1 \) and \( U \) have orthonormal columns, so must \( C \). We can now write

\[ \| p_1(\Sigma) \|_F^2 - \frac{\epsilon\sigma_{k+1}^2}{2} \leq \| UCC^T U^T U p_1(\Sigma) \|_F^2 = \| UCC^T p_1(\Sigma) \|_F^2 = \| C^T p_1(\Sigma) \|_F^2. \]

Letting \( c_i \) be the \( i \)th row of \( C \), expanding out these norms gives

\[ \sum_{i=1}^{r} p_1(\sigma_i)^2 - \frac{\epsilon\sigma_{k+1}^2}{2} \leq \sum_{i=1}^{r} ||c_i||^2 p_1(\sigma_i)^2. \tag{6} \]

Since \( C \)’s columns are orthonormal, its rows all have norms upper bounded by 1. So \( ||c_i||^2 p_1(\sigma_i)^2 \leq p_1(\sigma_i)^2 \) for all \( i \). So for all \( l \leq r \), (6) gives us

\[ \sum_{i=1}^{l} (1 - ||c_i||^2)p_1(\sigma_i)^2 \leq \sum_{i=1}^{r} (1 - ||c_i||^2)p_1(\sigma_i)^2 \leq \frac{\epsilon\sigma_{k+1}^2}{2}. \]

Recall that \( m \) is the number of singular values with \( \sigma_i \geq (1 + \epsilon/2)\sigma_{k+1} \). By Property 2 of Lemma 5, for all \( i \leq m \) we have \( \sigma_i \leq p_1(\sigma_i) \). This gives, for all \( l \leq m \):

\[ \sum_{i=1}^{l} (1 - ||c_i||^2)^2 \leq \frac{\epsilon\sigma_{k+1}^2}{2} \quad \text{and so} \]

\[ \sum_{i=1}^{l} \sigma_i^2 - \frac{\epsilon\sigma_{k+1}^2}{2} \leq \sum_{i=1}^{r} ||c_i||^2 \sigma_i^2. \]
Converting these sums back to norms yields $\|\Sigma_l\|_F^2 - \frac{\epsilon \sigma_{k+1}^2}{2} \leq \|C^\top \Sigma_l\|_F^2$ and therefore $\|A_l\|_F^2 - \frac{\epsilon \sigma_{k+1}^2}{2} \leq \|Y_1 Y_1^\top A_l\|_F^2$ and

$$\|A_l\|_F^2 - \|Y_1 Y_1^\top A_l\|_F^2 \leq \frac{\epsilon \sigma_{k+1}^2}{2}. \quad (7)$$

Now $Y_1 Y_1^\top A_l$ is a rank $l$ approximation to $A$ falling within the column span of $Y$ and hence within the column span of $Q$. By Lemma 2, the best rank $l$ Frobenius approximation to $A$ within $Q$ is given by $QU_l(QU_l)^\top A$. So we have

$$\|A_l\|_F^2 - \|QU_l(QU_l)^\top A\|_F^2 = \mathcal{E}(Z_l, A) \leq \frac{\epsilon \sigma_{k+1}^2}{2},$$

giving Property 1.

For Algorithm 1, we instead choose $p_1(x) = (1+\epsilon/2)\sigma_{k+1} \cdot \left(\frac{x}{(1+\epsilon/2)\sigma_{k+1}}\right)^{2q+1}$. For $q = \Theta(\log d/\epsilon)$, this polynomial satisfies the necessary properties: for all $i \geq k+1$, $p_1(\sigma_i) \leq O\left(\frac{\epsilon}{\sigma_{k+1}}\sigma_{k+1}^2\right)$ and for all $i \leq m$, $\sigma_i \leq p_1(\sigma_i)$. Further, up to a rescaling, $p_1(A)\Pi = K$ so $Y_1$ spans the same space as $K$. Therefore since Algorithm 1 returns $Z$ with $Z_l$ equal to the best rank $l$ Frobenius norm approximation to $A$ within the span of $K$, for all $l$ we have:

$$\|QU_l(QU_l)^\top A\|_F^2 \geq \|Y_1 Y_1^\top A_l\|_F^2 \geq \|A_l\|_F^2 - \frac{\epsilon \sigma_{k+1}^2}{2},$$

giving the proof.

Proof of Property 2

Property 1 and the fact that $\mathcal{E}(Z_l, A)$ is always positive immediately gives Property 2 for $l \leq m$. So we need to show that it holds for $m < l \leq k$. Note that if $w$, the number of singular values with $\frac{1}{1+\epsilon/2}\sigma_k \leq \sigma_i < \sigma_k$ is equal to 0, then $\sigma_{k+1} < \frac{1}{1+\epsilon/2}\sigma_k$, so $m = k$ and we are done. So we assume $w \geq 1$ henceforth. Again, we first prove the statement for Algorithm 2 and then explain how the proof extends to the simpler case of Algorithm 1.

