SIAM J. MATRIX ANAL. APPL.
Vol. 44, No. 3, pp. 1218–1244

KRYLOV-AWARE STOCHASTIC TRACE ESTIMATION*

TYLER CHEN[†] AND ERIC HALLMAN[‡]

Abstract. We introduce an algorithm for estimating the trace of a matrix function $f(\mathbf{A})$ using implicit products with a symmetric matrix $\mathbf{A}$. Existing methods for implicit trace estimation of a matrix function tend to treat matrix-vector products with $f(\mathbf{A})$ as a black box to be computed by a Krylov subspace method. Like other recent algorithms for implicit trace estimation, our approach is based on a combination of deflation and stochastic trace estimation. However, we take a closer look at how products with $f(\mathbf{A})$ are integrated into these approaches which enables several efficiencies not present in previously studied methods. In particular, we describe a Krylov subspace method for computing a low-rank approximation of a matrix function by a computationally efficient projection onto Krylov subspace.

Key words. spectral function, Hutchinson's method, quadratic trace estimation, low-rank approximation, block-Lanczos method, Krylov subspace method

MSC codes. 15A16, 65F50, 65F60, 68W25

DOI. 10.1137/22M1494257

1. Introduction.

We consider the task of estimating $\operatorname{tr}(f(\mathbf{A})) = \sum_{i=1}^{d} f(\lambda_i)$ under the assumption that $\mathbf{A}$ is accessed by matrix-vector products (matvecs). Here $f:\mathbb{R}\rightarrow\mathbb{R}$ is a scalar function, $\mathbf{A}\in\mathbb{R}^{d\times d}$ is a symmetric matrix with eigenvalues $\{\lambda_i\}$, and $f(\mathbf{A})$ is the corresponding matrix function. Many algorithms for this problem can be broken into two components: (i) computing the trace of an arbitrary symmetric matrix $\mathbf{B}\in\mathbb{R}^{d\times d}$ accessing $\mathbf{B}$ only by matvecs; and (ii) approximating matvecs with $\mathbf{B}=f(\mathbf{A})$ by means of a Krylov subspace method. Each of these individual tasks has been studied extensively, and several analyses have aimed to balance the costs of the two components [43, 29, 55, 7].

It is well known that $\mathbb{E}[\boldsymbol{\psi}^\mathsf{T}\mathbf{B}\boldsymbol{\psi}] = \operatorname{tr}(\mathbf{B})$ if $\boldsymbol{\psi}\in\mathbb{R}^d$ satisfies $\mathbb{E}[\boldsymbol{\psi}\boldsymbol{\psi}^\mathsf{T}] = \mathbf{I}$. The quadratic trace estimator[1] $\boldsymbol{\psi}^\mathsf{T}\mathbf{B}\boldsymbol{\psi}$ forms the backbone of the most common stochastic trace estimation algorithms. When the entries of $\boldsymbol{\psi}$ are independent and identically distributed (i.i.d.) standard Gaussians, it is known that the estimator has variance $2\|\mathbf{B}\|_\mathsf{F}^2$, where $\|\cdot\|_\mathsf{F}$ denotes the Frobenius norm. Thus, if the spectrum of $\mathbf{B}$ decays quickly, it can be advantageous to compute a low-rank approximation to $\mathbf{B}$ and apply the quadratic trace estimator to the remainder. Similar intuition holds for other common choices of $\boldsymbol{\psi}$ such as i.i.d. Rademacher (±1) entries or i.i.d. Gaussian entries normalized so that $\|\boldsymbol{\psi}\|_2=\sqrt{d}$.

A number of past works have aimed to combine low-rank approximation with quadratic trace estimation [21, 58, 60, 20, 37, 40, 39, 43, 4, 12]. Perhaps the most well known is the Hutch++ algorithm [39] for approximating the trace of an arbitrary

*Received by the editors May 4, 2022; accepted for publication (in revised form) by D. Kressner May 15, 2023; published electronically August 18, 2023.
https://doi.org/10.1137/22M1494257

Funding: This material is based on work supported by the National Science Foundation under grants DGE-1762114 and DMS-1745654. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[†]New York University, New York, NY 10012, USA (tyler.chen@nyu.edu).
[‡]North Carolina State University, Raleigh, NC 27695, USA (eric.r.hallman@gmail.com).

[1] Quadratic trace estimators are sometimes called Hutchinson's trace estimators although they were used prior to Hutchinson's paper [31].

implicit matrix[2] $\mathbf{B}$. However, in the case $\mathbf{B} = f(\mathbf{A})$, Hutch++ does IMA not take advantage of knowledge that matvecs with $\mathbf{B} = f(\mathbf{A})$ are typically approximated by a Krylov subspace method. Moreover, at least in its original form, Hutch++ and related variants must be run separately for each implicit matrix. This is in contrast to simple quadratic trace estimation based Krylov subspace methods which essentially produce a quadrature approximation that can be used to simultaneously approximate $\operatorname{tr}(f(\mathbf{A}))$ for multiple functions f efficiently [2, 3, 48, 7].

The primary goal of this paper is to show that when matvecs with $f(\mathbf{A})$ are computed via a Krylov subspace method, more efficient algorithms for estimating $\operatorname{tr}(f(\mathbf{A}))$ are possible. This is enabled by two critical observations. First, rather than treating $f(\mathbf{A})$ as an arbitrary matrix, it makes sense to think about how the spectrum of $f(\mathbf{A})$ depends on the spectrum of $\mathbf{A}$. Indeed, the natural primitive operation in our setting is matvecs with $\mathbf{A}$. Second, matvecs with $f(\mathbf{A})$ and a set of vectors can be approximated very efficiently when the vectors themselves are elements of a Krylov subspace generated with $\mathbf{A}$.

Contributions. Our main contribution is Algorithm 3.1, which combines low-rank approximation with quadratic trace estimation to estimate $\operatorname{tr}(f(\mathbf{A}))$. It is similar in style to Hutch++ but exploits the structure of a block-Krylov space to reduce the required number of matvecs. In particular, we show how a low-rank approximation to $f(\mathbf{A})$ can be computed more efficiently, including for multiple functions f simultaneously. In subsection 3.1 we present the details of this algorithm and discuss its relation to past work. In section 4 we present two variants: Algorithm 4.1, an adaptive version that takes as input an error tolerance and failure probability, and Algorithm 4.2, a version designed for situations with a limited amount of memory available. Numerical experiments in section 5 show that our methods compare favorably to existing ones.

1.1. Motivating example.

Consider the matrix function $f(\mathbf{A}) = \exp(-\beta\mathbf{A})$ parameterized by the scalar $\beta > 0$. This function arises in equilibrium quantum thermodynamics as the partition function $Z(\beta) \equiv \operatorname{tr}(\exp(-\beta\mathbf{A}))$ and gives us access to properties of a quantum system such as the specific heat, magnetization, and entropy. Evaluating the dependence of $Z(\beta)$ on the inverse Boltzmann temperature β is of general interest in the study of quantum systems [58, 48, 13, 7].

As depicted in Figure 1.1, the spectrum of $\exp(-\beta\mathbf{A})$ depends strongly on the value of β. The spectrum decay influences which algorithms are most suited for

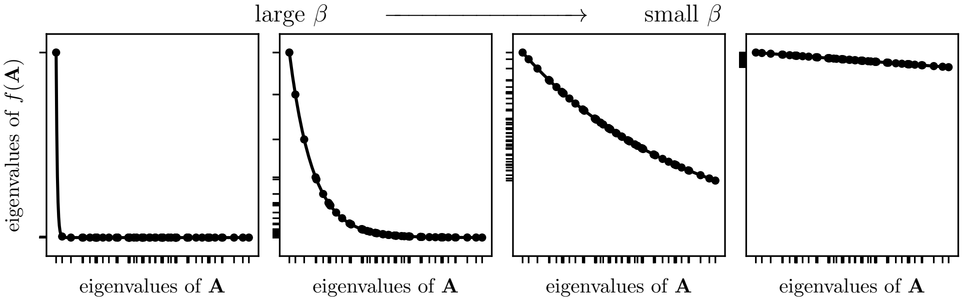

FIG. 1.1. *Dependence of eigenvalues of* $f(\mathbf{A}) = \exp(-\beta\mathbf{A})$ *on the parameter* β. *Observe that the decay in the spectrum of* $\exp(-\beta\mathbf{A})$ *can be extremely fast if* β *is large (left) or extremely slow if* β *is small (right). We would like an algorithm which can approximate* $\operatorname{tr}(\exp(-\beta\mathbf{A}))$ *for many values of* β *simultaneously.*

[2] By "implicit matrix," we mean that $\mathbf{B}$ is assumed to be accessible only through matvecs.

computing the trace of the matrix function. For instance, when β is large, the partition function $Z(\beta)$ is determined almost entirely by the smallest eigenvalue of $\mathbf{A}$ (ground state energy) and approaches based on low-rank approximation are effective [47, 40, 36]. On the other hand, when β is small, the spectrum of $\exp(-\beta\mathbf{A})$ is relatively flat and standard quadratic trace estimators [21, 49, 31] work very well. For intermediate β, a combination of these approaches is effective; the contribution to the trace of the top several eigenvalues can be computed directly using a low-rank approximation, and the contribution of the remaining eigenvalues can be computed using quadratic trace estimators.

