FAST ESTIMATION OF $\mathtt{tr}(F(A))$ VIA STOCHASTIC LANCZOS QUADRATURE

SHASHANKA UBARU *, JIE CHEN †, AND YOUSEF SAAD*

Abstract. The problem of estimating the trace of matrix functions appears in applications ranging from machine learning, to scientific computing, and computational biology to name just a few. This paper presents an *inexpensive* method to estimate the trace of $f(A)$ for cases where f is analytic inside a closed interval. The method combines three key ingredients, namely, the stochastic trace estimator, Gaussian quadrature, and the Lanczos algorithm. As examples, we consider the problems of estimating the log-determinant ($f(t) = \log(t)$), the Schatten p-norms ($f(t) = t^{p/2}$), the Estrada index ($f(t) = e^t$) and the trace of a matrix inverse ($f(t) = \frac{1}{t}$). We establish multiplicative and additive error bounds for the approximations obtained by the method. In addition, we present error bounds for other useful tools such as approximating the log-likelihood function in the context of maximum likelihood estimation of Gaussian processes. Numerical experiments illustrate the performance of the proposed method on different problems arising from various applications.

1. Introduction. The problem of estimating the trace of matrix functions appears frequently in applications of machine learning, signal processing, scientific computing, statistics, computational biology and computational physics [5, 16, 36, 34, 19, 28, 30, 2, 25]. Developing fast and scalable algorithms to perform this task has long been a primary focus of research in these and other fields. An important instance of the trace estimation problem is that of approximating $\log(Det(A))$, called the log-determinant of a matrix A. Log-determinants of covariance and precision matrices play an important role in Gaussian graphical models and Gaussian processes [36, 34]. Log-determinant computations also appear in applications such as kernel learning [13], discrete probabilistic models [1], Bayesian Learning [32], spatial statistics [3] and Markov field models [42, 25, 8].

Another instance of the trace estimation problem in applications is that of estimating Schatten p-norms, particularly the nuclear norm, since this norm is used as the convex surrogate of the matrix rank. This appears in convex optimization problems, e.g., in the context of matrix completion [9], in differential privacy problems [26], and in sketching and streaming models [30, 2]. In uncertainty quantification and in lattice quantum chromodynamics [28, 43] it is necessary to estimate the trace of the inverse of covariance matrices. Estimating the Estrada index (trace of exponential function), is another illustration of the problem. Other applications include protein indexing [16], statistical thermodynamics [17] and information theory [10].

The trace estimation problems mentioned above can be formulated as follows: given a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$, compute an approximation of the trace of the matrix function $f(A)$, i.e.,

$$\mathtt{tr}(f(A)) = \sum_{i=1}^{n} f(\lambda_i), \tag{1.1}$$

where λ_i, $i = 1, \ldots, n$ are the eigenvalues of A, and f is the desired function. A naive approach for estimating the trace of matrix functions is to compute this trace from the eigenvalues of the matrix. A popular approach to compute the log-determinant of a matrix is to exploit its Cholesky decomposition [21]. Indeed, given the Cholesky decomposition $A = LL^\top$, the log-determinant of A is given by $\log\det(A) = 2\sum_i \log(L_{ii})$. Computing the Schatten norms in a standard way would typically require the singular value decomposition (SVD) of the matrix. These methods have cubic computational complexity (in terms of the matrix dimension, i.e., $O(n^3)$ cost) in general, and are not viable for large scale applications. In this paper, we study inexpensive methods for accurately estimating these traces of functions of large matrices.

This paper is a study of the a method we call the Stochastic Lanczos Quadrature (SLQ) for approximating the trace of functions of large matrices [5, 6, 19]. The method combines three key ingredients. First, the stochastic trace estimator, also called the *Hutchinson method* [27] is considered for approximating the trace.

*Computer Science & Engineering, University of Minnesota, Twin Cities. {ubaru,saad} @umn.edu. The work of these authors was supported by NSF under grant NSF/CCF-1318597.

†IBM (affiliation) — Work supported in part by the XDATA program of the Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

Next, the quadratic form that appears in the trace estimator is expressed as a Riemann-Stieltjes integral, and the Gauss quadrature rule is used to approximate this integral. Finally, the Lanczos algorithm is used to obtain the weights and the nodes of the quadrature rule (see section 3 for details). We establish multiplicative and additive approximation error bounds for the trace obtained by using the method. To the best of our knowledge, such error bounds for SLQ have not appeared in the prior literature. We show that the Lanczos Quadrature approximation will have faster convergence rate compared to popular methods such as those based on Chebyshev or Taylor series expansions. The analysis can be extended to any matrix functions that are analytic inside a closed interval and are analytically continuable to an open Bernstein ellipse [39].

We consider several important trace estimation problems and their applications. We discuss the log-determinant computation, estimation of the Estrada index and the trace of matrix inverse, and show how the SLQ method can be used to approximately estimate these rapidly. We also adapt our method for fast estimation of the nuclear norm and Schatten-p norms of large matrices. In addition, we establish error bounds for the approximation of log-likelihoods in the context of maximum likelihood estimation of Gaussian processes. Several numerical experiments are presented to demonstrate the superiority of the proposed method over existing methods in practice.

A plethora of methods have been developed in the literature to deal with trace estimation problems. In the following, we discuss some of the works that are closely related to SLQ, particularly those that invoke the stochastic trace estimator. The stochastic trace estimator has been employed for a number of applications in the literature, for example, for estimating the diagonal of a matrix [7], for counting eigenvalues inside an interval [15], score function approximation in Gaussian processes [38], and numerical rank estimation [40, 41]. For the log-determinant computation, a few methods have been proposed, which also invoke the stochastic trace estimator. These methods differ in the approach used to approximate the log function. Article [25] used the Chebyshev polynomial approximations for the log function. The log function was approximated using the Taylor series expansions in [44]. Article [8] provided an improved analysis for the log-determinant computations using these Taylor series expansions. Aune et. al [3] adopted the method proposed in [23] to estimate the log function. Here, the Cauchy integral formula of the log function is considered and the Trapezoidal rule is invoked to approximate the integral. This method is equivalent to using a rational approximation for the function. The method requires solving a series of linear systems and is generally expensive. The functions can also be approximated by means of least squares polynomials as proposed in [12].

Not many fast algorithms are available in the literature to approximate the nuclear norm and Schatten-p norms; see [30, 2] for discussions. Article [24] extends the idea of using Chebyshev expansions developed in [15, 25] to for approximate the trace of various matrix functions including Schatten norms, the Estrada index and the trace of matrix inverse. Related articles on estimating the trace of the inverses of matrices are [43, 11].

A key objective of this work is to demonstrate how the powerful Lanczos algorithm can be employed to solve trace estimation problems for matrix functions. The Lanczos algorithm has clear advantages over the above mentioned (recently proposed) methods such as Chebyshev expansions, Taylor series expansions and rational function approximations. To understand the pros and cons of the Lanczos Quadrature method, let us first examine the three classes of techniques that are commonly used, namely the Lanczos approach, polynomial approximation methods, and rational approximation methods. A practical disadvantage of the Lanczos method is that it requires to store the Lanczos vectors and to orthogonalize or re-orthogonalize these vectors. Polynomial approximations are more economical in terms of storage. In addition, the convergence of polynomial approximation methods does not depend of the matrix. As a result it is easy to estimate the error (these are a-posteriori error estimates for a given degree of the polynomial, not the error bounds of the kind we present in this paper) by only analyzing the function. For example, all that is needed to get the error for the exponential function is to have an idea of the error made in approximating the exponential by the given polynomial in an interval containing the spectrum of A. Such a-posteriori error estimates do not require any computations with the matrix A. This is in contrast with the Lanczos approach for which such errors are generally not as straightforward. There are no known good extensions of the a-posteriori error estimates given in [37] for the Lanczos approach to more general functions than the exponential. One known disadvantage of

polynomial approximation methods is that we must provide an interval that is known to contain the whole spectrum. There is no such requirement for the Lanczos approach. In Section 4, we show that the convergence rate obtained by the Lanczos method will be better than those reached by any polynomial (Chebyshev or Taylor series) approximations. Rational approximation, for example [23], converges the fastest, probably much faster than the Lanczos and the polynomial approximations. However, a major disadvantage of this approach is that we need to solve a number of shifted linear systems. This will be expensive in general, and prohobitive in many realistic cases.

The polynomial approximation methods mentioned earlier use three different strategies to obtain a good polynomial: Taylor series expansions [44], Chebyshev expansions [24], and least squares approximations [12]. The Taylor series approach converges too slowly and is usually not appealing. Chebyshev is a good choice in many scenarios but if the function to approximate has a steep derivative, then the expansion may converge slowly. In the extreme case, if there is a discontinuity (example the sign/step function), then Chebyshev expansions exhibit oscillations and this is known as the Gibbs phenomenon. The least squares approach addresses this issue by first approximating the function with steep derivatives by using a spline, where more knots are placed around the problematic areas, and then in turn approximating the spline by a least squares polynomial. However, we show that the Lanczos method converges faster than any polynomial methods. Section 5 also illustrates the superior performance of the Lanczos method compared to the methods presented in [25, 44] via several numerical experiments.