Intuitively, Property 1 follows from the guarantee that there is a rank $m$ subspace of $\text{span}(K)$ that aligns with $A$ nearly as well as the space spanned by $A$’s top $m$ singular vectors. To prove Property 2 we must show that there is also some rank $k$ subspace in $\text{span}(K)$ whose components all align nearly as well with $A$ as $u_k$, the $k^{th}$ singular vector of $A$. The existence of such a subspace ensures that $Z$ performs well, even on singular vectors in the intermediate range $[\sigma_k, (1+\epsilon/2)\sigma_{k+1}]$. Let $p_2$ be the polynomial from Lemma 5 with $\alpha = \frac{1}{1+\epsilon/2}\sigma_k$, $\gamma = \epsilon/2$, and $q \geq c \log(d/\epsilon)/\sqrt{\epsilon}$ for some fixed constant $c$. Let $Y_2 \in \mathbb{R}^{n \times k}$ be an orthonormal basis for the span of $p_2(A)\Pi$. Again, as long as we choose $q$ to be odd, $p_2(A)$ only contains odd powers of $A$ and so $Y_2$ falls within the span of the Krylov subspace from Algorithm 2. We wish to show that for every unit vector $x$ in the column span of $Y_2$, $\|x^\top A\|_2 \geq \frac{1}{1+\epsilon/2}\sigma_k$.

Let $A_{\text{inner}} = A_{r\setminus k} - A_{r\setminus (k+w)}$, $A_{\text{inner}} = U\Sigma_{\text{inner}}V^\top$ where $\Sigma_{\text{inner}}$ contains only the singular values $\sigma_{k+1}, \ldots, \sigma_{k+w}$. These are the $w$ intermediate singular values of $A$ falling in the range $\left[\frac{1}{1+\epsilon/2}\sigma_k, \sigma_k\right)$. Let $A_{\text{outer}} = A - A_{\text{inner}} = U\Sigma_{\text{outer}}V^\top$. $\Sigma_{\text{outer}}$ contains all large singular values of $A$ with $\sigma_i \geq \sigma_k$ and all small singular values with $\sigma_i < \frac{1}{1+\epsilon/2}\sigma_k$. 

12
Let $Y_{inner} \in \mathbb{R}^{n \times \min\{k, w\}}$ be an orthonormal basis for the columns of $p_2(A_{inner})\Pi$. Similarly let $Y_{outer} \in \mathbb{R}^{n \times k}$ be an orthonormal basis for the columns of $p_2(A_{outer})\Pi$.

Every column of $Y_{inner}$ falls in the column span of $A_{inner}$ and hence the column span of $U_{inner} \in \mathbb{R}^{n \times w}$, which contains only the singular vectors of $A$ corresponding to the inner singular values. Similarly, the columns of $Y_{outer}$ fall within the span of $U_{outer} \in \mathbb{R}^{n \times r-w}$, which contains the remaining left singular vectors of $A$. So the columns of $Y_{inner}$ are orthogonal to those of $Y_{outer}$ and $[Y_{inner}, Y_{outer}]$ forms an orthogonal basis. For any unit vector $x \in \text{span}(p_2(A)\Pi) = \text{span}(Y_2)$ we can write $x = x_{inner} + x_{outer}$ where $x_{inner}$ and $x_{outer}$ are orthogonal vectors in the spans of $Y_{inner}$ and $Y_{outer}$ respectively. We have:

$$\|x^\top A\|_2^2 = \|x_{inner}^\top A\|_2^2 + \|x_{outer}^\top A\|_2^2. \quad (8)$$

We will lower bound $\|x^\top A\|_2^2$ by considering each contribution separately. First, any unit vector $x' \in \mathbb{R}^n$ in the column span of $Y_{inner}$ can be written as $x' = U_{inner}z$ where $z \in \mathbb{R}^w$ is a unit vector.

$$\|x'^\top A\|_2^2 = z^\top U_{inner}^\top A A^\top U_{inner}z = z^\top \Sigma_{inner}^2 z \geq \left(\frac{1}{1+\epsilon/2}\sigma_k\right)^2 \geq (1-\epsilon)\sigma_k^2. \quad (9)$$

Note that we’re abusing notation slightly, using $\Sigma_{inner} \in \mathbb{R}^{w \times w}$ to represent the diagonal matrix containing all singular values of $A$ with $\frac{1}{1+\epsilon/2}\sigma_k \leq \sigma_i \leq \sigma_k$ without diagonal entries of 0.

We next apply the argument used to prove Property 1 to $p_2(A_{outer})\Pi$. The $(k+1)^{th}$ singular value of $A_{outer}$ is equal to $\sigma_{k+w+1} \leq \frac{1}{1+\epsilon/2}\sigma_k = \alpha$. So applying (7) we have for all $l \leq k$,

$$\|A_l\|_F^2 - \|Y_{outer}\|_l (Y_{outer})_l^\top A_l\|_F^2 \leq \frac{\epsilon\sigma_k^2}{2}. \quad (10)$$

Note that $A_{outer}$ has the same top $k$ singular vectors at $A$ so $(A_{outer})_l = A_l$. Let $x' \in \mathbb{R}^n$ be any unit vector within the column space of $Y_{outer}$ and let $Y_{outer} = (I-x'x'^\top)Y_{outer}$, i.e. the matrix with $x'$ projected off each column. We can use (10) and the optimality of the SVD for low rank approximation to obtain:

$$\|A_k\|_F^2 - \|Y_{outer}Y_{outer}^\top A_k\|_F^2 = \|A_k\|_F^2 - \|Y_{outer}Y_{outer}^\top A_k\|_F^2 \leq \frac{\epsilon\sigma_k^2}{2}$$

$$\|A_k\|_F^2 - \|A_k-1\|_F^2 - \frac{\epsilon\sigma_k^2}{2} \leq \|x'x'^\top A_k\|_F^2$$

$$(1-\epsilon/2)\sigma_k^2 \leq \|x^\top A\|_F^2. \quad (11)$$

Plugging (9) and (11) into (8) yields that, for any $x \in \text{span}(Y_2)$, i.e. $\text{span}(p_2(A)\Pi)$,

$$\|x^\top A\|_2^2 = \|x_{inner}^\top A\|_2^2 + \|x_{outer}^\top A\|_2^2 \geq (\|x_{inner}\|_2^2 + \|x_{outer}\|_2^2) (1-\epsilon)\sigma_k^2 \geq (1-\epsilon)\sigma_k^2. \quad (12)$$

So, we have identified a rank $k$ subspace $Y_2$ within our Krylov subspace such that every vector in its span aligns at least as well with $A$ as $u_k$.