1.2. Notation.

Bolded capital letters $\mathbf{A}$, $\mathbf{\Omega}$ denote matrices. Bolded lowercase Roman and Greek letters $\mathbf{x}$, $\mathbf{y}$, $\boldsymbol{\psi}$ denote vectors. The vector $\mathbf{e}_i$ denotes the ith column of the identity matrix $\mathbf{I}_n$, whose size n can be inferred from context. Similarly, the matrix $\mathbf{E}_i$ denotes the Kronecker product $\mathbf{e}_i \otimes \mathbf{I}_b$, i.e., columns $(i-1)b+1$ through ib of $\mathbf{I}_{nb}$, where n and b can be inferred from context. The all-zero matrix is $\mathbf{0}$, and its dimensions can be inferred from context.

The transpose of a matrix is $\mathbf{A}^\mathsf{T}$. The number of columns of $\mathbf{A}$ is $\operatorname{cols}(\mathbf{A})$. An *orthonormal* matrix $\mathbf{Q}$ is one with orthonormal columns, i.e., $\mathbf{Q}^\mathsf{T}\mathbf{Q}=\mathbf{I}$. If a matrix has the eigenvalue decomposition $\mathbf{A}=\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T}$, the matrix function $f(\mathbf{A})$ is $\mathbf{U}f(\mathbf{\Lambda})\mathbf{U}^\mathsf{T}$, where $f(\mathbf{\Lambda})$ applies f to each diagonal entry of $\mathbf{\Lambda}$. For a symmetric matrix $\mathbf{A}$, a *dominant eigenspace* of $\mathbf{A}$ of dimension k is one corresponding to k eigenvalues of $\mathbf{A}$ of largest magnitude. If $\mathbf{A}$ has no repeated eigenvalues, then the dominant eigenspace is unique for each k. We will write $\lambda_{\max}$ and $\lambda_{\min}$ for the largest and smallest eigenvalues of $\mathbf{A}$, $\|\mathbf{A}\|_2$ for the operator norm, and $\|\mathbf{A}\|_\mathsf{F}$ for the Frobenius norm. We denote by $\sigma_{\max}(\mathbf{B})$ and $\sigma_{\min}(\mathbf{B})$ the largest and smallest singular values of $\mathbf{B}$.

The submatrix consisting of rows r through r' and columns c through c' is denoted by $[\mathbf{B}]_{r:r',c:c'}$. A colon with no indices refers to all rows or columns. Thus, $[\mathbf{B}]_{:,1:2}$ denotes the first two columns of $\mathbf{B}$, and $[\mathbf{B}]_{3,:}$ denotes the third row of $\mathbf{B}$.

2. Background.

2.1. Implicit trace estimation.

Given an orthonormal matrix $\mathbf{Q}\in\mathbb{R}^{d\times b}$, define

$$\widehat{\mathbf{B}} \equiv \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{B} + (\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})\mathbf{B}\mathbf{Q}\mathbf{Q}^\mathsf{T},$$
$$\widetilde{\mathbf{B}} \equiv (\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})\mathbf{B}(\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T}),$$

and note that the linearity of the trace operator implies that

$$\operatorname{tr}(\mathbf{B}) = \operatorname{tr}(\widehat{\mathbf{B}}) + \operatorname{tr}(\widetilde{\mathbf{B}}). \tag{2.1}$$

The cyclic property of the trace implies that

$$\operatorname{tr}(\widehat{\mathbf{B}}) = \operatorname{tr}(\mathbf{Q}^\mathsf{T}\mathbf{B}\mathbf{Q}),$$

which allows us to efficiently compute this term exactly. If we define $\mathbf{y}_i \equiv (\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})\boldsymbol{\psi}_i$, where $\boldsymbol{\psi}_i$ is drawn from some spherically symmetric distribution, then the $\mathbf{y}_i/\|\mathbf{y}_i\|_2$ are distributed uniformly on the unit hypersphere in the complement of the column span of $\mathbf{Q}$. Therefore,

$$(d-b)\mathbb{E}\left[\frac{\mathbf{y}_i\mathbf{y}_i^\mathsf{T}}{\mathbf{y}_i^\mathsf{T}\mathbf{y}_i}\right] = (\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T}),$$

where $(\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})$ acts as the identity operator on the complement of the column span of $\mathbf{Q}$. Again using the cylic property of the trace and that $(\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})^2=(\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})$, we see that

$$\operatorname{tr}(\widetilde{\mathbf{B}})=\operatorname{tr}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})\mathbf{B}(\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T}))=(d-b)\mathbb{E}\left[\frac{\mathbf{y}_i^\mathsf{T}\mathbf{B}\mathbf{y}_i}{\mathbf{y}_i^\mathsf{T}\mathbf{y}_i}\right].$$

Thus, we obtain an unbiased estimator

$$\operatorname{tr}(\mathbf{B})\approx\operatorname{tr}(\mathbf{Q}^\mathsf{T}\mathbf{B}\mathbf{Q})+\frac{d-b}{m}\sum_{i=1}^{m}\frac{\mathbf{y}_i^\mathsf{T}\mathbf{B}\mathbf{y}_i}{\mathbf{y}_i^\mathsf{T}\mathbf{y}_i}. \tag{2.2}$$

A simple choice of spherically symmetric distribution for $\boldsymbol{\psi}_i$ is for the entries to be i.i.d. Gaussians. In this case, while $\mathbf{y}_i^\mathsf{T}\mathbf{y}_i$ concentrates around $d-b$ when $d-b$ is large, for finite $d-b$, using the normalizing factors $(d-b)/\mathbf{y}_i^\mathsf{T}\mathbf{y}_i$ can significantly improve performance over using the unnormalized vectors $\mathbf{y}_i$ alone. The effect is particularly noticeable when the spectrum of $\widetilde{\mathbf{B}}$ is flat. The use of such a normalization in the context of stochastic trace estimation was suggested in [12, section 2.3].

The critical observation about an estimator like (2.2) is that if $\mathbf{Q}$ is close to the dominant eigenspace of $\mathbf{B}$, then $\widetilde{\mathbf{B}}$ may have a Frobenius norm significantly smaller than that of $\mathbf{B}$. The estimator $\mathbf{y}_i^\mathsf{T}\mathbf{B}\mathbf{y}_i/\mathbf{y}_i^\mathsf{T}\mathbf{y}_i$ used in (2.2) has variance closely related to $\|\widetilde{\mathbf{B}}\|_\mathsf{F}^2$, so the required number of random vectors used may be reduced significantly compared to if quadratic trace estimation was used with $\mathbf{B}$ directly.

A natural approach to finding a $\mathbf{Q}$ near the dominant eigenspace of $\mathbf{B}$ is by *sketching* [27]. This can be done by taking $\mathbf{Q}$ to be an orthonormal basis for $\mathbf{B}\boldsymbol{\Omega}$, where $\boldsymbol{\Omega}\in\mathbb{R}^{d\times b}$ is a random test matrix. Thus, we arrive at Algorithm 2.1 which returns an estimate to $\operatorname{tr}(\mathbf{B})$ using $2b+m$ matvecs with $\mathbf{B}$. This is essentially the Hutch++ algorithm of [39], except that the latter specifies that $b=m$ and that $\boldsymbol{\Omega}$ and $\boldsymbol{\Psi}$ have i.i.d. Rademacher entries, and the normalization scheme of [12] was not used.