The outline of the paper is as follows: section 2 is a discussion of the various applications that lead to estimating the trace of matrix functions. Section 3 describes the Stochastic Lanczos Quadrature method in detail. A modified approach of the SLQ method that is more suitable for the Schatten norm estimation is also given. This alternate approach is also appropriate when the input matrix has a large number of singular values close to zero or is non-symmetric. Section 4 lays out the theoretical analysis for the SLQ method. The analysis is applicable for any function that is analytic inside a closed interval (analytically continuable to an open Bernstein ellipse). We establish the approximation error bounds for the computation of different matrix function traces mentioned in sec. 2 using the SLQ method. Section 5 presents numerical experiments.

2. Applications. This section is a brief survey of applications that require the computation of the trace of matrix functions. Such calculations arise in different ways in many disciplines and what follows is just a small set of representative applications. Much more information can be obtained from following the cited references.

2.1. Log-determinant. As previously mentioned, the computation of log-determinant has numerous applications in machine learning and other related fields. The logarithm of the determinant of a given positive definite matrix $A \in \mathbb{R}^{n \times n}$, is equal to the trace of the logarithm of the matrix, i.e.,

$$\log \det(A) = \mathtt{tr}(\log(A)) = \sum_{i=1}^{n} \log(\lambda_i).$$

So, estimating the log-determinant of a matrix is equivalent to estimating the trace of the matrix function $f(A) = \log(A)$. Thus, we can invoke SLQ to estimate this trace.

Suppose the positive definite matrix A has its eigenvalues inside the interval $[\lambda_{\min}, \lambda_{\max}]$, then the logarithm function $f(t) = \log(t)$ is analytic over this interval. This helps us extend our theoretical analysis for logdet computation. When computing the log-determinant of a matrix, the case $\lambda_{\min} = 0$ is obviously excluded, where the function has its singularity. The Lanczos algorithm requires the input matrix to be symmetric. If A is non-symmetric, we can consider either the matrix[1] $A^\top A$, or use the Golub-Kahan-bidiagonalization algorithm described later.

2.2. Log-likelihood. The problem of computing the likelihood function occurs in applications related to Gaussian processes [36, 34]. Maximum Likelihood Estimation (MLE) is a popular approach used for

[1]The matrix product need not be formed explicitly since the Lanczos algorithm requires only matrix vector products.

parameter estimation when high dimensional Gaussian models are used, especially in computational and spatial statistics. The objective in parameter estimation is to maximize the log-likelihood function with respect to a hyperparameter vector ξ:

$$\log p(z \mid \xi) = -\frac{1}{2} z^\top S(\xi)^{-1} z - \frac{1}{2} \log\det S(\xi) - \frac{n}{2}\log(2\pi), \tag{2.1}$$

where z is the data vector and $S(\xi)$ is the covariance matrix parameterized by ξ. The second term (log-determinant) in (2.1) can be computed by using the SLQ method. We observe that the first term in (2.1) resembles the quadratic form that appears in the trace estimator, and it can be also computed by using the Lanczos Quadrature method. That is, we can estimate the term $z^\top S(\xi)^{-1} z$ using m steps of the Lanczos algorithm applied to $z/\|z\|$ as the starting vector, then compute the quadrature rule for the inverse function $f(t) = t^{-1}$, and rescale the result by $\|z\|^2$. In section 4, we give further details on this and present the error bounds for the log-likelihood function estimation by the SLQ method and also describe the choice of m.

2.3. Computing the Schatten p-norms. Another important problem that arises in applications is the estimation of the nuclear norm and the Schatten p-norms of large matrices (a few applications were mentioned earlier). Given an input matrix $X \in \mathbb{R}^{d\times n}$, the nuclear norm of X is defined as $\|X\|_* = \sum_{i=1}^{r} \sigma_i$, where σ_i are the singular values of X and r is its rank. Suppose we define a positive semidefinite matrix A as[1] $A = X^\top X$ or $A = XX^\top$. Then, the nuclear norm of X can also be expressed as

$$\|X\|_* = \sum_{i=1}^{r} \sigma_i = \sum_{i=1}^{r} \sqrt{\lambda_i},$$

where the λ_i's are the eigenvalues of A. Hence, we can consider the symmetric positive semidefinite matrix $A = X^\top X$, and compute the nuclear norm of X as

$$\|X\|_* = \mathtt{tr}(f(A));\ f(t) = \sqrt{t}.$$

To estimate the above trace, we can invoke the SLQ method described in this work. The Schatten p-norm of a general matrix X is defined as

$$\|X\|_p = \Big(\sum_{i=1}^{r} \sigma_i^p\Big)^{1/p} = \Big(\sum_{i=1}^{r} \lambda_i^{p/2}\Big)^{1/p}.$$

Hence, Schatten p-norms (the nuclear norm being a special case with $p = 1$) are the traces of matrix functions of A with $f(t) = t^{p/2}$, and these can also be computed inexpensively using the SLQ method. The functions $f(t) = t^{p/2}$ have singularity at zero. Input matrices whose Schatten norms we seek are likely to have singular values are equal or close to zero (low rank or numerically low rank). However, we show in section 4.5 that such input matrices can be easily handled with a simple modification before applying SLQ.

2.4. Trace of a matrix inverse and the Estrada index. Other frequent matrix function trace estimation problems include estimating the trace of matrix inverse and the Estrada index. As the name indicates, the matrix inverse trace estimation problem amounts to computing the trace of the inverse function $f(t) = \frac{1}{t}$ of a positive definite matrix $A \in \mathbb{R}^{n\times n}$, whose eigenvalues lie in the interval $[\lambda_{\min}, \lambda_{\max}]$ with $\lambda_{\min} > 0$.

Estimation of the Estrada index of graphs is popular in computational biology. This problem accounts to estimating the trace of the exponential function, i.e., $f(t) = \exp(t)$. Both inverse and exponential functions are analytic in the appropriate intervals of interest. Meaning, we can extend the analysis presented in this paper to obtain approximation error bounds, when the SLQ method is used for their computation.

2.5. Other applications. The stochastic Lanczos quadrature method has been employed in the literature for a few related trace estimation problems before. One of the methods proposed by Ubaru et. al [41] for estimating the numerical rank of large matrices (called the Lanczos approximation approach there) is equivalent to the SLQ method discussed here. The function f for this numerical rank estimation problem turns out to be a step function with a value of one above an appropriately chosen threshold. That is, the numerical rank of a matrix is the trace of an appropriate step function of the matrix. The article also proposes an approach to choose this threshold based on the spectral density of the matrix.

An interesting related problem, which is mentioned in [24] is testing the positive definiteness of a matrix. This problem is also equivalent to estimating the trace of a step function of the matrix, with a value of one in a different interval. However, note that the step function has a discontinuity at the point of inflexion (the point where it goes from zero to one) and hence we cannot apply the analysis developed in this paper directly. Also, the degree or the number of Lancozs steps required might be very high in practice. A workaround this issue, that is proposed in [24] (is also mentioned in [41]) is to first approximate the step function by a shifted and scaled hyperbolic tangent function of the form $\tilde{f}(t) = \frac{1}{2}(1 + \tanh(\alpha t))$, where α is an appropriately chosen constant, and then approximate the trace of this surrogate function $\tilde{f}(t)$.

Another problem where SLQ was previously used was in approximating the spectral density of a matrix [31]. The spectral density , also known as Density of States (DOS) of a matrix is a probability density distribution that measures the likelihood of finding eigenvalues of the matrix at a given point on the real line. Being a distribution, the spectral density of a matrix can we written as a sum of delta functions of the eigenvalues of the matrix. That is, the spectral density is defined as

$$\phi(t) = \frac{1}{n} \sum_{i=1}^{n} \delta(t - \lambda_i),$$

where δ is the Dirac distribution or Dirac δ-function. Lin et. al [31] demonstrated how the Lanczos algorithm can be used to approximately estimate the spectral density (equivalent to the SLQ method). The idea is to replace the delta functions by a surrogate Gaussian blurring functions. Then, the spectral density is approximated by estimating the trace of this blurring function using the Lanczos algorithm.

3. Stochastic Lanczos Quadrature. The Lanczos Quadrature method was developed by Gene Golub and his collaborators in a series of articles [20, 5, 6, 19]. The idea of combining the stochastic trace estimator with the Lanczos Quadrature method appeared in [5, 6] for estimating the trace of the inverse and the determinant of matrices. Given a symmetric positive definite matrix[2] $A \in \mathbb{R}^{n \times n}$, we wish to compute the trace of the matrix function $f(A)$, i.e., the expression given by (1.1), where we assume that the function f is is analytic inside a closed interval containing the spectrum of A. To estimate the trace, we invoke the stochastic trace estimator [27], which is a Monte Carlo type method that uses only matrix vector products. The attraction of this method is that it is very inexpensive compared to the methods based on the computing all the eigenvalue of the matrix. The method estimates the trace $\mathtt{tr}(f(A))$ by generating random vectors $u_l, l = 1, \ldots, \mathrm{n_v}$, with Rademacher distribution, form unit vectors $v_l = u_l / \|u_l\|_2$, and then computing the average over the samples $v_l^\top f(A) v_l$,

$$\mathtt{tr}(f(A)) \approx \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} v_l^\top f(A) v_l. \tag{3.1}$$

Hutchinson originally proposed to use vectors with ± 1 entries of equal probability. It has since been shown that vectors from any other random distributions of zero mean and unit 2-norm also work [7, 4]. Strictly speaking, the prior results [4, 35] on which our bounds in section 4 are based, compute the approximation

[2]This matrix may be the sample covariance matrix of the input data matrix X, or may also be the form $X^\top X$ or $XX^\top$ for the given general rectangular matrix X.

as $\|u_l\|_2^2 * v_l^\top f(A) v_l$, where $v_l = u_l/\|u_l\|_2$, rather than $n * v_l^\top f(A) v_l$. However, for Rademacher vectors, $\|u_l\|_2^2 = n$. For any other random vectors, in expectation the two approaches are the same, as long as $\mathbb{E}[v_l v_l^\top] = I$. Hence, for computing the trace we only need to estimate the scalars of the form $v^\top f(A) v$, and the explicit computation of $f(A)$ is never needed.