Now, for any $m \leq l \leq k$, consider $E(Z_l, A)$. We know that given $Z_{l-1}$, we can form a rank $l$ matrix $\tilde{Z}_l$ in our Krylov subspace simply by appending a column $x$ orthogonal to the $l-1$ columns of $Z_{l-1}$ but falling in the span of $Y_2$. Since $Y_2$ has rank $k$, finding such a column is always always
possible. Since \( Z_l \) is the optimal rank \( l \) Frobenius norm approximation to \( A \) falling within our Krylov subspace we have:

\[
\mathcal{E}(Z_l, A) \leq \mathcal{E}(Z_l, A) = \|A_l\|_F^2 - \|Z_lZ_l^\top A\|_F^2 \\
= \sigma_l^2 + \|A_{l-1}\|_F^2 - \|Z_{l-1}Z_{l-1}^\top A\|_F^2 - \|xx^\top A\|_F^2 \\
= \mathcal{E}(Z_{l-1}, A) + \sigma_l^2 - \|xx^\top A\|_F^2 \\
\leq \mathcal{E}(Z_{l-1}, A) + (1 + \epsilon/2) \sigma_{k+1}^2 - (1 - \epsilon) \sigma_{k+1}^2 \\
\leq \mathcal{E}(Z_{l-1}, A) + 3\epsilon \cdot \sigma_{k+1}^2,
\]

which gives Property 2.

Again, a nearly identical proof applies for Algorithm 1. We just choose \( p_2(x) = \sigma_k \left( \frac{x}{\sigma_k} \right)^{2q+1} \). For \( q = \Theta(\log d/\epsilon) \) this polynomial satisfies the necessary properties: for all \( i \geq k \), \( p_1(\sigma_i) \leq O \left( \frac{\sigma_i}{\sigma_k^2} \right) \) and for all \( i \leq k \), \( \sigma_i \leq p_2(\sigma_i) \).

**Proof of Property 3**

By Properties 1 and 2 we already have, for all \( l \leq k \), \( \mathcal{E}(Z_l, A) \leq \epsilon \sigma_{k+1}^2 + (l - m) \cdot 3\epsilon \sigma_{k+1}^2 \leq (1 + k - m) \cdot 3\epsilon \sigma_{k+1}^2 \). So if \( k - m \leq w \) then we immediately have Property 3.

Otherwise, \( w < k - m \) so \( w < k \) and thus \( p_2(A_{inner}) \Pi \in \mathbb{R}^{n \times k} \) only has rank \( w \). It has a null space of dimension \( k - w \). Choose any \( z \) in this null space. Then \( p_2(A) \Pi z = p_2(A_{inner}) \Pi z + p_2(A_{outer}) \Pi z = p_2(A_{outer}) \Pi z \). In other words, \( p_2(A) \Pi z \) falls entirely within the span of \( Y_{outer} \).

So, there is a \( k - w \) dimensional subspace of \( \text{span}(Y_2) \) that is entirely contained in \( \text{span}(Y_{outer}) \). For \( l \leq m + w \), then Properties 1 and 2 already give us \( \mathcal{E}(Z_l, A) \leq \epsilon \sigma_{k+1}^2 + (l - m) \cdot 3\epsilon \sigma_{k+1}^2 \leq (w + 1) \cdot 3\epsilon \sigma_{k+1}^2 \). So consider \( m + w \leq l \leq k \). Given \( Z_m \), to form a rank \( l \) matrix \( Z_l \) in our Krylov subspace we need to append \( m \) orthonormal columns. We can choose \( m \) \( k - w - m \), \( l - m \) columns, \( X_1 \), from the \( k - w \) dimensional subspace within \( \text{span}(Y_2) \) that is entirely contained in \( \text{span}(Y_{outer}) \). If necessary (i.e. \( k - w - m \leq l - m \)), We can then choose the remaining \( l - (k - w) \) columns \( X_2 \) from the span of \( Y_2 \).