2.2. Krylov subspace methods for matrix function approximation.

It is common to use Krylov subspace methods to approximate products with $f(\mathbf{A})$ [30, 17]. Specifically, expressions of the form $f(\mathbf{A})\mathbf{Z}$ and $\mathbf{Z}^\mathsf{T}f(\mathbf{A})\mathbf{Z}$ can be approximated using information from the block-Krylov subspace

$$\mathcal{K}_{q+1}(\mathbf{A},\mathbf{Z})=\operatorname{span}\{\mathbf{Z},\mathbf{A}\mathbf{Z},\ldots,\mathbf{A}^q\mathbf{Z}\}. \tag{2.3}$$

Here the span is interpreted as the span of the union of the columns of the constituent matrices.

The block-Lanczos algorithm [23] takes a starting matrix $\mathbf{Z}\in\mathbb{R}^{d\times b}$ with QR factorization $\mathbf{Z}=\mathbf{Q}_1\mathbf{R}_1$, and after q steps, computes a matrix

$$\bar{\mathbf{Q}}_{q+1}=[\mathbf{Q}_1,\mathbf{Q}_2,\ldots,\mathbf{Q}_{q+1}]\in\mathbb{R}^{d\times(q+1)b} \tag{2.4}$$

Algorithm 2.1 Implicit stochastic trace estimation [39].

```
procedure IMPLICIT-TRACE($\mathbf{B}, b, m$)
    sample Gaussian matrices $\boldsymbol{\Omega}\in\mathbb{R}^{d\times b}$, $\boldsymbol{\Psi}\in\mathbb{R}^{d\times m}$
    $\mathbf{Q} = \textsc{orth}(\mathbf{B}\boldsymbol{\Omega})$
    $t_{\text{defl}} = \operatorname{tr}(\mathbf{Q}^\mathsf{T}\mathbf{B}\mathbf{Q})$
    $\mathbf{Y} = (\mathbf{I}-\mathbf{Q}\mathbf{Q}^\mathsf{T})\boldsymbol{\Psi}$                    ▷ $\mathbf{Y} = [\mathbf{y}_1,\ldots,\mathbf{y}_m]$
    $t_{\text{rem}} = \frac{d-b}{m}\sum_{i=1}^{m}\mathbf{y}_i^\mathsf{T}\mathbf{B}\mathbf{y}_i/\mathbf{y}_i^\mathsf{T}\mathbf{y}_i$
    return $t_{\text{defl}} + t_{\text{rem}}$
```

whose columns form an orthonormal basis for $\mathcal{K}_{q+1}(\mathbf{A},\mathbf{Z})$. The block columns of $\bar{\mathbf{Q}}_{q+1}$ satisfy a symmetric block-tridiagonal recurrence

$$\mathbf{A}\bar{\mathbf{Q}}_q = \bar{\mathbf{Q}}_q\mathbf{T}_q + \mathbf{Q}_{q+1}\mathbf{R}_{q+1}\mathbf{E}_q^{\mathsf{T}}, \tag{2.5}$$

where

$$\mathbf{T}_q = \begin{bmatrix} \mathbf{M}_1 & \mathbf{R}_2^{\mathsf{T}} & & \\ \mathbf{R}_2 & \ddots & \ddots & \\ & \ddots & \ddots & \mathbf{R}_q^{\mathsf{T}} \\ & & \mathbf{R}_q & \mathbf{M}_q \end{bmatrix} \in \mathbb{R}^{qb\times qb}.$$

We call a recurrence of the form (2.5) a block-Lanczos recurrence.[3]

Algorithm 2.2 presents pseudocode for the block-Lanczos algorithm. For reasons which will become apparent, we also include an additional input parameter n, so that the algorithm runs for $q+n$ iterations but only performs the reorthogonalization step in line 11 for the first q iterations. In order to keep the presentation as simple as possible, we assume for Algorithm 2.2 that the Krylov subspace $\mathcal{K}_{q+n}(\mathbf{A},\mathbf{Z})$ has dimension $(q+n)b$. If this assumption is not met, the algorithm can terminate unsuccessfully. A fully practical implementation of the block-Lanczos algorithm should handle rank-deficient blocks by deflation or other means; see, for instance, [62].

After q steps of the process have been carried out, we obtain the approximation

$$f(\mathbf{A})\mathbf{Z} \approx \bar{\mathbf{Q}}_q f(\mathbf{T}_q)\bar{\mathbf{Q}}_q^{\mathsf{T}}\mathbf{Z} = \bar{\mathbf{Q}}_q\left[f(\mathbf{T}_q)\right]_{:,1:b}\mathbf{R}_1 \tag{2.6}$$

and the quadratic form approximation

$$\mathbf{Z}^{\mathsf{T}} f(\mathbf{A})\mathbf{Z} \approx \mathbf{Z}^{\mathsf{T}}\bar{\mathbf{Q}}_q f(\mathbf{T}_q)\bar{\mathbf{Q}}_q^{\mathsf{T}}\mathbf{Z} = \mathbf{R}_1^{\mathsf{T}}\left[f(\mathbf{T}_q)\right]_{1:b,1:b}\mathbf{R}_1, \tag{2.7}$$

where $\mathbf{Z} = \mathbf{Q}_1\mathbf{R}_1$. These approximations are exact when f is a low-degree polynomial.

Algorithm 2.2 Block-Lanczos.

```
 1: procedure BLOCK-LANCZOS($\mathbf{A}, \mathbf{Z}, q, n$)
 2:   $\mathbf{Q}_1, \mathbf{R}_1 = \textsc{qr}(\mathbf{Z})$
 3:   for $k = 1, 2, \ldots, q+n$ do
 4:     if $k = 1$ then
 5:       $\mathbf{Z} = \mathbf{A}\mathbf{Q}_1$
 6:     else
 7:       $\mathbf{Z} = \mathbf{A}\mathbf{Q}_k - \mathbf{Q}_{k-1}\mathbf{R}_k^{\mathsf{T}}$
 8:     $\mathbf{M}_k = \mathbf{Q}_k^{\mathsf{T}}\mathbf{Z}$
 9:     $\mathbf{Z} = \mathbf{Z} - \mathbf{Q}_k\mathbf{M}_k$
10:     if $2 \le k \le q$ then
11:       $\mathbf{Z} = \mathbf{Z} - \bar{\mathbf{Q}}_{k-1}(\bar{\mathbf{Q}}_{k-1}^{\mathsf{T}}\mathbf{Z})$      ▷ reorthogonalize against $\bar{\mathbf{Q}}_{k-1}$ (2.4)
12:     $\mathbf{Q}_{k+1}, \mathbf{R}_{k+1} = \textsc{qr}(\mathbf{Z})$
13:   return $\mathbf{T}_{q+n}$, $\bar{\mathbf{Q}}_{q+1}$, optionally $\mathbf{R}_1$
```

[3] The dimensions of $\bar{\mathbf{Q}}_{q+1}$ and $\mathbf{T}_q$ are given under the assumption that the blocks $\mathbf{R}_1, \mathbf{R}_2, \ldots$ all have rank b.

Lemma 2.1. *Suppose the Krylov subspace $\mathcal{K}_q(\mathbf{A}, \mathbf{Z})$ has dimension qb (so that no rank-deficient blocks are encountered). Then the approximation* (2.6) *is exact when f is a polynomial of degree at most $q-1$, and the approximation* (2.7) *is exact when f is a polynomial of degree at most $2q-1$.*

Proofs of all lemmas and theorems are given in section 6.

Lemma 2.1 yields a simple bound on the rate of convergence of these approximations.