The scalar (quadratic form) quantities $v^\top f(A) v$ are computed by transforming them to a Riemann-Stieltjes integral problem, and then employing the Gauss quadrature rule to approximate this integral. Consider the eigen-decomposition of A as $A = Q \Lambda Q^\top$. Then, we can write the scalar product as,

$$v^\top f(A) v = v^\top Q f(\Lambda) Q^\top v = \sum_{i=1}^{n} f(\lambda_i) \mu_i^2, \tag{3.2}$$

where μ_i are the components of the vector $Q^\top v$. The above sum can be considered as a Riemann-Stieltjes integral given by,

$$I = v^\top f(A) v = \sum_{i=1}^{n} f(\lambda_i) \mu_i^2 = \int_a^b f(t) d\mu(t), \tag{3.3}$$

where the measure $\mu(t)$ is a piecewise constant function defined as

$$\mu(t) = \begin{cases} 0, & \text{if } t < a = \lambda_n, \\ \sum_{j=1}^{i} \mu_j^2, & \text{if } \lambda_i \le t < \lambda_{i-1}, \\ \sum_{j=1}^{n} \mu_j^2, & \text{if } b = \lambda_1 \le t. \end{cases} \tag{3.4}$$

Next, the integral can be estimated using the Gauss quadrature rule [22] given by

$$\int_a^b f(t) d\mu(t) \approx \sum_{k=0}^{m} \omega_k f(\theta_k), \tag{3.5}$$

where $\{\omega_k\}$ are the weights and $\{\theta_k\}$ are the nodes of the $(m+1)$-point Gauss quadrature rule, which are unknowns and need to be determined.

An elegant way to compute the nodes and the weights of the quadrature rule is to use the Lanczos algorithm [19]. For a given real symmetric matrix $A \in \mathbb{R}^{n \times n}$ and a starting vector w_0 of unit 2-norm, the Lanczos algorithm generates an orthonormal basis W_{m+1} for the *Krylov subspace* Span$\{w_0, Aw_0, \ldots, A^m w_0\}$ such that $W_{m+1}^\top A W_{m+1} = T_{m+1}$, where T_{m+1} is an $(m+1) \times (m+1)$ tridiagonal matrix. For details see [21]. The columns w_k of W_{m+1} are related as

$$w_k = p_{k-1}(A) w_0, \; k = 1, \ldots, m,$$

where p_k are the Lanczos polynomials. The vectors w_k are orthonormal, and we can show that the Lanczos polynomials are orthogonal with respect to the measure $\mu(t)$ in (3.4); see Theorem 4.2 in [19]. Therefore, the nodes and the weights of the quadrature rule in (3.5) can be computed as the eigenvalues and the squares of the first entries of the eigenvectors of T_{m+1}. We can approximate the quadratic form (3.2) as,

$$v^\top f(A) v \approx \sum_{k=0}^{m} \tau_k^2 f(\theta_k) \quad \text{with} \quad \tau_k^2 = \left[e_1^\top y_k \right]^2, \tag{3.6}$$

where $(\theta_k, y_k), k = 0, 1, ..., m$ are eigenpairs of T_{m+1} by using v as the starting vector w_0. Thus, the trace of matrix function $f(A)$ can be computed as,

$$\mathtt{tr}(f(A)) \approx \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} \left(\sum_{k=0}^{m} (\tau_k^{(l)})^2 f(\theta_k^{(l)}) \right), \tag{3.7}$$

where $(\theta_k^{(l)}, \tau_k^{(l)}), k = 0, 2, ..., m$ are eigenvalues and the first entries of the eigenvectors of the tridiagonal matrix $T_{m+1}^{(l)}$ corresponding to the starting vectors $v_l, l = 1, \ldots, \mathrm{n_v}$. This method is far less costly than computing the eigenvalues of the matrix A for the purpose of computing the trace via (1.1). The Stochastic Lanczos Quadrature algorithm corresponding to this procedure is sketched in Algorithm 1.

Algorithm 1 Trace of a matrix function by SLQ using the Lanczos algorithm

Input: A symmetric PSD matrix $A \in \mathbb{R}^{n \times n}$, the analytic function f, degree m and $\mathrm{n_v}$.
Output: The approximate trace Γ of $f(A)$.
for $l = 1$ to $\mathrm{n_v}$ **do**
1. Generate a Rademacher random vector u_l, form unit vector $v_l = u_l / \|u_l\|_2$.
2. $T = \text{Lanczos}(A, v_l, m + 1)$;
Apply $m + 1$ steps of Lanczos to A with v_l as the starting vector.
3. $[Y, \Theta] = \texttt{eig}(T)$; and compute $\tau_k = [e_1^\top y_k]$ for $k = 0, \ldots, m$
4. $\Gamma \leftarrow \Gamma + \sum_{k=0}^{m} \tau_k^2 f(\theta_k)$.
end for
Output $\Gamma = \frac{n}{\mathrm{n_v}} \Gamma$.

In section 4, we establish strong theoretical error bounds for this approach for functions analytic inside a closed interval. We show that the convergence rate of Quadrature methods is faster than other polynomial expansion methods, e.g., Chebyshev polynomials.

Golub Kahan Bidiagonalization. When the input matrix X has a large number of singular values close to zero, the Lanczos algorithm might encounter some numerical issues. In such scenarios, it is advantageous to use the Golub Kahan Bidiagonalization (G-K-B) algorithm [18] in place of the Lanczos algorithm. For details on the Lanczos and the G-K-B algorithms, see, e.g., [19]. Suppose B_{m+1} is the bidiagonal matrix obtained by the G-K-B algorithm, then the matrix $T_{m+1} = B_{m+1}^\top B_{m+1}$ will be the Lanczos Jacobi matrix corresponding to $X^\top X$ [19]. The singular values ϕ_k of B_{m+1} are such that $\phi_k = \sqrt{\theta_k}$, for $k = 0, \ldots, m$, where θ_k are the eigenvalues of T_{m+1}. Thus, the Shatten p- norms (the square root function and its powers) can be computed using the singular values of the bidiagonal matrix B_{m+1} obtained from m steps of the G-K-B algorithm. Similarly, traces of functions of non-Hermitian matrices can also be computed using this algorithm. Algorithm 2 presents a version of the Stochastic Lanczos Quadrature method that uses the G-K-B bidiagonalization.

Algorithm 2 Trace of a matrix function by SLQ using the G-K-B algorithm

Input: $X \in \mathbb{R}^{d \times n}$, analytic function f (with $A = X^\top X$, $\tilde{f} : \tilde{f}(t) = f(t^2)$), m and $\mathrm{n_v}$.
Output: The approximate trace Γ of $f(A)$.
for $l = 1$ to $\mathrm{n_v}$ **do**
1. Generate a Rademacher random vector v_l, such that $\|v_l\|_2 = 1$.
2. $B = \text{GKB}(X, v_l, m + 1)$;
Apply $m + 1$ steps of GKB to X with v_l as the starting vector.
3. $[U, \Phi] = \texttt{svd}(B)$; and compute $\tau_k = [e_1^\top u_k]$ for $k = 0, \ldots, m$
4. $\Gamma \leftarrow \Gamma + \sum_{k=0}^{m} \tau_k^2 \tilde{f}(\phi_k)$.
end for
Output $\Gamma = \frac{n}{\mathrm{n_v}} \Gamma$.

Computational Cost. Since we apply m steps of the Lanczos or the G-K-B algorithm for $\mathrm{n_v}$ different starting vectors, the cost of the Stochastic Lanczos Quadrature method will be $O((\mathrm{nnz}(A)m + nm^2)\mathrm{n_v})$, where $\mathrm{nnz}(A)$ is the number of nonzeros in A. Typically both m and $\mathrm{n_v}$ are much smaller than input dimension n. Hence, the method will be very inexpensive for large sparse matrices.

4. Analysis. In this section, we present the error bounds for approximating the trace of a matrix function using SLQ. Additive error bounds are also established for the log-determinant approximation of a positive definite matrix and the log-likelihood function estimation. The nuclear norm and Schatten-p norms estimation of a general matrix is discussed in the latter part of the section.

THEOREM 4.1. *Consider a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$ and condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$, and let f be a function that is analytic and either positive or negative (does not cross zero) inside this interval, and whose absolute maximum and minimum values in the interval are M_ρ and m_ρ respectively. Let ε, η be constants in the interval $(0, 1)$. Then for SLQ parameters satisfying:*

- $m \geq \frac{\sqrt{\kappa}}{4} \log \frac{K}{\varepsilon}$ *number of Lanczos steps, and*
- $\mathrm{n_v} \geq \frac{24}{\varepsilon^2} \log(2/\eta)$ *number of starting Rademacher vectors,*

where $K = \frac{(\lambda_{\max}-\lambda_{\min})(\sqrt{\kappa}-1)^2 M_\rho}{\sqrt{\kappa} m_\rho}$, the output Γ of the Stochastic Lanczos Quadrature method is such that:

$$\Pr\Big[|\mathtt{tr}(f(A)) - \Gamma| \leq \varepsilon \, |\mathtt{tr}(f(A))| \Big] \geq 1 - \eta. \tag{4.1}$$

To prove the theorem, we first derive error bounds for the Lanczos Quadrature approximation (which gives the convergence rate), using the facts that an m-point Gauss Quadrature rule is exact for any $2m-1$ degree polynomial and that the function is analytic inside an interval (and is analytically continuable tom a Bernstein ellipse). We then combine this bound with the error bounds for the stochastic trace estimator to obtain the above result.