Similar to our argument when considering a single vector in the span of \( Y_{outer} \), letting \( Y_{outer} = (I - X_1X_1^\top) Y_{outer} \), we have by (10):

\[
\|A_k\|_F^2 - \|Y_{outer}Y_{outer}^\top A_k\|_F^2 \leq \frac{\epsilon \sigma_{k+1}^2}{2} \\
\|A_k\|_F^2 - \|\overline{Y}_{outer} \overline{Y}_{outer}^\top A_k\|_F^2 - \|X_1X_1^\top A_k\|_F^2 \leq \frac{\epsilon \sigma_{k+1}^2}{2} \\
\|A_k\|_F^2 - \|A_{k-[k-w-m,l-m]}\|_F^2 \leq \frac{\epsilon \sigma_k^2}{2} \leq \|X_1X_1^\top A_k\|_F^2 \\
\sum_{i=k-[k-w-m,l-m]+1}^k \sigma_i^2 \leq \|X_1X_1^\top A_k\|_F^2.
\]

By applying (12) directly to each column of \( X_2 \) we also have:

\[
(l + w - k)\sigma_k^2 - (l + w - k)\sigma_k^2 \leq \|X_2X_2^\top A\|_F^2 \\
(l + w - k)\sigma_{k+1}^2 - (l + w - k)\sigma_{k+1}^2 \leq \|X_2X_2^\top A\|_F^2.
\]
Assume that \( \min\{k - w - m, l - m\} = k - w - m \). Similar calculations show the same result when \( \min\{k - w - m, l - m\} = l - m \). We can use the above two bounds to obtain:

\[
\mathcal{E}(Z_l, A) \leq \mathcal{E}(Z_l, A) = \|A_l\|_F^2 - \|Z_l Z_l^T A\|_F^2
\]

\[
= \sum_{i=m+1}^l \sigma_i^2 + \|A_m\|_F^2 - \|Z_m Z_m^T A\|_F^2 - \|X_1 X_1^T A\|_F^2 - \|X_2 X_2^T A\|_F^2
\]

\[
\leq \mathcal{E}(Z_m, A) + \sum_{i=m+1}^{l-w} \sigma_i^2 - \sum_{i=m+1}^k \sigma_i^2 + \frac{\epsilon \sigma_{k+1}^2}{2} - (l + w - k)\sigma_{k+1}^2 + (l + w - k)\epsilon \sigma_{k+1}^2
\]

\[
\leq \sum_{i=m+1}^{m+w} \sigma_i^2 - w\sigma_{k+1}^2 + (l + w - k + 3/2)\epsilon \sigma_{k+1}^2
\]

\[
\leq (w + 1) \cdot 3\epsilon \cdot \sigma_{k+1}^2
\]

giving Property 3 for all \( l \leq k \).

\[\square\]

### 5.2 Error Bounds for Simultaneous Iteration and Block Lanczos

With Lemma 9 in place, we can easily prove that Simultaneous Iteration and Block Lanczos both achieve the low rank approximation and PCA guarantees (1), (2), and (3).

**Theorem 10** (Near Optimal Spectral Norm Error Approximation). With probability 99/100, Algorithms 1 and 2 return \( Z \) satisfying (2):

\[
\|A - ZZ^T A\|_2 \leq (1 + \epsilon)\|A - A_k\|_2.
\]

**Proof.** Let \( m \) be the number of singular values with \( \sigma_i \geq (1 + \epsilon/2)\sigma_{k+1} \). If \( m = 0 \) then we are done since any \( Z \) will satisfy \( \|A - ZZ^T A\|_2 \leq \|A\|_2 = \sigma_1 \leq (1 + \epsilon/2)\sigma_{k+1} \leq (1 + \epsilon)\|A - A_k\|_2 \). Otherwise, by Property 1 of Lemma 9,

\[
\mathcal{E}(Z_m, A) \leq \frac{\epsilon \sigma_{k+1}^2}{2}
\]

\[
\|A - Z_m Z_m^T A\|_F^2 \leq \|A - A_m\|_F^2 + \frac{\epsilon \sigma_{k+1}^2}{2}.
\]

Additive error in Frobenius norm directly translates to additive spectral norm error. Specifically, applying Theorem 3.4 of [Gu14], which we also prove as Lemma 15 in Appendix A,

\[
\|A - Z_m Z_m^T A\|_2^2 \leq \|A - A_m\|_2^2 + \frac{\epsilon \sigma_{k+1}^2}{2} \leq \sigma_{k+1}^2 + \frac{\epsilon \sigma_{k+1}^2}{2}
\]

\[
\leq (1 + \epsilon/2)\sigma_{k+1}^2 + \frac{\epsilon \sigma_{k+1}^2}{2} \leq (1 + \epsilon)\|A - A_k\|_2^2.
\]

Finally, \( Z_m Z_m^T A = ZZ_{m}^T A \) and so by Lemma 3 we have \( \|A - ZZ^T A\|_2^2 \leq \|A - Z_m Z_m^T A\|_2^2 \), which combines with (13) to give the result.

\[\square\]
Theorem 11 (Near Optimal Frobenius Norm Error Approximation). With probability 99/100, Algorithms 1 and 2 return \( \mathbf{Z} \) satisfying (1):

\[
\| \mathbf{A} - \mathbf{ZZ}^\top \mathbf{A} \|_F \leq (1 + \epsilon)\| \mathbf{A} - \mathbf{A}_k \|_F.
\]

Proof. By Property 3 of Lemma 9 we have:

\[
\mathcal{E}(\mathbf{Z}_i; \mathbf{A}) \leq (w + 1) \cdot 3 \epsilon \cdot \sigma_{k+1}^2
\]

\[
\| \mathbf{A} - \mathbf{ZZ}^\top \mathbf{A} \|_F^2 \leq \| \mathbf{A} - \mathbf{A}_k \|_F^2 + (w + 1) \cdot 3 \epsilon \cdot \sigma_{k+1}^2.
\]

(14)

\( w \) is defined as the number of singular values with \( \frac{1}{1 + \epsilon/2} \sigma_k \leq \sigma_i < \sigma_k \). So \( \| \mathbf{A} - \mathbf{A}_k \|_F^2 \geq w \cdot (\frac{1}{1 + \epsilon/2} \sigma_k)^2 \). Plugging into (14) we have:

\[
\| \mathbf{A} - \mathbf{ZZ}^\top \mathbf{A} \|_F^2 \leq \| \mathbf{A} - \mathbf{A}_k \|_F^2 + (w + 1) \cdot 3 \epsilon \cdot \sigma_{k+1}^2 \leq (1 + 10 \epsilon)\| \mathbf{A} - \mathbf{A}_k \|_F^2.
\]

Adjusting constants on the \( \epsilon \) gives us the result.