Lemma 2.2. *Under the assumptions of Lemma* 2.1*, for $q > 0$, the approximation* (2.6) *satisfies*

$$\|f(\mathbf{A})\mathbf{Z} - \bar{\mathbf{Q}}_q \left[f(\mathbf{T}_q)\right]_{:,1:b} \mathbf{R}_1\|_2 \leq 2\|\mathbf{Z}\|_2 \min_{\deg(p)<q} \left(\max_{x \in [\lambda_{\min}, \lambda_{\max}]} |f(x) - p(x)| \right),$$

and the approximation (2.7) *satisfies*

$$\|\mathbf{Z}^{\mathsf{T}} f(\mathbf{A})\mathbf{Z} - \mathbf{R}_1^{\mathsf{T}} \left[f(\mathbf{T}_q)\right]_{1:b,1:b} \mathbf{R}_1\|_2 \leq 2\|\mathbf{Z}\|_2^2 \min_{\deg(p)<2q} \left(\max_{x \in [\lambda_{\min}, \lambda_{\max}]} |f(x) - p(x)| \right).$$

Assuming the interval $[\lambda_{\min}, \lambda_{\max}]$ is known, Lemma 2.2 allows us to apply standard bounds from approximation theory to choose n [53]. For instance, if $\|\mathbf{A}\|_2 \leq 1$ and f is analytically continuable to a Bernstein ellipse E_ρ, where it satisfies $|f(x)| \leq M$ for all $x \in E_\rho$, then the error of the best polynomial approximation to f on $[-1, 1]$ is bounded by $M\rho^{-n}/(\rho - 1)$. Bounds for nonanalytic functions, including ν-times differentiable functions whose $(\nu + 1)$th derivative is of bounded variation, can also be found in [53].

The Lanczos algorithm without reorthogonalization is highly susceptible to the impacts of finite precision arithmetic. While the rates of convergence of the approximations (2.6) and (2.7) are typically reduced when the algorithm is run in finite precision arithmetic, the bound in Lemma 2.2 can still be expected to hold to close degree, *even without reorthogonalization.* This has been shown rigorously for the block size one case [11, 34, 42]. In practice, the bounds in Lemma 2.2 are often very pessimistic, even in finite precision arithmetic. Developing spectrum-dependent a priori and a posteriori bounds suitable for use as practical stopping criteria is an active area of research [18, 19, 32, 15, 14, 16, 6, 61].

Alternate methods for matrix function approximation include explicit polynomial approaches such as those based on Chebyshev series [58, 29]. While these approaches tend to satisfy similar bounds to Lemma 2.2, they typically do not outperform Lanczos-based methods [7, 6].

2.3. A simple algorithm for the trace of a matrix function.

As mentioned in the introduction, it is common to use the ideas from subsection 2.2 to implement the matvecs with $f(\mathbf{A})$ in algorithms like Algorithm 2.1. Such an approach has been used in numerical experiments in [39, 43], and an implementation is described explicitly in Algorithm 2.3.

If n and q are large enough so that products with $f(\mathbf{A})$ are computed almost exactly, then approximation to $\operatorname{tr}(f(\mathbf{A}))$ output by Algorithm 2.3 should be close to that generated by Algorithm 2.1. The following result, which we prove in section 6, guarantees that small errors in the computation of the projection matrix $\mathbf{Q}$ do not significantly impact the guarantees for Algorithm 2.1. The remaining impacts of the error in the approximation (2.6) are analyzed formally in Theorem 3.2.

Algorithm 2.3 Implicit stochastic trace estimation for functions.

```
procedure IMPLICIT-TRACE-FUNC($\mathbf{A}, f, b, q, m, n$)
  sample Gaussian matrices $\mathbf{\Omega} \in \mathbb{R}^{d \times b}$, $\mathbf{\Psi} \in \mathbb{R}^{d \times m}$
  $\mathbf{T}_q, \bar{\mathbf{Q}}_q, \mathbf{R}_1 = $ BLOCK-LANCZOS($\mathbf{A}, \mathbf{\Omega}, 0, q$)
  $\mathbf{Q} = $ ORTH($\bar{\mathbf{Q}}_q [f(\mathbf{T}_q)]_{:,1:b} \mathbf{R}_1$)          ▷ $\approx$ ORTH($f(\mathbf{A})\mathbf{\Omega}$)
  $\mathbf{T}_n, \bar{\mathbf{Q}}_n = $ BLOCK-LANCZOS($\mathbf{A}, \mathbf{Q}, 0, n$)
  $t_{\text{defl}} = \operatorname{tr}([f(\mathbf{T}_n)]_{1:b,1:b})$          ▷ $\approx \operatorname{tr}(\mathbf{Q}^{\mathsf{T}} f(\mathbf{A}) \mathbf{Q})$
  $\mathbf{Y} = (\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathsf{T}})\mathbf{\Psi}$          ▷ $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$
  for $i = 1, 2, \ldots, m$ do
    $\mathbf{T}_n^{(i)} = $ BLOCK-LANCZOS($\mathbf{A}, \mathbf{y}_i, 0, n$)
    $t_{\text{rem}} = t_{\text{rem}} + \frac{d-b}{m} [f(\mathbf{T}_n^{(i)})]_{1,1}$          ▷ $\frac{d-b}{m} \mathbf{y}_i^{\mathsf{T}} \mathbf{B} \mathbf{y}_i / \mathbf{y}_i^{\mathsf{T}} \mathbf{y}_i$
  return $t_{\text{defl}} + t_{\text{rem}}$
```

LEMMA 2.3. *Suppose Algorithm* 2.3 *terminates successfully. Define*

$$\Delta \equiv \|f(\mathbf{A})\mathbf{\Omega} - \bar{\mathbf{Q}}_q \left[f(\mathbf{T}_q)\right]_{:,1:b} \mathbf{R}_1\|_2 / \|\mathbf{\Omega}\|_2,$$

where $\bar{\mathbf{Q}}_q$*,* $\mathbf{T}_q$*, and* $\mathbf{R}_1$ *are as in Algorithm* 2.3*. Suppose* $\mathbf{V} \equiv \text{ORTH}(f(\mathbf{A})\mathbf{\Omega})$ *and* $\mathbf{Q} \equiv \text{ORTH}(\bar{\mathbf{Q}}_q \left[f(\mathbf{T}_q)\right]_{:,1:b} \mathbf{R}_1)$ *are full rank and introduce orthogonal projectors*

$$\mathbf{P}_{\mathbf{V}} \equiv (\mathbf{I} - \mathbf{V}\mathbf{V}^{\mathsf{T}}), \qquad \mathbf{P}_{\mathbf{Q}} \equiv (\mathbf{I} - \mathbf{Q}\mathbf{Q}^{\mathsf{T}}).$$

Then,

$$\|\mathbf{P}_{\mathbf{Q}} f(\mathbf{A}) \mathbf{P}_{\mathbf{Q}} - \mathbf{P}_{\mathbf{V}} f(\mathbf{A}) \mathbf{P}_{\mathbf{V}}\|_{\mathsf{F}} \leq 2\sqrt{2b}\, \kappa(f(\mathbf{A}))\, \kappa(\mathbf{\Omega}) \Delta.$$

Note that while Δ is random, Lemma 2.2 gives a deterministic prior bound for how large q must be set, in terms of the best polynomial approximation to f on $[\lambda_{\min}, \lambda_{\max}]$, to ensure Δ is small (assuming the block-Lanczos algorithm Algorithm 2.2 terminates successfully). We state the lemma in terms of Δ rather than the bound in Lemma 2.2 as there are many bounds for Lanczos besides the simple ones stated in Lemma 2.2.

In almost all situations where sketching is used, the block size b is much smaller than the dimension d. In such cases, the condition number of $\mathbf{\Omega}$ will be relatively small (e.g., < 4) with high probability [56]. Thus, Lemma 2.3 says that the projection step in Algorithm 2.3 behaves almost identically to that of Algorithm 2.1, as long as $f(\mathbf{A})\mathbf{\Omega}$ is computed fairly accurately in Algorithm 2.3.

Note also that

$$\|\mathbf{P}_{\mathbf{V}} f(\mathbf{A}) \mathbf{P}_{\mathbf{V}}\|_{\mathsf{F}} \leq \|f(\mathbf{A}) \mathbf{P}_{\mathbf{V}}\|_{\mathsf{F}} = \|f(\mathbf{A}) - f(\mathbf{A}) \mathbf{V}\mathbf{V}^{\mathsf{T}}\|_{\mathsf{F}}.$$

Bounds for the quality of the low-rank approximation $f(\mathbf{A})\mathbf{V}\mathbf{V}^{\mathsf{T}}$ to $f(\mathbf{A})$ are well studied; see, for instance, [39, 27]. In particular, $f(\mathbf{A})\mathbf{V}\mathbf{V}^{\mathsf{T}}$ has nearly the same approximation error as the best rank-k approximation to $f(\mathbf{A})$ in the Frobenius (or spectral) norm, at least when $b = k + p$ for some small p.