4.1. Convergence rate for the Lanczos Quadrature. In order to prove Theorem 4.1, we first establish the convergence rate for the Lanczos Quadrature approximation of the quadratic form. Recall that the quadratic form $v^\top f(A) v$ can be written as a Riemann Stieltjes integral I, as given in (3.6). Let I_m denote the $(m+1)$-point Gauss Quadrature rule that approximates the integral I, given by

$$I_m = \sum_{k=0}^{m} \omega_k f(\theta_k),$$

where $\{\omega_k\}$ are the weights and $\{\theta_k\}$ are the nodes, computed by using $m+1$ steps of the Lanczos algorithm. The well known error analysis for the Gauss Quadrature rule is given by [19],

$$|I - I_m| = \frac{f^{2m}(\zeta)}{2m!} \int_a^b \Big[\prod_{k=0}^{m} (t - \theta_k) \Big]^2 d\mu(t), \tag{4.2}$$

for some $a < \zeta < b$. However, this analysis might not be useful for our purpose, since the higher derivatives of both the logarithm and the square root function become excessively large in the interval of interest. Hence, in this work, we establish (novel) improved error analysis for the Lanczos Quadrature approximations, using some classical results developed in the literature, with fact that functions of interest are analytic over a certain interval. The following theorem presents this result:

THEOREM 4.2. *Let a function g be analytic in $[-1, 1]$ and analytically continuable in the Bernstein ellipse E_ρ with foci ± 1 and sum of major and minor axis equal to $\rho > 1$, where it satisfies $|g(z)| \leq M_\rho$. Then the $(m+1)$-step Lanczos Quadrature approximation satisfies*

$$|I - I_m| \leq \frac{4 M_\rho}{(\rho^2 - 1)\rho^{2m}}. \tag{4.3}$$

Proof. In order to prove the above theorem, we follow the arguments developed in [33] to estimate the error in Gaussian integration of analytic functions. In [33], Rabinowitz considers an arbitrary integration rule (Gauss Quadrature rule being an example) to approximate the general integration of analytic functions over

the interval $[-1, 1]$. This result is generally covered in standard textbooks, e.g., [39, Thm. 19.3]. In our case, I is a Riemann Stieltjes integral with a specific measure given in (3.4)

For the given function g that is analytic over the interval $[-1, 1]$, let us consider the $2m$ degree Chebyshev polynomial approximation of $g(t)$, i.e.,

$$P_{2m} = \sum_{j=0}^{2m} a_j T_j(t) \approx g(t).$$

We know that the $m + 1$ point Gauss Quadrature rule is exact for any polynomial of degree upto $2m + 1$, see [19, Thm. 6.3] or [39, Thm. 19.1]. This can also be deduced from the error term in (4.2). Hence, the error in integrating g is the same as the error in integrating $g - P_{2m}$. Thus, we have

$$\begin{aligned}
|I - I_m| &= |I(g - P_{2m}) - I_m(g - P_{2m})| \leq |I(g - P_{2m})| + |I_m(g - P_{2m})| \\
&= \left| I\Big(\sum_{j=2m+1}^{\infty} a_j T_j(t) \Big) \right| + \left| I_m\Big(\sum_{j=2m+1}^{\infty} a_j T_j(t) \Big) \right| \\
&\leq \sum_{j=2m+1}^{\infty} |a_j| \Big[|I(T_j)| + |I_m(T_j)| \Big]
\end{aligned}$$

Next, we obtain bounds for the three terms inside the summation above.

If the function g is analytic in $[-1, 1]$ and analytically continuable in the Bernstein ellipse E_ρ, then for the Chebyshev coefficients we have from Theorem 8.1 in [39] and eq. (14) in [33],

$$|a_j| \leq \frac{2M_\rho}{\rho^j}.$$

Next, for the Quadrature rule $I_m(T_j)$, we have

$$I_m(T_j) = \sum_{k=0}^{m} \tau_k^2 T_j(\theta_k) \leq \sum_{k=0}^{m} |\tau_k^2||T_j(\theta_k)| \leq 1.$$

This inequality results from the fact that, for $f(t) = 1$, the quadrature rule is exact, and the integral is equal 1 ($v_l^\top f(A) v_l = v_l^\top v_l = 1$). Therefore, the weights must sum to 1. The maximum value of T_j inside the interval is 1. Finally, in order to bound the Riemann-Stieltjes integral $I(T_j)$, we use the following:

$$I(T_j) = v^\top T_j(A) v \leq \lambda_{\max}(T_j(A)) = 1,$$

by the min-max theorem and $\|v\| = 1$. Therefore,

$$|I - I_m| \leq \sum_{j=2m+1}^{\infty} \frac{2M_\rho}{\rho^j}[1 + 1].$$

Since the Gauss quadrature rule is a symmetric rule [33], the error in integration of $T_j(t)$ for any odd j will be equal to zero. Thus, we get the result in the theorem

$$|I - I_m| \leq \frac{4M_\rho}{(\rho^2 - 1)\rho^{2m}}.$$

□

REMARK 1. *The convergence rate for the Chebyshev polynomial approximation of an analytic function is $O(1/\rho^m)$, see Theorem 8.2 in [39]. Hence, the Lanczos Quadrature approximation is twice as fast as the*

Chebyshev approximation. Moreover, it is known that the Gauss quadrature has the maximal polynomial order of accuracy [39].

Theorem 4.2 holds for functions that are analytic over $[-1, 1]$ (and analytically continuable in the Bernstein ellipse E_ρ). The functions considered in this paper such as logarithm, inverse, exponential and square root functions are analytic over $[\lambda_{\min}, \lambda_{\max}]$ for $\lambda_{\min} > 0$. Hence, we need to use the following transform in order to get the right interval.

If $f(x)$ is analytic on $[\lambda_{\min}, \lambda_{\max}]$, then $g(t) = f\left[\left(\frac{\lambda_{\max}-\lambda_{\min}}{2}\right)t + \frac{\lambda_{\max}+\lambda_{\min}}{2}\right]$ is analytic on $[-1, 1]$. If we denote the error in the Quadrature rule for approximating the integral of function f as $E(f)$, then we have

$$E(f) = \left(\frac{\lambda_{\max} - \lambda_{\min}}{2}\right) E(g).$$

The function g will have its singularity at $t_0 = \alpha = -\frac{\kappa+1}{\kappa-1}$. Hence, we choose the ellipse E_ρ with the semimajor axis length of $|\alpha|$ where g is analytic on and inside. Then, the convergence rate ρ will be

$$\rho = \alpha \pm \sqrt{\alpha^2 - 1} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} > 1.$$

The sign is chosen such that $\rho > 1$. From theorem 4.2, the error $E(g) \leq \frac{4M_\rho}{(\rho^2-1)\rho^{2m}}$, where $|g(z)| < M_\rho$ inside E_ρ. Hence, the error $E(f)$ will be

$$E(f) = \left(\frac{\lambda_{\max} - \lambda_{\min}}{2}\right) \frac{4M_\rho}{(\rho^2-1)\rho^{2m}} = \frac{(\lambda_{\max} - \lambda_{\min})(\sqrt{\kappa}-1)^2 M_\rho}{2\sqrt{\kappa}\rho^{2m}},$$

with ρ defined as above. Thus, for a function f that is analytic on $[\lambda_{\min}, \lambda_{\max}]$ and $C = \frac{(\lambda_{\max}-\lambda_{\min})(\sqrt{\kappa}-1)^2 M_\rho}{2\sqrt{\kappa}}$, we have

$$\left| v^\top f(A) v - \sum_{k=0}^{m} \tau_k^2 f(\theta_k) \right| \leq \frac{C}{\rho^{2m}}. \tag{4.4}$$

4.2. Approximation error of the trace estimator. The quadratic form $v^\top f(A)v$ for which we derived the error bounds in the previous section comes from the Hutchinson trace estimator. Let us denote this trace estimator as $\mathtt{tr}_{\mathrm{n_v}}(A) = \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} v_l^\top A v_l$. The convergence analysis for the stochastic trace estimator was developed in [4], and improved in [35] for sample vectors with different probability distributions. We state the following theorem which is proved in [35].

THEOREM 4.3. *Let A be an $n \times n$ symmetric positive semidefinite matrix and $v_l, l = 1, \ldots, \mathrm{n_v}$ be random starting vectors with zero mean and unit 2 norm drawn from a distribution D. Then, for D=Huntchinson distribution (Rademacher distribution) with $\mathrm{n_v} \geq 6\log(2/\eta)/\varepsilon^2$, and for D=Gaussian distribution with $\mathrm{n_v} \geq 8\log(2/\eta)/\varepsilon^2$, we have*

$$\Pr\left[|\mathtt{tr}_{\mathrm{n_v}}(A) - \mathtt{tr}(A)| \leq \varepsilon |\mathtt{tr}(A)|\right] \geq 1 - \eta.$$

The above theorem can be used to bound the trace of any matrix function $f(A)$, if the function is either positive or negative inside the spectrum interval. Also recall that, for general distribution D, strictly speaking, the above bound holds for $\|u_l\|^2 * v_l^\top f(A) v_l$, where $v_l = u_l/\|u_l\|$|, rather than $n * v_l^\top f(A) v_l$. Therefore, the theorem holds for the square root function, its powers, exponential and the inverse function. However, for the logarithm function, different scenarios occur depending on the spectrum, which will be discussed later. Let Γ be the output of the Stochastic Lanczos Quadrature method to estimate the trace of such functions, given by

$$\Gamma = \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} \left(\sum_{k=0}^{m} (\tau_k^{(l)})^2 f(\theta_k^{(l)}) \right). \tag{4.5}$$

The following Lemma will be needed.