Theorem 12 (Per Vector Quality Guarantee). With probability 99/100, Algorithms 1 and 2 return \( \mathbf{Z} \) satisfying (3):

\[
\forall i, \left| \mathbf{u}_i^\top \mathbf{A} \mathbf{u}_i - \mathbf{z}_i^\top \mathbf{A} \mathbf{z}_i \right| \leq \epsilon \sigma_{k+1}^2.
\]

Proof. First note that \( \mathbf{z}_i^\top \mathbf{A} \mathbf{z}_i \leq \mathbf{u}_i^\top \mathbf{A} \mathbf{z}_i \). This is because \( \mathbf{z}_i^\top \mathbf{A} \mathbf{z}_i = \mathbf{z}_i^\top \mathbf{Q} \mathbf{Q}^\top \mathbf{A} \mathbf{z}_i = \sigma_i(\mathbf{Q}^\top \mathbf{A}) \) by our choice of \( \mathbf{z}_i \). \( \sigma_i(\mathbf{Q}^\top \mathbf{A}) \leq \sigma_i(\mathbf{A})^2 \) (since applying a projection to \( \mathbf{A} \) will decrease each of its singular values (which follows for example from the Courant-Fischer min-max principle). Then by Property 2 of Lemma 9 we have, for all \( i \leq k \),

\[
\| \mathbf{A}_i \|_F^2 - \| \mathbf{Z}_i \|_F^2 \leq \| \mathbf{A}_{i-1} \|_F^2 - \| \mathbf{Z}_{i-1} \|_F^2 + 3 \epsilon \sigma_{k+1}^2
\]

\[
\sigma_i^2 \leq \| \mathbf{z}_i \mathbf{z}_i^\top \mathbf{A} \|_F^2 + 3 \epsilon \sigma_{k+1}^2 = \mathbf{z}_i^\top \mathbf{A} \mathbf{z}_i + 3 \epsilon \sigma_{k+1}^2.
\]

\( \sigma_i^2 = \mathbf{u}_i^\top \mathbf{A} \mathbf{u}_i \), so simply adjusting constants on \( \epsilon \) gives the result.

5.3 Improved Convergence With Spectral Decay

In addition to the traditional Simultaneous Iteration and Block Lanczos methods (Algorithms 1 and 2), our analysis applies to the common modification of running the algorithms with \( \Pi \in \mathbb{R}^{n \times p} \) for \( p \geq k \) [RST09, HMT11]. This technique can significantly accelerate both methods for matrices with decaying singular values. For simplicity, we focus on Block Lanczos, although all arguments immediately extend to the simpler Simultaneous Iteration.

In order to avoid inverse dependence on the potentially small singular value gap \( \frac{\sigma_k}{\sigma_{k+1}} - 1 \), the number of iterations of Block Lanczos inherently depends on \( 1/\sqrt{c} \). This ensures that our matrix polynomial sufficiently separates small singular values from larger ones. However, when \( \sigma_k > (1 + \epsilon)\sigma_{k+1} \) we can actually use \( q = \Theta \left( \frac{\log(d/\epsilon)/\sqrt{\min\{1, 1/\sigma_{k+1}\}}}{\sigma_{k+1}} \right) \) iterations, which is sufficient for separating the top \( k \) singular values significantly from the lower values. Specifically, in the Block Lanczos case, if we set \( \alpha = \sigma_{k+1} \) and \( \gamma = \frac{\sigma_k}{\sigma_{k+1}} - 1 \), we know that with
\( q = \Theta \left( \frac{\log(d/\epsilon)}{\sqrt{\min\{1, \frac{\sigma_k}{\sigma_{k+1}} - 1\}}}, \right) \), (5) still holds. We can then just follow the proof of Lemma 9 and show that Property 1 holds for all \( l \leq k \) (not just for \( l \leq m \) as originally proven). This gives Property 2 and Property 3 trivially.

Furthermore, for \( p \geq k \), the exact same analysis shows that \( q = \Theta \left( \frac{\log(d/\epsilon)}{\sqrt{\min\{1, \frac{\sigma_k}{\sigma_{p+1}} - 1\}}} \right) \) suffices. When \( A \)'s spectrum decays rapidly, so \( \sigma_{p+1} \leq c \cdot \sigma_k \) for some constant \( c < 1 \) and some \( p \) not much larger than \( k \), we can obtain significantly faster runtimes. Our \( \epsilon \) dependence becomes logarithmic, rather than polynomial:

**Theorem 13** (Gap Dependent Convergence). With probability 99/100, for any \( p \geq k \), Algorithm 1 or 2 initialized with \( \Pi \sim \mathcal{N}(0,1)^{d \times p} \) returns \( Z \) satisfying guarantees (1), (2), and (3) as long as we set \( q = \Theta \left( \frac{\log(d/\epsilon)}{\sqrt{\min\{1, \frac{\sigma_k}{\sigma_{p+1}} - 1\}}}, \right) \) or \( \Theta \left( \frac{\log(d/\epsilon)}{\sqrt{\min\{1, \frac{\sigma_k}{\sigma_{p+1}} - 1\}}}, \right) \), respectively.