3. Krylov-aware function approximation.

The central insight of this paper is that instead of treating the product $\mathbf{B}\mathbf{\Omega} = f(\mathbf{A})\mathbf{\Omega}$ as a black-box routine, Algorithm 2.3 can be made more efficient by exploiting the structure of the block-Krylov space built with $\mathbf{A}$ and $\mathbf{\Omega}$. In particular, we make use of the fact that if $\bar{\mathbf{Q}}_{q+1}$ is an orthonormal basis for $\mathcal{K}_{q+1}(\mathbf{A}, \mathbf{\Omega})$, then

$$\mathcal{K}_n(\mathbf{A}, \bar{\mathbf{Q}}_{q+1}) = \mathcal{K}_{q+n}(\mathbf{A}, \mathbf{\Omega}). \tag{3.1}$$

Indeed, we have that $\operatorname{span}(\bar{\mathbf{Q}}_{q+1}) = \mathcal{K}_{q+1}(\mathbf{A},\mathbf{\Omega}) = \operatorname{span}\{\mathbf{\Omega},\mathbf{A\Omega},\ldots,\mathbf{A}^q\mathbf{\Omega}\}$. Thus,

$$\begin{aligned}\mathcal{K}_n(\mathbf{A},\bar{\mathbf{Q}}_{q+1}) &= \operatorname{span}\{\bar{\mathbf{Q}}_{q+1},\mathbf{A}\bar{\mathbf{Q}}_{q+1},\ldots,\mathbf{A}^{n-1}\bar{\mathbf{Q}}_{q+1}\}\\ &= \operatorname{span}\{\mathbf{\Omega},\mathbf{A\Omega},\ldots,\mathbf{A}^q\mathbf{\Omega},\\ &\qquad\qquad \mathbf{A\Omega},\mathbf{A}^2\mathbf{\Omega},\ldots,\mathbf{A}^{q+1}\mathbf{\Omega},\\ &\qquad\qquad\qquad \ddots\\ &\qquad\qquad\qquad \mathbf{A}^q\mathbf{\Omega},\mathbf{A}^{q+1}\mathbf{\Omega},\ldots,\mathbf{A}^{q+n-1}\mathbf{\Omega}\} = \mathcal{K}_{n+q}(\mathbf{A},\mathbf{\Omega}).\end{aligned}$$

We note that the block-Lanczos algorithm Algorithm 2.2 can be implemented to automatically respect (3.1) without breaking down[4].

The relation (3.1) suggests that, given $\bar{\mathbf{Q}}_{q+1}$, we can construct an approximation to $\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}} f(\mathbf{A})\bar{\mathbf{Q}}_{q+1}$ using $n-1$ additional products with matrices of size just $n\times b$. In particular, we can use the approximation

$$\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}}\bar{\mathbf{Q}}_{q+n+1} f(\mathbf{T}_{q+n})\bar{\mathbf{Q}}_{q+n+1}^{\mathsf{T}}\bar{\mathbf{Q}}_{q+1} = [f(\mathbf{T}_{q+n})]_{1:(q+1)b,1:(q+1)b}. \tag{3.2}$$

This approximation is exact when f is a sufficiently low-degree polynomial. In particular, we have the following lemma.

Lemma 3.1. *Assuming the Krylov subspace $\mathcal{K}_{q+n}$ has dimension $(q+n)b$ so that no rank-deficient blocks are encountered, the approximation* (3.2) *is exactly equal to $\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}} f(\mathbf{A})\bar{\mathbf{Q}}_{q+1}$ whenever f is a polynomial of degree at most $2n-1$.*

In addition, the "Krylov-aware" approach yields a low-rank approximation to $f(\mathbf{A})$. In particular,

$$\bar{\mathbf{Q}}_{q+1}\,[f(\mathbf{T}_{q+n})]_{1:(q+1)b,1:(q+1)b}\,\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}} \tag{3.3}$$

gives an approximation to the symmetric projection of $f(\mathbf{A})$ onto $\mathcal{K}_{q+1}(\mathbf{A},\mathbf{\Omega})$ which is itself a low-rank approximation to $f(\mathbf{A})$. A theoretical analysis of how well (3.3) works for low-rank approximation would be an interesting topic of further study. In particular, for functions f such as the square root which compress the eigenvalues of $\mathbf{A}$, it seems reasonable that this approach would significantly outperform approaches based on approximating $f(\mathbf{A})\mathbf{\Omega}$ with a Krylov subspace method (see [44] for some related theoretical results in this direction).

3.1. Main algorithm.

Our "Krylov-aware" approach to stochastic trace estimation is summarized in Algorithm 3.1. We make a few observations about the implementation:

- In line 3, the columns of $\bar{\mathbf{Q}}_{q+1}$ should be kept orthonormal. The remaining columns of $\bar{\mathbf{Q}}_{q+n+1}$ do not need to be reorthogonalized against $\bar{\mathbf{Q}}_{q+1}$.
- In line 7, the Krylov basis vectors do not need to be reorthogonalized.
- Lines 6–8 can be run in parallel, blocking matvecs with $\mathbf{A}$. While it would also be possible to run block-Lanczos$(\mathbf{A},\mathbf{Y},0,n)$, doing so does not seem to significantly improve the quality of the estimate since the columns of $\mathbf{Y}$ do not share any notable structure with respect to $\mathbf{A}$.

[4] Suppose that, when a rank-deficient matrix $\mathbf{Z}$ is encountered during the block-Lanczos process, the QR factorization in line 12 of Algorithm 2.2 produces an orthonormal basis for the column space of $\mathbf{Z}$ and the block size is decreased. Then, in exact arithmetic, $q+n$ steps of block-Lanczos on the orthonormal matrix $\mathbf{Q}_1\in\mathbb{R}^{d\times b}$ produces the same output as n steps of block-Lanczos on the input $\bar{\mathbf{Q}}_{q+1}\in\mathbb{R}^{d\times(q+1)b}$, where $\bar{\mathbf{Q}}_{q+1}$ is the result of running q steps of block-Lanczos on $\mathbf{Q}_1$. We do not provide such an implementation as the details would obfuscate the main point of this paper.

Algorithm 3.1 Krylov-aware stochastic trace estimation for matrix functions.

```
1: procedure KRYLOV-AWARE-TRACE($\mathbf{A}, f, b, q, m, n$)
2:     Sample Gaussian matrices $\mathbf{\Omega} \in \mathbb{R}^{d\times b}$ and $\mathbf{\Psi} \in \mathbb{R}^{d\times m}$
3:     $\mathbf{T}_{q+n}, \bar{\mathbf{Q}}_{q+1} = $ BLOCK-LANCZOS($\mathbf{A}, \mathbf{\Omega}, q, n$)
4:     $t_{\text{defl}} = \operatorname{tr}\left([f(\mathbf{T}_{q+n})]_{1:(q+1)b,1:(q+1)b}\right)$        $\triangleright \approx \operatorname{tr}(\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}} f(\mathbf{A})\bar{\mathbf{Q}}_{q+1})$
5:     $\mathbf{Y} = (\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}})\mathbf{\Psi}$        $\triangleright \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$
6:     for $i = 1, 2, \ldots, m$ do
7:         $\mathbf{T}_n^{(i)} = $ BLOCK-LANCZOS($\mathbf{A}, \mathbf{y}_i, 0, n$)
8:         $t_{\text{rem}} = t_{\text{rem}} + \frac{d-(q+1)b}{m}[f(\mathbf{T}_n^{(i)})]_{1,1}$        $\triangleright \approx \frac{d-(b+1)q}{m}\mathbf{y}_i^{\mathsf{T}} f(\mathbf{A})\mathbf{y}_i/\mathbf{y}_i^{\mathsf{T}}\mathbf{y}_i$
9:     return $t_{\text{defl}} + t_{\text{rem}}$
```

- Lines 6–8 can also be run in parallel with the final n iterations of line 3.
- The algorithm can easily be adapted to approximate $\operatorname{tr}(f(\mathbf{A}))$ for many functions f (which do not even need to be known in advance) at minimal additional cost.