LEMMA 4.4. *Let $A \in \mathbb{R}^{n\times n}$ be a positive definite matrix with eigenvalue in $[\lambda_{\min}, \lambda_{\max}]$ and condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$, and f is an analytic function in this interval with $|f(z)| \leq M_\rho$. Then, the following inequality holds:*

$$|\mathtt{tr}_{\mathrm{n_v}}(f(A)) - \Gamma| \leq \frac{nC}{\rho^m},$$

where $\rho = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}$ and $C = \frac{(\lambda_{\max}-\lambda_{\min})(\sqrt{\kappa}-1)^2 M_\rho}{2\sqrt{\kappa}}$.

Proof. The lemma follows from the equation (4.4). We have

$$\begin{aligned}
|\mathtt{tr}_{\mathrm{n_v}}(f(A)) - \Gamma| &= \frac{n}{\mathrm{n_v}} \left| \sum_{l=1}^{\mathrm{n_v}} v_l^\top f(A) v_l - \sum_{l=1}^{\mathrm{n_v}} I_m^{(l)} \right| \\
&\leq \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} |v_l^\top f(A) v_l - I_m^{(l)}| \\
&\leq \frac{n}{\mathrm{n_v}} \sum_{l=1}^{\mathrm{n_v}} \frac{C}{\rho^{2m}} = \frac{nC}{\rho^{2m}}
\end{aligned}$$

□

Now, we are ready to prove main theorem. First, we obtain the condition on the number of Lanczos steps m by setting

$$\frac{C}{\rho^{2m}} \leq \frac{\varepsilon}{2} f_{\min}(\lambda) \Rightarrow \frac{K}{\varepsilon} \leq \rho^{2m},$$

where $K = \frac{2C}{f_{\min}(\lambda)}$ and $f_{\min}(\lambda)$ is the absolute minimum of the function in the interval $[\lambda_{\min}, \lambda_{\max}]$. Taking log on both sides we have

$$\log \frac{K}{\varepsilon} \leq 2m \log \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right) \leq \frac{4m}{\sqrt{\kappa}}.$$

Then, from Lemma 4.4 we have

$$|\mathtt{tr}_{\mathrm{n_v}}(f(A)) - \Gamma| \leq \frac{\varepsilon n}{2} f_{\min}(\lambda) \leq \frac{\varepsilon}{2} |\mathtt{tr}(f(A))|. \tag{4.6}$$

From Theorem 4.3, we have

$$\Pr\left[|\mathtt{tr}(f(A)) - \mathtt{tr}_{\mathrm{n_v}}(f(A))| \leq \frac{\varepsilon}{2} |\mathtt{tr}(f(A))| \right] \geq 1 - \eta. \tag{4.7}$$

Combining the above two inequalities (4.6) and (4.7) leads to the result in Theorem 1.

$$\begin{aligned}
1 - \eta &\leq \Pr\left[|\mathtt{tr}(f(A)) - \mathtt{tr}_{\mathrm{n_v}}(f(A))| \leq \frac{\varepsilon}{2} |\mathtt{tr}(f(A))| \right] \\
&\leq \Pr\left[|\mathtt{tr}(f(A)) - \mathtt{tr}_{\mathrm{n_v}}(f(A))| + |\mathtt{tr}_{\mathrm{n_v}}(f(A)) - \Gamma| \leq \frac{\varepsilon}{2} |\mathtt{tr}(f(A))| + \frac{\varepsilon}{2} |\mathtt{tr}(f(A))| \right] \\
&\leq \Pr\left[|\mathtt{tr}(f(A)) - \Gamma| \leq \varepsilon \, |\mathtt{tr}(f(A))| \right].
\end{aligned}$$

Theorem 1 can be used to obtain the bounds for the approximation errors in the estimation of the log-determinants and the nuclear norms of matrices using the Stochastic Lanczos Quadrature method. As discussed earlier, the Lanczos Quadrature approximation converges twice as fast as does the Chebyshev approximation for a given analytic function.

For comparison, note that for the Chebyshev approximations [25], the degree required is $m = \Theta(\sqrt{\kappa}\log\frac{\kappa}{\varepsilon})$ and for Taylor approximations [8], $m = O(\kappa\log\frac{\kappa}{\varepsilon})$. Recall from Remark 1, the Lanczos algorithm is superior to the Chebyshev expansions because the former approximation converges twice as fast as does the latter. Clearly, the Lanczos also converges faster than the Taylor. The above theorem can now be used to establish the error bounds for approximating the log-determinants and the Schatten p-norms. The quality and the complexity of the algorithms depend on the condition number κ, since matrix function approximations become harder when matrices have wider spectra, and we will require higher degree approximations.

4.3. Bounds for Log-determinant. For the logarithm function, we encounter three different scenarios depending on the spectrum of the matrix. The first case is when $\lambda_{\max} < 1$, the log function $\log(A)$ is negative definite and the log-determinant will always be negative. Thus, the conditions of Theorem 4.1 are satisfied. Similarly, Theorem 4.1 holds in the second case when $\lambda_{\min} > 1$, since $\log(A)$ is positive definite. In the third case when $\lambda_{\min} < 1$ and $\lambda_{\max} > 1$, however, we cannot obtain multiplicative error bounds of the form given in Theorem 4.1, since the log function will cross zero inside the interval. In the worst case, the log-determinant can be zero. One simple workaround to avoid this case is to scale the matrix such that its eigenvalues are either all smaller than 1 or all greater than 1; however, such an approach requires the computation of the extreme eigenvalues of A. The following corollary gives additive error bounds without scaling; it holds for any PSD matrix.

COROLLARY 4.5. *Given $\varepsilon, \eta \in (0,1)$, a PSD matrix $A \in \mathbb{R}^{n\times n}$ with its eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$, and condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$, for SLQ parameters:*

- $m \geq \frac{\sqrt{\kappa}}{4}\log\frac{K_1}{\varepsilon}$ *number of Lanczos steps, and*
- $\mathrm{n_v} \geq \frac{24}{\varepsilon^2}(\log(1+\kappa))^2\log(2/\eta)$ *number of starting vectors,*

where $K_1 = \lambda_{\max}\sqrt{\kappa}\log(\lambda_{\max}+\lambda_{\min})$, we have

$$\Pr\left[\,|\log\det(A) - \Gamma| \leq \varepsilon n\right] \geq 1-\eta, \tag{4.8}$$

where Γ is the output of the Stochastic Lanczos Quadrature method for log-determinant computation.

Proof. The proof of the Corollary follows from the above proof of Theorem 4.1. In the logarithm case, Theorem 4.2, and particularly equation (4.4) still holds, and so does lemma 4.4. The K_1 value is obtained by setting $\frac{n\lambda_{\max}\sqrt{\kappa}M_\rho}{2\rho^{2m}} \leq \frac{\varepsilon n}{2}$ and $M_\rho = \log(\lambda_{\max}+\lambda_{\min})$. For theorem 4.3, we consider the fact that, if $B = \frac{A}{\lambda_{\max}+\lambda_{\min}}$, then

$$\log\det A = \log\det B + n\log(\lambda_{\max}+\lambda_{\min}).$$

Since the matrix B has its eigenvalues inside $(0,1)$, the logarithm function is negative and we hence can apply Theorem 4.3 with $f(A) = \log\left(\frac{A}{\lambda_{\max}+\lambda_{\min}}\right)$, and then add and subtract $n\log(\lambda_{\max}+\lambda_{\min})$ to get an equation of the form (4.7). We can then prove the corollary using $|\log\det B| \leq n\log(1+\kappa)$. ∎

4.4. Bounds for Log-likelihood function. Recall the log-likelihood function defined in (2.1). The log-determinant term in it can be bounded as above. The first term $z^\top S(\xi)^{-1}z$ is computed using the Lanczos quadrature method with $z/\|z\|$ as the starting vector for the Lanczos algorithm. The following corollary gives the error bounds for the log-likelihood function estimation by SLQ, which follows from Theorem 4.1 and Corollary 4.5.