This theorem may prove especially useful in practice because, on many architectures, multiplying a large \( A \) by 2\( k \) or even 10\( k \) vectors is not much more expensive than multiplying by \( k \) vectors. Additionally, it should still be possible to perform all steps for post-processing \( K \) in memory, again limiting additional runtime costs due to its larger size.

Finally, we note that while Theorem 13 is more reminiscent of classical gap-dependent bounds, it still takes substantial advantage of the fact that we’re looking for nearly optimal low rank approximations and principal components instead of attempting to converge precisely to \( A \)'s true singular values. This allows the result to avoid dependence on the gap between adjacent singular values, instead varying only with \( \frac{\sigma_k}{\sigma_{p+1}} \), which should be much larger.

### 6 Acknowledgements

We thank David Woodruff, Aaron Sidford, and Richard Peng for several valuable conversations. Additionally, Michael Cohen was very helpful in discussing many details of this project, including the ultimate form of Lemma 9. This work was partially supported by NSF Graduate Research Fellowship Grant No. 1122374, AFOSR grant FA9550-13-1-0042, DARPA grant FA8650-11-C-7192, and the NSF Center for Science of Information.

### References

[Bau57] Friedrich L. Bauer. Das verfahren der treppeniteration und verwandte verfahren zur lösung algebraischer eigenwertprobleme. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 8(3):214–235, 1957.

[BDMI14] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014. Preliminary version in the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2011.

[CD74] Jane Cullum and W.E. Donath. A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. In *IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes*, pages 505–509, 1974.
Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 81–90, 2013.

Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004. Preliminary version in the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1999.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1):158–183, 2006.

Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Proceedings of the 10th International Workshop on Randomization and Computation (RANDOM)*, pages 292–303, 2006.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004. Preliminary version in the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1998.

Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, 1996.

Gene H. Golub, Franklin T. Luk, and Michael L. Overton. A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Trans. Math. Softw.*, 7(2):149–169, 1981.

Gene Golub and Richard Underwood. The block Lanczos method for computing eigenvalues. *Mathematical Software*, (3):361–377, 1977.

Ming Gu. Subspace iteration randomization and singular value problems. *Computing Research Repository (CoRR)*, abs/1408.2208, 2014.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

Nathan P Halko. *Randomized methods for computing low-rank approximations of matrices*. PhD thesis, University of Colorado, 2012.

Nathan Halko, Per-Gunnar Martinsson, Yoel Shkolnisky, and Mark Tygert. An algorithm for the principal component analysis of large data sets. *SIAM J. Sci. Comput.*, 33(5):2580–2594, 2011.
[HMT11] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[IBM14] IBM Research Division, Skylark Team. libskylark: Sketching-based Distributed Matrix Computations for Machine Learning. IBM Corporation, Armonk, NY, 2014.

[Lan50] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4), 1950.

[Lar01] Rasmus Munk Larsen. Combining implicit restart and partial reorthogonalization in Lanczos bidiagonalization. http://sun.stanford.edu/~rmunk/PROPACK/paper.pdf, 2001. Presentation at U.C. Berkeley, sponsored by Stanford’s Scientific Computing and Computational Mathematics (succeeded by the Institute for Computational and Mathematical Engineering).

[Lar05] Rasmus Munk Larsen. *PROPACK*: Software for large and sparse SVD calculations. Stanford University, Solar Observatories Group, Stanford, CA, 2005. http://sun.stanford.edu/~rmunk/PROPACK/.

[ME11] Aditya Krishna Menon and Charles Elkan. Fast algorithms for approximating the singular value decomposition. *ACM Transactions on Knowledge Discovery from Data*, 5(2):13:1–13:36, 2011. http://cseweb.ucsd.edu/~akmenon/code/.

[MH02] J.C. Mason and D.C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2002.

[Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11:50–59, 1960.

[MM13] Michael W Mahoney and Xiangrui Meng. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 91–100, 2013.

[MRT06] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the approximation of matrices. Technical Report 1361, Yale University, 2006.

[NN13] Jelani Nelson and Huy L. Nguyen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 117–126, 2013.

[PTRV00] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–235, 2000. Preliminary version in the 17th Symposium on Principles of Database Systems (PODS), 1998.

[RST09] Vladimir Rokhlin, Arthur Szlam, and Mark Tygert. A randomized algorithm for principal component analysis. *SIAM J. Matrix Anal. Appl.*, 31(3):1100–1124, 2009.

[Rut70] H. Rutishauser. Simultaneous iteration method for symmetric matrices. *Numerische Mathematik*, 16(3):205–223, 1970.
A Appendix

Frobenius Norm Low Rank Approximation

We first give a deterministic Lemma, from which the main approximation result follows.