We can provide a simple error guarantee for Algorithms 2.3 and 3.1, given a prescribed choice of b, q, and n. We discuss heuristics and intuition for how b and q can be chosen in the next section.

THEOREM 3.2. *Assume that* $\mathbf{A}$, f, b, q, n *are such that Algorithms* 2.3 *and* 3.1 *terminate successfully with probability one. Set* $\hat{\mathbf{Q}} \equiv \mathbf{Q}$ *in the case of Algorithm* 2.3 *and* $\hat{\mathbf{Q}} \equiv \bar{\mathbf{Q}}_{q+1}$ *in the case of Algorithm* 3.1*, and define* $\hat{b}$ *as the number of columns in* $\hat{\mathbf{Q}}$*. Define the random variable* Δ *by*

$$\Delta \equiv \max\{\Delta_0, \Delta_1, \ldots, \Delta_m\}, \qquad \begin{cases} \Delta_0 & \equiv \left\|\hat{\mathbf{Q}}^{\mathsf{T}} f(\mathbf{A})\hat{\mathbf{Q}} - [f(\mathbf{T}_{q+n})]_{1:\hat{b},1:\hat{b}}\right\|_2 / \|\hat{\mathbf{Q}}\|_2^2, \\ \Delta_i & \equiv \left|\mathbf{y}_i^{\mathsf{T}} f(\mathbf{A})\mathbf{y}_i - \|\mathbf{y}_i\|_2^2 [f(\mathbf{T}_n^{(i)})]_{1,1}\right| / \|\mathbf{y}_i\|_2^2. \end{cases}$$

Here $\mathbf{T}_{q+n}$, $\mathbf{T}_n^{(i)}$, *and* $\mathbf{y}_i = (\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathsf{T}})\boldsymbol{\psi}_i$ *are as in each algorithm. Then, the outputs of the Algorithms* 2.3 *and* 3.1 *satisfy*

$$\left|\operatorname{tr}(f(\mathbf{A})) - \mathbb{E}\big[(t_{\text{defl}} + t_{\text{rem}})\big]\right| \leq d\,\mathbb{E}\big[\Delta\big],$$

$$\mathbb{V}\big[t_{\text{defl}} + t_{\text{rem}}\big] \leq \left(\sqrt{V} + d\sqrt{\mathbb{E}\big[\Delta^2\big]}\right)^2,$$

where, with $\mathbf{F} \equiv (\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathsf{T}}) f(\mathbf{A})(\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathsf{T}})$,

$$V \equiv \frac{2}{m}\left(\frac{d-\hat{b}}{d-\hat{b}+2}\right)\left(\mathbb{E}\big[\|\mathbf{F}\|_{\mathsf{F}}^2\big] - \frac{\mathbb{E}\big[\operatorname{tr}(\mathbf{F})^2\big]}{d-\hat{b}}\right).$$

Moreover, with $\mathbf{P}_{\hat{\mathbf{Q}}} \equiv \mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathsf{T}}$, *and provided the same matrix* $\mathbf{\Omega}$ *is used in both algorithms,* $\|\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}} f(\mathbf{A})\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}}\|_{\mathsf{F}}^2 \leq \|\mathbf{P}_{\mathbf{Q}} f(\mathbf{A})\mathbf{P}_{\mathbf{Q}}\|_{\mathsf{F}}^2$ *and so*

$$\mathbb{E}\big[\|\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}} f(\mathbf{A})\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}}\|_{\mathsf{F}}^2\big] \leq \mathbb{E}\big[\|\mathbf{P}_{\mathbf{Q}} f(\mathbf{A})\mathbf{P}_{\mathbf{Q}}\|_{\mathsf{F}}^2\big].$$

We make several comments on Theorem 3.2. First assuming the algorithm terminates successfully, Δ can be upper bounded by a deterministic quantity which, for reasonable functions f, tends to zero as $n \to \infty$; see subsection 2.2. Second, the variance bound is obtained using the fact that $\frac{d-\hat{b}}{m}\mathbf{y}_i^\mathsf{T} f(\mathbf{A})\mathbf{y}_i/\mathbf{y}_i^\mathsf{T}\mathbf{y}_i$ has known variance V (with respect to the randomness in $\boldsymbol{\psi}_i$) [21, 38]. The concentration of measure phenomenon for the uniform distribution on the sphere [35] implies that this random variable is sub-Gaussian [45]. However, fine-grained concentration inequalities are not as readily available in the literature as the inequalities for Gaussian or Rademacher vectors [39, 43]. Finally, the result shows that if $d\Delta$ is small relative to $\max(1, \sqrt{V})$, then Algorithm 3.1 cannot perform significantly worse than Algorithm 2.3. Our experiments in section 5 show that Algorithm 3.1 often performs significantly better than Algorithm 2.3.

Lemma 2.3 in conjunction with Lemma 2.2 implies that when q increases, the quality of the projection matrix $\mathbf{Q}$ by Algorithm 2.3 converges to the projection matrix which would be generated if products with $f(\mathbf{A})$ were computed exactly (i.e., to the projection matrix generated by Hutch++). For many cases of interest, the latter is a poor way to spend one's computational budget. For example, if $\mathbf{A}$ is symmetric positive semidefinite and $f(x) = x^{1/2}$, then a better projection matrix $\mathbf{Q}$ could be obtained simply by running one step of subspace iteration with $\mathbf{A}$ itself! On the other hand, one can expect the block-Krylov subspace $\bar{\mathbf{Q}}_{q+1}$ to continue to improve as q increases.[5] Exactly how much the block-Krylov space improves, however, is a more difficult question to answer. We discuss the topic further in subsection 3.3.

3.2. Choice of parameters.

In all, Algorithm 3.1 requires $b(q+n)+mn$ matvecs with $\mathbf{A}$ and a minimum of $q+n$ matrix loads, matching the costs of Algorithm 2.3.

This interpretation alone is perhaps too optimistic, since Algorithm 3.1 requires storing and orthogonalizing $\bar{\mathbf{Q}}_{q+1}$. In many situations, the size of $\bar{\mathbf{Q}}_{q+1}$ will be limited by these storage and reorthogonalization costs rather than the number of matvecs with $\mathbf{A}$. Suppose that we have some predetermined limit on the number of vectors to be used for deflation, i.e., $b(q+1)$ is bounded by a constant. What block size will be most useful? Using a small block size will reduce the number of matvecs, while using a larger block size will allow for greater parallelism and reduce the number of matrix loads.

Methods focused on computing a small number of eigenvectors to high accuracy have tended to use small block sizes (including $b = 1$) [59, 52] while more recent analyses of randomized block-Lanczos have tended to consider large b and small q [41, 57, 10, 26]; see [38] for more background. For the problems of finding a low-rank approximation or a few eigenpairs of $\mathbf{A}$, the total number of matrix loads is roughly proportional to q, so the trade-off between the Krylov-depth q and block-size b is not just in the total number of matvecs, but also the number of matrix loads.

In our situation the minimum number of matrix loads is $q + n$. Since n will typically be relatively high (i.e., not some small constant like 3) due to the need to construct a sufficiently large Krylov subspace such that products with $f(\mathbf{A})$ are computed to sufficient accuracy, making q small cannot significantly reduce the overall number of matrix loads. In other words, the presumed benefit of our algorithm is in the reduced number of matvecs required, rather than a reduction in the number of matrix loads. This suggests that when the size of the deflation space is limited, it makes sense to take b small and q large.

[5] If $\mathbf{A}$ has only a few distinct eigenvalues, the dimension of the block-Krylov subspace may stop growing. However, in such cases the approximation $f(\mathbf{A})\boldsymbol{\Omega}$ is exact, so the projection space used by Algorithm 3.1 is no worse than what is used by Algorithm 2.3 (which also becomes exact and therefore is the same as the space used by Algorithm 2.1).

Table 3.1

Approximate costs of algorithms. Here storage is measured in terms of the number of vectors of length d stored. For simplicity, we assume each implicit product with $f(\mathbf{A})$ in Hutch++ is sequentially implemented using n steps of the Lanczos method for matrix function approximation. The number of matrix loads of $\mathbf{A}$, which is more important than the total number of matvecs in some situations, is not displayed here. In general, the number of matrix loads can be reduced at the cost of increased storage.