COROLLARY 4.6. *Given a data vector $z \in \mathbb{R}^n$, a covariance matrix $S(\xi) \in \mathbb{R}^{n\times n}$ with hyperparameter ξ and its eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$, and constants $\varepsilon, \eta \in (0,1)$, for SLQ parameters:*

- $m_1 \geq \frac{\sqrt{\kappa}}{4}\log\frac{K_1}{\varepsilon}$, $m_2 \geq \frac{\sqrt{\kappa}}{4}\log\frac{K_2}{\varepsilon}$ *and*
- $\mathrm{n_v} \geq \frac{24}{\varepsilon^2}(\log(1+\kappa))^2\log(2/\eta)$,

where K_1 is defined in Corollary 4.5 and $K_2 = \kappa^{3/2}\|z\|^2$, we have

$$\Pr\left[\,|\log p(z \mid \xi) - \Gamma| \leq \varepsilon(n+1)\right] \geq 1-\eta, \tag{4.9}$$

where $\Gamma = -\Gamma_1 - \Gamma_2 - \frac{n}{2}\log(2\pi)$, Γ_1 *is the output of SLQ with parameters* m_1 *and* $\mathrm{n_v}$*, and* Γ_2 *is the output of the Lanczos Quadrature method for approximating* $z^\top S(\xi)^{-1} z$ *with* m_2 *steps of Lanczos.* For Corollary 2, we bound the log-determinant term Γ_1 obtained by SLQ using Corollary 1, and Γ_2 the Lanczos Quadrature of $z^\top S(\xi)^{-1} z$ is bounded using theorem 4.2 and eq. (4.4) ($M_\rho = 1/\lambda_{\min}$) with scaling $\|z\|^2$.

4.5. Schatten p-norms estimation. When estimating the nuclear and Schatten p-norms, we encounter the following issue when approximating the square root function. *In order to obtain strong theoretical results (exponential convergence) for a given function* $f(t)$*, the function must be analytic in the interval defined (in our case,* t *will be defined over* $[0, 1]$*). However, the square root function is non-differentiable at* $t = 0$. This will be a major stumbling block for rank-deficient matrices since the interval of eigenvalues now contains zero.

Shifting the Spectrum. To overcome the issue mentioned above, we propose the following remedy, which is based on the key observation, proper to the computation of nuclear norm, that the small and zero singular values do not contribute much to the norm itself. In other words, the nuclear norm of a matrix depends mainly on the top singular values.

The idea is then to shift the spectrum of the matrix by a small $\delta > 0$ such that no eigenvalues of the matrix A are equal to zero. That is, we replace A by $A + \delta I$, such that the eigenvalues of the new shifted matrix will be $\lambda_i + \delta$. For the square root function, the error is given by,

$$\sqrt{\lambda_i + \delta} - \sqrt{\lambda_i} = \frac{\delta}{\sqrt{\lambda_i + \delta} + \sqrt{\lambda_i}}.$$

Hence, the error in the large eigenvalues will be small. The error in the nuclear norm will be

$$\sum_{i=1}^{n} \sqrt{\lambda_i + \delta} - \sum_{i=1}^{n} \sqrt{\lambda_i} = \sum_{i=1}^{n} \frac{\delta}{\sqrt{\lambda_i + \delta} + \sqrt{\lambda_i}}. \tag{4.10}$$

For the shifted matrix, the eigenvalues will be in the interval $[\delta, \lambda_{\max} + \delta]$. Now, theorem 4.1 holds in this interval (square root function will be positive) and we can obtain the approximation error bounds. The error due to shifting will be small, and can also be corrected using the Taylor series expansion of the square root function (details omitted). Algorithm 2 will be better suited for nuclear norm estimation.

Bounds for Schatten p-norms. For Schatten p-norms, if the input matrix has full rank ($\lambda_{\min} > 0$), then Theorem 4.1 is directly applicable, since the square root function is positive and analytic in the interval. For rank deficient matrices (has zero singular values), we will encounter the above problem, and we need to shift the spectrum by δ. From (4.10), we can upper bound the error due to shifting by $n\sqrt{\delta}$. Thus, the shift δ is chosen such that this error due to shifting is at most $\varepsilon \|X\|_p^p$. Here, the value of $\|X\|_p$ can be taken to be roughly $\mathrm{poly}(n)$. We can then compute $\|X\|_p$ of the shifted matrix using SLQ. Theorem 4.1 will give us the error bounds.

Similarly, we can obtain the error bounds for estimating the trace of matrix inverses and the Estrada indices of matrices using SLQ from Theorem 4.1 with appropriate substitutions. These substitutions will be similar to those used in [24] for obtaining bounds for the Chebyshev approximations of these functions.

5. Numerical Experiments. In this section, we present several examples to illustrate the performance of the SLQ method in various applications. First, we evaluate its performance for log-determinant computation of large matrices, and compare the performance against other related stochastic methods.

In the first experiment (Figure 5.1(a)), we compare the relative errors obtained by the SLQ method for different degrees chosen, and compare it against the stochastic Chebyshev [25] (as implemented by the authors) and the stochastic Taylor series expansions method [44]. We consider the sparse matrix `california` (a graph Laplacian matrix) of size 9664×9664, $\mathrm{nnz} \approx 10^5$ and $\kappa \approx 5 \times 10^4$ from the University of Florida (UFL) sparse matrix collection [14]. The number of starting vectors $\mathrm{n_v} = 100$ in all three cases. The figure shows that our method is superior in accuracy compared to the other two methods. With just a degree of around 50, we get 4 digits of accuracy, while Chebyshev expansions give only 1-2 digits of accuracy and Taylor series

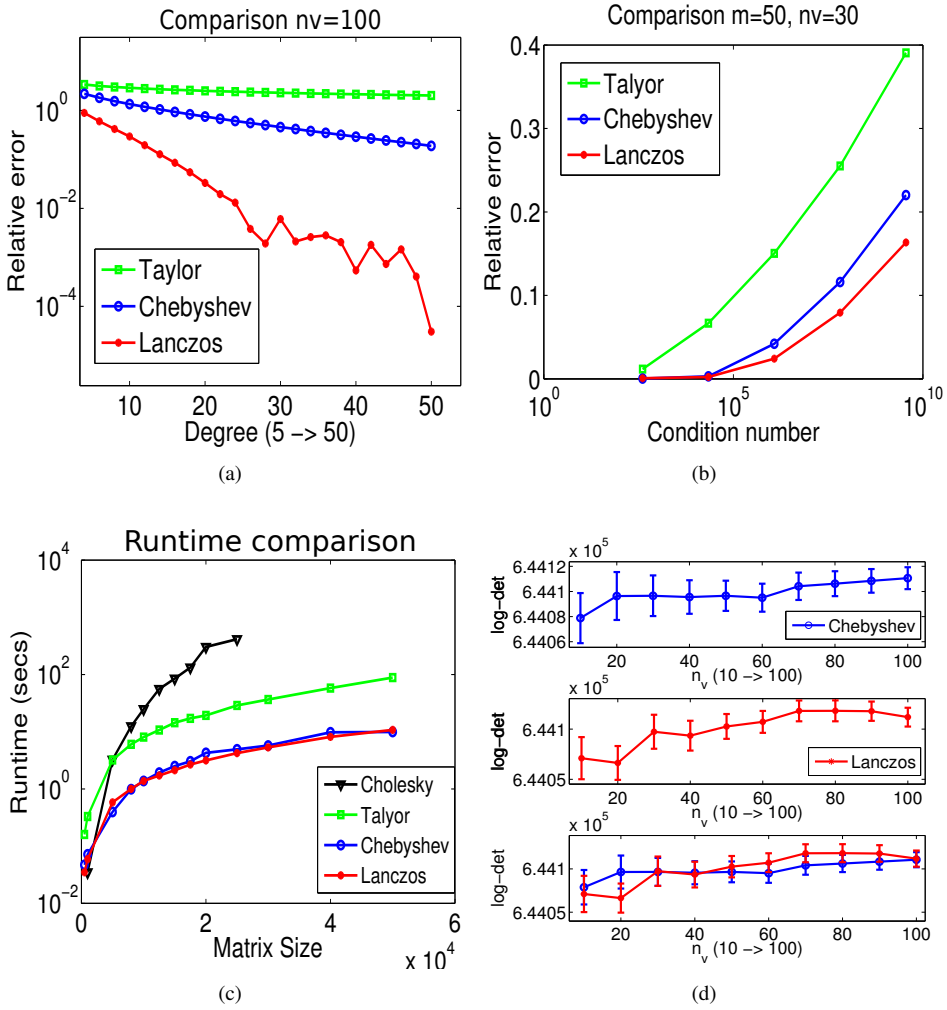

FIG. 5.1. *Performance comparison between SLQ, Chebyshev and Taylor series expansions: (a) Relative error vs. degree m, (b) Relative error vs. condition number of the matrices, (c) runtime comparison against Cholesky decomposition and (d) estimation and standard error vs. number of starting vectors.*

expansions are very inaccurate for such low degrees. In the second experiment, we evaluate the performance of our method with respect to the condition number of the matrix. We consider a Hadamard[3] matrix H of size 8192 and form the test matrix as $HDH^\top$, where D is a diagonal matrix with entries such that the desired condition number is obtained. Figure 5.1(b) plots the relative errors obtained by the three stochastic methods for the log-determinant estimations of the matrices with different condition numbers. The degree and the number of starting vectors used in all three cases were $m = 50$ and $\mathrm{n_v} = 30$. Again, we observe the superior accuracy of SLQ.

In the third experiment, we compare the runtime of the three algorithms for log-determinant estimation of large sparse matrices. The matrices have used 10% nonzeros in each row. (For example, $N = 20000; rho = 10/N; A = sprand(N, N, rho); A = A' * A + lmin * speye(N);$) These are the same matrices used in Fig. 1 of [24]. We also include the runtime for the Cholesky decomposition. For a fair comparison, we chose

[3] A Hadamard matrix is chosen since its eigenvalues are known apriori and is easy to generate. Reproducing the experiment will be easier.

TABLE 5.1
Log-determinant computation of real datasets from UFL matrix collection with $\mathrm{n_v} = 30$.