Lemma 14 (Special case of Lemma 4.4 of [Woo14], originally proven in [BDMI14]). Let $A \in \mathbb{R}^{n \times d}$ have SVD $A = U \Sigma V^\top$, let $S \in \mathbb{R}^{d \times k}$ be any matrix such that rank $(V_k^\top S) = k$, and let $C \in \mathbb{R}^{n \times k}$ be an orthonormal basis for the column span of $AS$. Then:

$$\|A - CC^\top A\|_F^2 \leq \|A - A_k\|_F^2 + \| (A - A_k) S (V_k^\top S)^+ \|_F^2.$$

Lemma 4 (Frobenius Norm Low Rank Approximation). Take any $A \in \mathbb{R}^{n \times d}$ and $\Pi \in \mathbb{R}^{d \times k}$ where the entries of $\Pi$ are independent Gaussians drawn from $\mathcal{N}(0, 1)$. If we let $Z$ be an orthonormal basis for span $(A \Pi)$, then with probability at least $99/100$, for some fixed constant $c$,

$$\|A - ZZ^\top A\|_F^2 \leq c \cdot dk \|A - A_k\|_F^2.$$
Proof. We follow [Woo14]. Apply Lemma 14 with \( S = I \). With probability 1, \( V_k^T S \) has full rank. So, to show the result we need to show that \( \| (A - A_k) S (V_k^T S)^+ \|_2^2 \leq c \| A - A_k \|_F^2 \) for some fixed \( c \). For any two matrices \( M \) and \( N \), \( \| MN \|_F \leq \| M \|_F \| N \|_2 \). This property is known as **spectral submultiplicativity**. Noting that \( \| U_{r \setminus k} \Sigma_{r \setminus k} \|_F = \| A - A_k \|_F^2 \) and applying submultiplicativity,

\[
\| (A - A_k) S (V_k^T S)^+ \|_2^2 \leq \| U_{r \setminus k} \Sigma_{r \setminus k} \|_F^2 \| V_{r \setminus k} S \|_2^2 \| (V_k^T S)^+ \|_2^2.
\]

By the rotational invariance of the Gaussian distribution, since the rows of \( V^T \) are orthonormal, the entries of \( V_k^T S \) and \( V_{r \setminus k} S \) are independent Gaussians. By standard Gaussian matrix concentration results (Fact 6 of [Woo14], also in [RV10]), with probability at least 99/100, \( \| V_{r \setminus k} S \|_2^2 \leq c_1 \cdot \max\{k, r - k\} \leq c_1 \hat{d} \) and \( \| (V_k^T S)^+ \|_2^2 \leq c_2 k \) for some fixed constants \( c_1, c_2 \). So,

\[
\| U_{r \setminus k} \Sigma_{r \setminus k} \|_F^2 \| V_{r \setminus k} S \|_2^2 \| (V_k^T S)^+ \|_2^2 \leq c \cdot d \| A - A_k \|_F^2
\]

for some fixed \( c \), yielding the result. Note that we choose probability 99/100 for simplicity – we can obtain a result with higher probability by simply allowing for a higher constant \( c \), which in our applications of Lemma 4 will only factor into logarithmic terms. \( \square \)

**Chebyshev Polynomials**

**Lemma 5** (Chebyshev Minimizing Polynomial). Given a specified value \( \alpha > 0 \), gap \( \gamma \in (0, 1] \), and degree \( q \geq 1 \), there exists a degree \( q \) polynomial \( p(x) \) such that:

1. \( p((1 + \gamma)\alpha) = (1 + \gamma)\alpha \)
2. \( p(x) \geq x \) for all \( x \geq (1 + \gamma)\alpha \)
3. \( p(x) \leq \frac{\alpha}{2\sqrt{\gamma + 1}} \) for all \( x \in [0, \alpha] \)

**Proof.** The required polynomial can be constructed using a standard Chebyshev polynomial of degree \( q \), \( T_q(x) \), which is defined by the three term recurrence:

\[
T_0(x) = 1 \\
T_1(x) = x \\
T_q(x) = 2xT_{q-1}(x) - T_{q-2}(x)
\]

Each Chebyshev polynomial satisfies the well known property that \( T_q(x) \leq 1 \) for all \( x \in [-1, 1] \) and we can write the polynomials in closed form [MH02]:

\[
T_q(x) = \frac{(x + \sqrt{x^2 - 1})^q + (x - \sqrt{x^2 - 1})^q}{2}.
\]

For Lemma 5, we simply set:

\[
p(x) = (1 + \gamma)\alpha \frac{T_q(x/\alpha)}{T_q(1 + \gamma)}.
\]
which is clearly of degree $q$ and well defined since, referring to (15), $T_q(x) > 0$ for all $x > 1$. Now,

$$p((1 + \gamma)\alpha) = (1 + \gamma)\alpha \frac{T_q(1 + \gamma)}{T_q(1 + \epsilon)} = (1 + \gamma)\alpha,$$

so $p(x)$ satisfies property 1. With property 1 in place, to prove that $p(x)$ satisfies property 2, it suffices to show that $p'(x) \geq 1$ for all $x \geq (1 + \gamma)\alpha$. By chain rule,

$$p'(x) = \frac{(1 + \gamma)}{T_q(1 + \gamma)} T'_q(x/\alpha).$$

Thus, it suffices to prove that, for all $x \geq (1 + \gamma)$,

$$(1 + \gamma)T'_q(x) \geq T_q(1 + \gamma). \quad (17)$$

We do this by showing that $(1 + \gamma)T'_{q}(1 + \gamma) \geq T_q(1 + \gamma)$ and then claim that $T''_q(x) \geq 0$ for all $x > (1 + \gamma)$, so (17) holds for $x > (1 + \gamma)$ as well. A standard form for the derivative of the Chebyshev polynomial is

$$T'_q = \begin{cases} 2q(T_{q-1} + T_{q-3} + \ldots + T_1) & \text{if } q \text{ is even}, \\ 2q(T_{q-1} + T_{q-3} + \ldots + T_2) + q & \text{if } q \text{ is odd}. \end{cases} \quad (18)$$