Algorithm	Number matvecs	Storage	Inner products
Algorithm 2.3	$b(q+n)+mn$	$O(b+n)$	$O(b^2+bn+bm+mn)$
Algorithm 3.1	$b(q+n)+mn$	$O(bq+n)$	$O(b^2(q^2+n)+bqm+mn)$
Algorithm 4.2	$b(qr+q+n)+mn$	$O(bq+n)$	$O(b^2(q^2r+n)+bqm+mn)$

To improve the quality of the approximation to the dominant eigenspace of $f(\mathbf{A})$ while limiting the size of $\bar{\mathbf{Q}}_{q+1}$ (and therefore reorthogonalization and storage costs), it is common to use restarting schemes for block-Lanczos [1, 62]. We describe how these approaches can be used with our algorithm in subsection 4.2. The costs of the algorithms described in this paper are summarized in Table 3.1. One feature worth noting is that since we assume the last n steps of the block-Lanczos process are done without reorthogonalization, the number of inner products required by our algorithms grow at worst proportional to n (as opposed to n^2).[6]

3.3. Relation to past work. A number of papers have integrated Krylov subspace methods into implicit trace approximation algorithms. The most widespread approach has been to use Krylov subspace methods to approximate products with $f(\mathbf{A})$ for quadratic trace estimators [2, 3, 49, 58, 29, 54, etc.]; see [7] for a recent review. This approach is well understood and has been studied in detail. More recently, a number of algorithms aim to incorporate low-rank approximation.

The paper [36] takes $\mathbf{Q}$ to be an orthonormal basis for the entire block-Krylov subspace $\mathcal{K}_{q+1}(\mathbf{A},\mathbf{\Omega})$ and uses it to estimate $\operatorname{tr}(\mathbf{A})$ and $\operatorname{tr}(\log(\mathbf{I}+\mathbf{A}))$ when $\mathbf{A}$ is positive definite. This method builds on the work in [47] (which uses randomized subspace iteration) and is a special case of Algorithm 3.1 with $n=1$ and $m=0$. Our approach works with any n which is critical for general matrix functions whose dominant eigenvalues may not align with those of $\mathbf{A}$.

The paper [60] estimates $\operatorname{tr}(\mathbf{A}^{-1})$ using a low-rank approximation to $\mathbf{A}$ for variance reduction. The authors note that the Krylov space produced with each new sample of the quadratic trace estimator contains information about the eigenvalues and eigenvectors of $\mathbf{A}$, and discuss the possibility of using this information to incrementally update the low-rank approximation. Low-rank approximation and quadratic trace estimation were combined in [40] in order to approximate $\operatorname{tr}(\exp(-\beta\mathbf{A}))$ and related quantities.

A more direct inspiration for our work is [5], in which the authors propose to estimate the diagonal of a matrix using as samples the vectors $\{\mathbf{v}_0,\ldots,\mathbf{v}_k\}$, where $\mathbf{v}_k \equiv T_k(\mathbf{A})\mathbf{v}_0$ and T_k is the kth Chebyshev polynomial of the first kind. By exploiting the shared structure of these vectors, they significantly reduce computational costs. The authors note that these samples are "correlated and not completely random"; our innovation is to apply their technique to the deflation step only, which will allow the trace estimator to remain unbiased.

[6] It is possible to reduce the storage costs of the Lanczos-based matrix function approximation from $O(n)$ to $O(1)$ using a two-pass approach. However, this doubles the number of matvec products from n to $2n$. Other low-memory approaches may be possible for certain functions [25].

All of [47, 36, 44] derive a priori bounds for low-rank approximation of matrix functions. In particular, [44] analyzes low-rank approximation of operator monotone functions using a Nyström-based approximation similar to what is used by a subcase of our restarted variant described in subsection 4.2 (again with $n=1$). This approach is combined with quadratic trace estimation to produce estimates of the trace of operator monotone functions. All of these analyses make critical use of the structure of f, which allows $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T} f(\mathbf{A})\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}$ to be approximated by $f(\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}\mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T})$ if $\mathbf{A}$ is well-approximated by $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}\mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}$. For arbitrary f, it seems possible that $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T} f(\mathbf{A})\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}$ provides a good approximation to $f(\mathbf{A})$ even if $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}\mathbf{A}\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T}$ does not provide a good approximation to $\mathbf{A}$. Thus, the observation that $\hat{\mathbf{Q}}^\mathsf{T} f(\mathbf{A})\hat{\mathbf{Q}}$ can be approximated efficiently if the span of the columns of $\hat{\mathbf{Q}}$ is itself a Krylov subspace (see (3.3)) is a critical aspect of our algorithm. However, this added complexity means that a priori bounds for the quality of the Krylov-aware low-rank approximation used in our algorithm are outside of the scope of this paper. Understanding the theoretical behavior of these algorithms is an interesting topic for further study.

4. Variants.

We discuss several variants of Algorithm 3.1 which may be more practical in many situations.

4.1. Adaptive variant.

One disadvantage of Algorithm 3.1 is that a user may not know ahead of time how much effort should be devoted to variance reduction as opposed to the quadratic trace estimator. In order to ameliorate this problem, we propose an adaptive variant (Algorithm 4.1) that takes parameters $\epsilon > 0$ and $\delta \in (0,1)$ and attempts to compute an estimate satisfying

$$\mathbb{P}\left(|\mathsf{est} - \operatorname{tr}(f(\mathbf{A}))| > \epsilon\right) < \delta.$$

The adaptive algorithm [43, Algorithm 2] provably satisfies such an error guarantee under the assumption that matvecs with $f(\mathbf{A})$ can be computed exactly. We model our algorithm instead on the slightly less rigorous A-Hutch++ [43, Algorithm 3], which is simpler while still having good empirical behavior.

We assume that the block-Lanczos process computes matvecs with $f(\mathbf{A})$ exactly—or more precisely, we assume that n is set large enough for the discrepancy to be ignored. For more detailed discussions on how to select the degree n depending on the function f and spectrum of $\mathbf{A}$, see [22, 54, 7].

We also assume that the block size b is fixed. Once this is done, the adaptive algorithm must decide first how many Lanczos iterations q to devote to variance reduction, and second how many vectors m to use for the quadratic trace estimator. The answer to the second part is straightforward, given an estimate of the remainder $\|\widetilde{\mathbf{B}}\|_\mathsf{F}$. In [8] it is shown that for user-specified tolerances $\epsilon > 0$ and $\delta \in (0,1)$,

$$\mathbb{P}\left[\left|\boldsymbol{\psi}^\mathsf{T}\widetilde{\mathbf{B}}\boldsymbol{\psi} - \operatorname{tr}(\widetilde{\mathbf{B}})\right| > \epsilon\right] < \delta \quad \text{if} \quad m \geq c\epsilon^{-2}(\|\widetilde{\mathbf{B}}\|_\mathsf{F}^2 + \epsilon\|\widetilde{\mathbf{B}}\|_2)\ln(2/\delta), \tag{4.1}$$

where c is a constant depending on the distribution of $\boldsymbol{\psi} \in \mathbb{R}^{d\times m}$.[7]

Relaxing the bound (4.1) via the approximation $\|\widetilde{\mathbf{B}}\|_\mathsf{F}^2 + \epsilon\|\widetilde{\mathbf{B}}\|_2 \approx \|\widetilde{\mathbf{B}}\|_\mathsf{F}^2$, the algorithm A-Hutch++ defines

$$C(\epsilon,\delta) \equiv 4\epsilon^{-2}\log(2/\delta) \tag{4.2}$$

[7] It suffices to take $c=4$ for Gaussian vectors and $c=16$ for Rademacher [8].