Matrices (Applications)	Size	Exact logdet	Chebyshev Expansions			Lanczos Quadrature		
			m	Estimate	time	m	Estimate	time
California (web search)	9664	-35163	150	-31657.9	1.02	55	-35112.3	1.55
qpband (optimization)	20000	5521	70	5480.1	0.95	30	5517.0	0.28
thermomechTC (Thermal)	102158	-546787	75	-546640.3	7.76	25	-546793.9	7.34
boneS01 (Model reduction)	127224	$1.1093e6$	150	$4.119e6$	26.15	35	$1.104e6$	17.59
ecology2 (2D/3D)	999999	$3.3943e6$	60	$3.3946e6$	70.8	30	$3.3949e6$	75.24

$m = \sqrt{\kappa}$ for the Chebyshev method, $m = \sqrt{\kappa}/2$ for SLQ and $m = 4\sqrt{\kappa}$ for Taylor series (will be less accurate since we need $m \approx \kappa$ for similar accuracy). Figure 5.1(c) plots the runtime of the four algorithms for different matrix sizes. We observe that the runtime of the SLQ method is equal or less than the runtime of the Chebyshev method. Note also that, both Chebyshev and Taylor methods require computation of the extreme eigenvalues. The relative errors we obtained by SLQ in practice are also lower than that obtained by the Chebyshev method. These two methods are both significantly faster than the one based on Cholesky. All experiments were conducted using `Matlab` on an Intel core i-5 3.3 GHz machine. All timings are reported using `cputime` function. Comparisons with Schur complement methods and rational approximations can be seen in Fig. 1 of [24], where it is shown that the Chebyshev method is superior to these two methods. Hence, we compare SLQ with only the Chebyshev method in the following experiments.

For very large matrices ($\sim 10^6$ and above), it is impractical to compute the exact log-determinants. To gauge the approximation quality, we approximate the estimator variance by using sample variance and show the standard errors. Figure 5.1(d) plots the log-determinants estimated and the error bars obtained for different number of starting vectors for the matrix `webbase-1M` (Web connectivity matrix) of size $10^6 \times 10^6$ obtained from the UFL database [14]. For Lanczos Quadrature, we chose degree $m = 30$, and for Chebyshev $m = 60$. The width of the error bars gives us a rough idea of how close we might be to the exact trace.

Table 5.1 gives some additional comparison results between Chebyshev expansions and SLQ methods on some large real datasets. All matrices were obtained from the University of Florida (UFL) sparse matrix collection [14] and are sparse. Some of these matrices were also used in [8] as test matrices. The matrices, their applications and sizes are listed in the first two columns of the table. The exact log-determinmants of the matrices are listed in the third column. For the first two matrices, their singular values are also available in the UFL database (logdet were computed using them). For the remaining matrices, the exact log-determinants are reported in [8], where the authors used Cholesky decomposition to obtain these values. For the Chebyshev method, we increment the degree m until either we achieve 2-3 digits of accuracy or $m = 150$. For SLQ, we increment the degree m (number of Lanczos steps) until we achieve 3-4 digits of accuracy. The degree used and the log-determinants estimated by these two methods are listed in the table along with the time taken (averaged over 5 trials) by these algorithms. In all experiments, the number of starting vectors $\mathrm{n_v} = 30$. We observe that, in all cases, results obtained by SLQ are way more accurate than the Chebyshev method. Also, SLQ requires at least 2-3 times lower degree m than Chevbyshev method to achieve more accurate results. Also, it is important to note that the Stochastic Chebyshev and Taylor series methods require computation of the largest and the smallest eigenvalues of the matrix.

Maximum Likelihood estimation for GRF. We now test our method for maximum likelihood (ML) estimation of Gaussian Random Fields (GRF). To illustrate the use of log-determinant calculation in GRFs, we simulate one such field by using the Wendland covariance function [34] with smoothness $q = 0$ on a 900×1200 grid ($n = 1.08 \times 10^6$); see fig. 5.2(a). To better demonstrate the fine details of this highly nonsmooth data, we have zoomed into the middle 300×400 grid and shown only this part. Next, we randomly sample ten percent of the data and used them to estimate the length scale of the function. These training data are the non-white pixels in fig. 5.2(b). We compute a local log-likelihood curve (as in (2.1)) shown in fig. 5.2(c) using SLQ with different values for the hyperparameter, which suggests a peak at 50. That is, MLE estimates using SLQ suggests the hyperparameter value to be 50. This coincides with the true value used for simulation. The log-determinants therein were computed using 100 Lanczos steps and 100 Hutchinson vectors. The matrix-vector multiplications were carried out through circulant embedding followed by FFT,

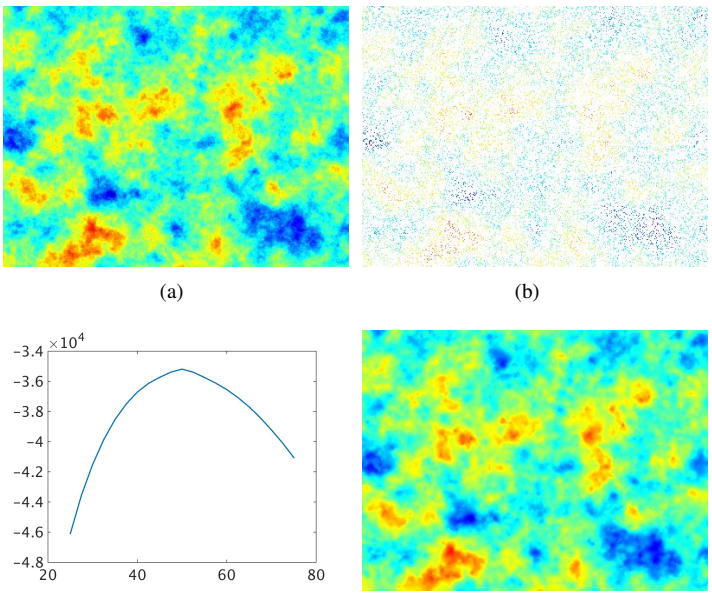

FIG. 5.2. *Estimation and prediction for a Gaussian random field. (a) The random field. (b) Training data (non-white pixels) for parameter estimation. (c) Log-likelihood; the horizontal axis denotes the length-scale parameter. (d) Prediction by using the estimated parameter.*

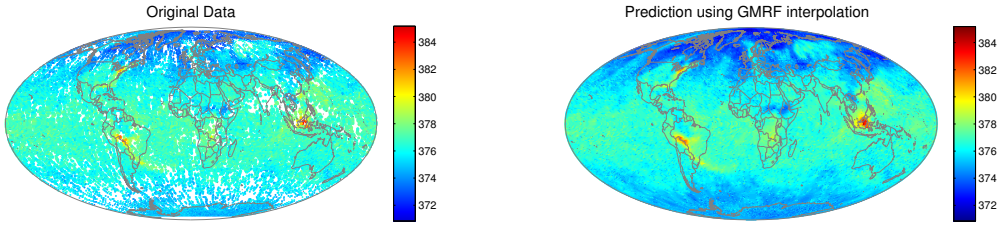

FIG. 5.3. *GMRF interpolation for CO_2 data. Left: Original data with missing values. Right: GMRF interpolated values with parameter $\xi = 0.2$.*

which resulted in an $O(n \log n)$ cost [21] (the covariance matrices are circulant). With the estimated length scale, we perform a prediction calculation for the rest of the data (white pixels in fig. 5.2(b)) and show the predicted values, together with the ten percent used for training, in fig. 5.2(d). We observe that the pattern obtained from the predicted values appears quite similar to the original data pattern. The relative difference between fig. 5.2(a) and fig. 5.2(d) is 0.27. This relative error is due the the randomness in sampling the data from the original and estimated covariance fields.

Spatial Analysis using GMRF for CO_2 data. Next, we consider the Gaussian Markov Random field (GMRF) [36] parameter estimation problem for real spatial data with missing entries. We use a global dataset of column-integrated CO_2 obtained from `http://niasra.uow.edu.au/cei/webprojects/UOW175995.html`. The values of column-integrated CO_2 are on a grid of $1.25°$ longitude by $1°$ latitude, which results in a total of $288 \times 181 = 52,128$ grid cells (matrix size) on the globe [29]. The dataset has $26,633$ observations. We assume GRMF model for the data and use maximum likelihood estimation to predict the remaining (missing) values. For the GMRF field, we considered the spatial autoregressive (SAR) model, i.e., the precision matrix is defined as $G(\xi) = \xi^4 C + \xi^2 G_1 + G_2$, where matrices C, G_1 and G_2 define the neighborhood (four, eight and 16 neighbors, respectively) and are sparse [36]. We obtain ML estimates using the SLQ method to choose the optimal parameter ξ. That is, we sweep through a set of values for ξ and estimate the log-likelihood

TABLE 5.2
Estimation of the sum of singular values of various matrices

Matrices (Applications)	Size	m	Exact Sum	Estimated Sum	Time (secs)	SVD time
Erdos992 (undirected graph)	6100	40	3292.06	3294.5	1.05	876.2 secs
deter3 (linear programming)	7047	30	16518.08	16508.46	1.62	1.3 hrs
California (web search)	9664	100	3803.74	3803.86	8.32	4.17 mins
FA (Pajek network graph)	10617	150	1306.79	1312.8	23.13	1.5 hrs
qpband (optimization)	20000	60	26708.14	26710.1	0.35	2.9 hrs

for the data given by $\log p(z \mid \xi) = \log\det G(\xi) - z^\top G(\xi) z - \frac{n}{2}\log(2\pi)$, and determine the parameter ξ that maximizes the log-likelihood. Figure 5.3(left) shows the sparse observations of the CO_2 data across the globe. The GMRF interpolation with the parameter $\xi = 0.2$ is given in fig.5.3(right).