(18) can be verified via induction once noting that the Chebyshev recurrence gives $T'_q = 2xT'_{q-1} + 2T_{q-1} - T_q$. Since $T_i(x) > 0$ when $x \geq 1$, we can conclude that $T'_q(x) \geq 2qT_{q-1}(x)$. So proving (17) for $x = (1 + \gamma)$ reduces to proving that

$$(1 + \gamma)2qT_{q-1}(1 + \gamma) \geq T_q(1 + \gamma). \quad (19)$$

Noting that, for $x \geq 1$, $(x + \sqrt{x^2 - 1}) > 0$ and $(x - \sqrt{x^2 - 1}) > 0$, it follows from (15) that

$$T_{q-1}(x) \left( (x + \sqrt{x^2 - 1}) + (x - \sqrt{x^2 - 1}) \right) \geq T_q(x),$$

and thus

$$\frac{T_q(x)}{T_{q-1}(x)} \leq 2x.$$

So, to prove (19), it suffices to show that $2(1 + \gamma) \leq (1 + \gamma)2q$, which is true whenever $q \geq 1$. So (17) holds for all $x = (1 + \gamma)$.

Finally, referring to (18), we know that $T''_q$ must be some positive combination of lower degree Chebyshev polynomials. Again, since $T_i(x) > 0$ when $x \geq 1$, we conclude that $T''_q(x) \geq 0$ for all $x \geq 1$. It follows that $T'_q(x)$ does not decrease above $x = (1 + \gamma)$, so (17) also holds for all $x > (1 + \gamma)$ and we have proved property 2.

To prove property 3, we first note that, by the well known property that $T_i(x) \leq 1$ for $x \in [-1, 1]$, $T_q(x/\alpha) \leq 1$ for $x \in [0, \alpha]$. So, to prove $p(x) \leq \frac{\alpha}{2q\sqrt{\alpha - 1}}$, we just need to show that

$$\frac{1}{T_q(1 + \gamma)} \leq \frac{1}{2q\sqrt{\alpha - 1}}. \quad (20)$$

Equation (15) gives $T_q(1 + \gamma) \geq \frac{\gamma}{2}(1 + \gamma + \sqrt{(1 + \gamma)^2 - 1})^q \geq \frac{\gamma}{2}(1 + \sqrt{\gamma})^q$. When $\gamma \leq 1$, $(1 + \sqrt{\gamma})^{1/\sqrt{\gamma}} \geq 2$. Thus, $(1 + \sqrt{\gamma})^q \geq 2q\sqrt{\gamma}$. Dividing by 2 gives $T_q(1 + \gamma) \geq 2q\sqrt{\gamma - 1}$, which gives (20) and thus property 3. \qed
Additive Frobenius Norm Error Implies Additive Spectral Norm Error

Lemma 15 (Theorem 3.4 of [Gu14]). For any \( A \in \mathbb{R}^{n \times d} \), let \( B \in \mathbb{R}^{n \times d} \) be any rank \( k \) matrix satisfying \( \|A - B\|_F^2 \leq \|A - A_k\|_F^2 + \eta \). Then

\[
\|A - B\|_F^2 \leq \|A - A_k\|_F^2 + \eta.
\]

Proof. We follow the proof given in [Gu14] nearly exactly, including it for completeness. By Weyl’s monotonicity theorem (Theorem 3.2 in [Gu14]), for any two matrices \( X, Y \in \mathbb{R}^{n \times d} \) with \( n \geq d \), for all \( i, j \) with \( i + j - 1 \leq n \) we have \( \sigma_{i+j-1}(X + Y) \leq \sigma_i(X) + \sigma_j(X) \). If we write \( A = (A - B) + B \) and apply this theorem, then for all \( 1 \geq i \geq n - k \),

\[
\sigma_{i+k}(A) \leq \sigma_i(A - B) + \sigma_{k+1}(B).
\]

Note that if \( n < d \), we can just work with \( A^\top \) and \( B^\top \). Now, \( \sigma_{k+1}(B) = 0 \) since \( B \) is rank \( k \), so:

\[
\|A - B\|_F^2 \leq \|A - A_k\|_F^2 + \eta
\]

\[
\sum_{i=1}^{n} \sigma_i^2(A - B) \leq \sum_{i=k+1}^{n} \sigma_i^2(A) + \eta
\]

\[
\sum_{i=1}^{n-k} \sigma_i^2(A - B) \leq \sum_{i=k+1}^{n} \sigma_i^2(A) + \eta
\]

\[
\sigma_k^2(A - B) + \sum_{i=2}^{n-k} \sigma_i^2(A) \leq \sum_{i=k+1}^{n} \sigma_i^2(A) + \eta
\]

\[
\sigma_1^2(A - B) \leq \sum_{i=k+1}^{n} \sigma_i^2(A) - \sum_{i=2}^{n-k} \sigma_i^2(A) + \eta
\]

\[
\sigma_1^2(A - B) \leq \sigma_{k+1}^2(A) + \eta.
\]

\( \square \)