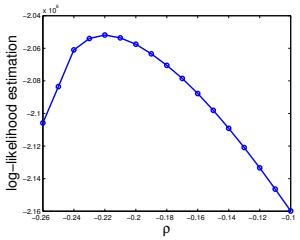

FIG. 5.4. *Log-likelihood estimation for square GMRF of size* 1000×1000 *with parameters* ξ.

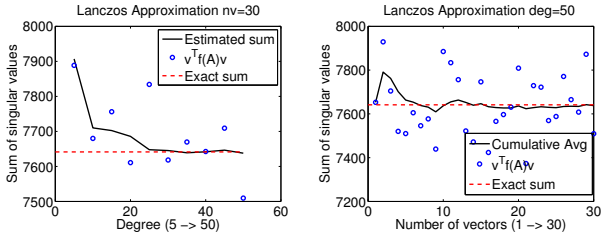

FIG. 5.5. *The nuclear norm estimated by SLQ for the example* `ukerbe1` *matrix (left) as a function of degree* m *and (right) as a function of number of starting vectors* $\mathrm{n_v}$.

Maximum Likelihood estimation for GMRF. In this experiment, we apply the SLQ method for Maximum Likelihood (ML) estimation of Gaussian Markov Random fields (GMRF) [36], which is a popular model used in many computer vision, spatial statistics and other applications. First, we consider a square grid of 1000×1000, so we have $n = 10^6$ variables and consider a GMRF on the grid with a precision matrix $Q(\xi) \in \mathbb{R}^{n \times n}$, where ξ is the correlation parameter (the correlation between each node and its four neighbors). A data vector z is sampled from the the GMRF model with parameter $\xi = -0.22$. The objective is to use MLE to estimate the parameter ξ of the GMRF model that best fits the data z observed, i.e., we sweep through a set of values for ξ and estimate the log-likelihood for the data given by $\log p(z \mid \xi) = \log\det Q(\xi) - z^\top Q(\xi) z - \frac{n}{2}\log(2\pi)$, and determine the parameter ξ that maximizes the log-likelihood. Figure 5.4 plots the log-likelihood obtained for different parameters ξ, and we see that log-likelihood is maximum for $\xi = -0.22$.

Nuclear Norm. Next, let us consider the estimation of the nuclear norm of a matrix for examining the effects of the parameters m and $\mathrm{n_v}$ in the SLQ performance. We will consider the same matrix *ukerbe1* of size 5981×5981 from the UFL database. The performance of the SLQ method in approximately estimating the sum of singular values of this matrix is given in figure 5.5.

The left figure plots the estimated nuclear norm for different number of Lanczos steps m used, with the number of starting vectors $\mathrm{n_v} = 30$ (black solid line). The right figure plots the $v^T f(A) v$ values (blue circles) obtained for different starting vectors and the cumulative average for Lanczos Quadrature of degree $m = 50$. The nuclear norm estimated for degree $m = 50$ and $\mathrm{n_v} = 30$ was 7640.62. The exact sum of singular values is 7641.44.

Finally, we employ our SLQ algorithm 2 with G-K-B for the nuclear norm estimation of real datasets. Table 5.2 lists the approximate nuclear norm estimated by our method for a set of matrices from various applications. All matrices were obtained from the UFL database [14] and are sparse. The matrices, their applications and sizes are listed in the first two columns of the table. We increment the degree m (number of G-K-B steps) until we achieve 3-4 digit accuracy. The degree used and the approximate sum obtained are listed in the table along with the exact sum and the time taken (averaged over 5 trials) by our algorithm. In all experiments, the number of starting vectors $\mathrm{n_v} = 30$. In addition, we also list the time taken to compute only

the top 2000 singular values of each matrices (computed using MATLAB's `svds` function which relies on ARPACK) in order to provide a rough illustration of the potential computational gain of our algorithm over partial SVD.

Experiments illustrating the performance of SLQ for approximating the spectral densities of matrices can be seen in [31], and for estimating the numerical rank of matrices can be seen in [41]. All the experiments reported here (and in [31, 41]) show that, we can obtain accurate estimation of the traces of various functions of large matrices using the Stochastic Lanczos Quadrature Approximation method, with just a low number of Lanczos steps and a low number of starting vectors. Both theory and experiments show that the SLQ method will be superior to other popular methods.

Acknowledgements. J. Chen is supported in part by the XDATA program of the Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

REFERENCES

[1] A. Anandkumar, F. Huang, D. J. Hsu, and S. M. Kakade. Learning mixtures of tree graphical models. In *Advances in Neural Information Processing Systems*, pages 1052–1060, 2012.

[2] A. Andoni, R. Krauthgamer, and I. Razenshteyn. Sketching and embedding are equivalent for norms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 479–488. ACM, 2015.

[3] E. Aune, D. P. Simpson, and J. Eidsvik. Parameter estimation in high dimensional Gaussian distributions. *Statistics and Computing*, 24(2):247–263, 2014.

[4] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8, 2011.

[5] Z. Bai, G. Fahey, and G. Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1):71–89, 1996.

[6] Z. Bai and G. H. Golub. Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices. *Annals of Numerical Mathematics*, 4:29–38, 1996.

[7] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. *Applied numerical mathematics*, 57(11):1214–1229, 2007.

[8] C. Boutsidis, P. Drineas, P. Kambadur, and A. Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *arXiv preprint arXiv:1503.00374*, 2015.

[9] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[10] R. Carbó-Dorca. Smooth function topological structure descriptors based on graph-spectra. *Journal of Mathematical Chemistry*, 44(2):373–378, 2008.

[11] J. Chen. How accurately should i compute implicit matrix-vector products when applying the hutchinson trace estimator? *IBM Technical Report, RC25581*, 2015.

[12] J. Chen, M. Anitescu, and Y. Saad. Computing f(a)b via least squares polynomial approximations. *SIAM Journal on Scientific Computing*, 33(1):195–222, 2011.

[13] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[14] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

[15] E. Di Napoli, E. Polizzi, and Y. Saad. Efficient estimation of eigenvalue counts in an interval. *ArXiv preprint ArXiv:1308.4275*, 2013.

[16] E. Estrada. Characterization of 3d molecular structure. *Chemical Physics Letters*, 319(5):713–718, 2000.

[17] E. Estrada and N. Hatano. Statistical-mechanical approach to subgraph centrality in complex networks. *Chemical Physics Letters*, 439(1):247–251, 2007.

[18] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

[19] G. H. Golub and G. Meurant. *Matrices, moments and quadrature with applications*. Princeton University Press, 2009.

[20] G. H. Golub and Z. Strakoš. Estimates in quadratic formulas. *Numerical Algorithms*, 8(2):241–268, 1994.

[21] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[22] G. H. Golub and J. H. Welsch. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969.

[23] N. Hale, N. J. Higham, and L. N. Trefethen. Computing a^α,\log(a), and related matrix functions by contour integrals. *SIAM Journal on Numerical Analysis*, 46(5):2505–2523, 2008.

[24] I. Han, D. Malioutov, H. Avron, and J. Shin. Approximating the spectral sums of large-scale matrices using chebyshev approximations. *arXiv preprint arXiv:1606.00942*, 2016.

[25] I. Han, D. Malioutov, and J. Shin. Large-scale log-determinant computation through stochastic chebyshev expansions. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 908–917, 2015.

[26] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pages 2339–2347, 2012.

[27] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

[28] V. Kalantzis, C. Bekas, A. Curioni, and E. Gallopoulos. Accelerating data uncertainty quantification by solving linear systems with multiple right-hand sides. *Numerical Algorithms*, 62(4):637–653, 2013.

[29] M. Katzfuss and N. Cressie. Tutorial on fixed rank kriging (frk) of co2 data. 2011.

[30] Y. Li, H. L. Nguyên, and D. P. Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1562–1581. SIAM, 2014.

[31] L. Lin, Y. Saad, and C. Yang. Approximating spectral densities of large matrices. *SIAM Review*, 58(1):34–65, 2016.

[32] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[33] P. Rabinowitz. Rough and ready error estimates in Gaussian integration of analytic functions. *Communications of the ACM*, 12(5):268–270, 1969.

[34] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[35] F. Roosta-Khorasani and U. Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, pages 1–26, 2014.

[36] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[37] Y. Saad. Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

[38] M. L. Stein, J. Chen, M. Anitescu, et al. Stochastic approximation of score functions for gaussian processes. *The Annals of Applied Statistics*, 7(2):1162–1191, 2013.

[39] L. N. Trefethen. *Approximation theory and approximation practice*. Siam, 2013.

[40] S. Ubaru and Y. Saad. Fast methods for estimating the numerical rank of large matrices. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 468–477, 2016.

[41] S. Ubaru, Y. Saad, and A.-K. Seghouane. Fast estimation of approximate matrix ranks using spectral densities. *arXiv preprint arXiv:1608.05754*, 2016.

[42] M. J. Wainwright and M. I. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing*, 54(6):2099–2109, 2006.

[43] L. Wu, A. Stathopoulos, J. Laeuchli, V. Kalantzis, and E. Gallopoulos. Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse. *arXiv preprint arXiv:1507.07227*, 2015.

[44] Y. Zhang and W. E. Leithead. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *journal of Statistical Computation and Simulation*, 77(4):329–348, 2007.