Algorithm 4.1 Adaptive Krylov-aware stochastic trace estimation.

```
procedure ADA-TRACE($\mathbf{A}, f, n, b, \epsilon, \delta$)
  Sample Gaussian $\mathbf{\Omega} \in \mathbb{R}^{d \times b}$
  $\mathbf{T}_{q+n}, \bar{\mathbf{Q}}_{q+1} =$ BLOCK-LANCZOS($\mathbf{A}, \mathbf{\Omega}$), run until $\widetilde{M}(q)$ minimized
  $t_{\text{defl}} = \operatorname{tr}\left([f(\mathbf{T}_{q+n})]_{1:(q+1)b,1:(q+1)b}\right)$
  Initialize $t_{\text{rem}} = 0$ and $t_{\text{fro}} = 0$
  Initialize $m_0 = \infty$ and $k = 0$
  while $m_k > k$ do
    $k = k + 1$
    Sample Gaussian $\boldsymbol{\psi}_k \in \mathbb{R}^d$
    Orthogonalize $\mathbf{y}_k = (\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}})\boldsymbol{\psi}_k$
    $\mathbf{T}_n^{(k)} =$ BLOCK-LANCZOS($\mathbf{A}, \mathbf{y}_k, n$)
    Increment $t_{\text{rem}} = t_{\text{rem}} + [f(\mathbf{T}_n^{(k)})]_{1,1}\|\mathbf{y}_k\|_2^2$        ▷ Estimate $\operatorname{tr}(\widetilde{\mathbf{B}})$
    Increment $t_{\text{fro}} = t_{\text{fro}} + \|[f(\mathbf{T}_n^{(k)})]_{:,1}\|_2^2\,\|\mathbf{y}_k\|_2^2$        ▷ Estimate $\|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2$
    $\alpha_k = \frac{1}{k}F_X^{-1}(\delta)$, where $X \sim \chi_k^2$        ▷ Inverse CDF
    $m_k = \frac{1}{k\alpha_k}C(\epsilon,\delta)t_{\text{fro}}$
  return $t_{\text{defl}} + \frac{1}{k}t_{\text{rem}}$
```

and proposes to use $m = C(\epsilon,\delta)\|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2$ samples. In practice A-Hutch++ increases m incrementally while simultaneously updating an estimate of $\|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2$, and we do the same in Algorithm 4.1. We use a slightly paraphrased version of [43, Lemma 2.2], which relies on a result from [46].

LEMMA 4.1. *Let $\mathbf{\Psi} \in \mathbb{R}^{d \times k}$ be a standard Gaussian matrix, and let $\widetilde{\mathbf{B}} \in \mathbb{R}^{d \times d}$. For any $\alpha \in (0,1)$, it holds that*

$$\mathbb{P}\left(\frac{1}{k\alpha}\|\widetilde{\mathbf{B}}\mathbf{\Psi}\|_{\mathsf{F}}^2 \leq \|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2\right) \leq \mathbb{P}(X \leq k\alpha), \tag{4.3}$$

where $X \sim \chi_k^2$ is a chi-squared random variable with k degrees of freedom.

For a user-specified failure tolerance δ, we can therefore define

$$\alpha_k \equiv \frac{1}{k}F_X^{-1}(\delta), \tag{4.4}$$

where F_X is the cumulative distribution function (CDF) of $X \sim \chi_k^2$. It follows from Lemma 4.1 that $\frac{1}{k\alpha_k}\|\widetilde{\mathbf{B}}\mathbf{\Psi}\|_{\mathsf{F}}^2$ will overestimate $\|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2$ with probability at least $1 - \delta$. It is shown in [43] that the sequence $\{\alpha_k\}$ increases monotonically and converges to 1. Algorithm 4.1 uses the same set of random vectors to incrementally estimate $\operatorname{tr}(\widetilde{\mathbf{B}})$ and $\|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2$ simultaneously, stopping when the number of samples k exceeds the sampling number

$$m_k \equiv \frac{1}{k\alpha_k}\|\widetilde{\mathbf{B}}\mathbf{\Psi}\|_{\mathsf{F}}^2 \gtrapprox \|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2.$$

Remark 1. The factor α_k is pessimistic if the stable rank $\|\widetilde{\mathbf{B}}\|_{\mathsf{F}}^2/\|\widetilde{\mathbf{B}}\|_2^2$ is large. Ideally, one should incrementally estimate the stable rank and use it in combination with bounds such as those in [24, 8] to obtain values for α_k closer to 1, and therefore smaller values for m_k. We stick with the definition (4.4) in order to focus on the more novel aspects of our algorithm.

All that remains is to determine the number of iterations q. The total number of matvecs used by Algorithm 3.1, which we use as a proxy for the computational cost, is

$$\begin{aligned}
M(q) &\equiv (q+n)b + nm \\
&\approx (q+n)b + nC(\epsilon,\delta)\|(\mathbf{I}-\bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}})f(\mathbf{A})(\mathbf{I}-\bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}})\|_{\mathsf{F}}^2 \\
&= (q+n)b + nC(\epsilon,\delta)\big(\|f(\mathbf{A})\|_{\mathsf{F}}^2 - 2\|f(\mathbf{A})\bar{\mathbf{Q}}_{q+1}\|_{\mathsf{F}}^2 + \|\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}}f(\mathbf{A})\bar{\mathbf{Q}}_{q+1}\|_{\mathsf{F}}^2\big) \\
&= qb - nC(\epsilon,\delta)\big(2\|f(\mathbf{A})\bar{\mathbf{Q}}_{q+1}\|_{\mathsf{F}}^2 - \|\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}}f(\mathbf{A})\bar{\mathbf{Q}}_{q+1}\|_{\mathsf{F}}^2\big) + \text{constant}.
\end{aligned}$$

Using n steps of block-Lanczos to estimate the above quantity, we get the objective function

$$\widetilde{M}(q) \equiv qb - nC(\epsilon,\delta)\big(2\|[f(\mathbf{T}_{q+n})]_{:,1:(q+1)b}\|_{\mathsf{F}}^2 - \|[f(\mathbf{T}_{q+n})]_{1:(q+1)b,1:(q+1)b}\|_{\mathsf{F}}^2\big).$$

As is done in [43, section 2.1.2], we propose to stop when $\widetilde{M}(q) > \widetilde{M}(q-1) > \widetilde{M}(q-2)$, which suggests that a local minimum has been found.[8] With this approach the quantity $\widetilde{M}(q)$ can only be estimated after a delay of n further iterations, but since Algorithm 3.1 already requires the computation of $\mathbf{T}_{q+n}$ this work is not wasted. Our approach also requires us to compute the eigenvalues of the block tridiagonal matrix $\mathbf{T}$ at each iteration, but in theory the eigenvalues can be updated cheaply with each iteration.

In practice, the true cost of the algorithm is not necessarily directly proportional to the number of matvecs. For instance, storage or reorthogonalization costs are often a computational bottleneck. In such cases, one can update the function $M(q)$ to account for such costs. However, this may make the resulting optimization problem involving $\widetilde{M}(q)$ more difficult.

Algorithm 4.1 presents the adaptive algorithm. As with Algorithm 3.1, the loop can be blocked for efficiency. Note here that the normalization of $\mathbf{y}_k$ differs from Algorithm 3.1. This is essentially an artifact of analysis, as fine-grained concentration inequalities depending on $\|\widehat{\mathbf{B}}\|_{\mathsf{F}}$ for vectors sampled from the hypersphere are less readily available. It is trivial and inexpensive to maintain both normalizations.

4.2. Low-memory variant. In practice, the memory and orthogonalization costs of block-Lanczos may limit the dimension of the Krylov subspace used for variance reduction. In such situations, we aim to find a subspace of $\mathcal{K}_{q+1}(\mathbf{A},\boldsymbol{\Omega})$ that approximates the dominant eigenspace of $f(\mathbf{A})$ without having to store a basis for the entire block-Krylov space. Some of the most widely used techniques for this task are based on restarting the Lanczos recurrence using a carefully chosen subspace of $\mathcal{K}_q(\mathbf{A},\boldsymbol{\Omega})$ [1, 50, 62]. For concreteness and clarity we focus on the implicitly restarted block-Lanczos method [1]. Other techniques may make more sense in specific situations.

Specifically, suppose the block-Lanczos process has been run for q iterations to obtain $\{\mathbf{Q}_k\}, \{\mathbf{M}_k\}, \{\mathbf{R}_k\}$ which satisfy

$$\mathbf{A}\bar{\mathbf{Q}}_q = \bar{\mathbf{Q}}_q\mathbf{T}_q + \mathbf{Q}_{q+1}\mathbf{R}_{q+1}\mathbf{E}_q^{\mathsf{T}}.$$

The restarting process involves updating $\boldsymbol{\Omega} = p(\mathbf{A})\boldsymbol{\Omega} = \bar{\mathbf{Q}}_q[p(\mathbf{T}_q)]_{:,1:b}\mathbf{R}_1$, where p is some degree-q polynomial, and then generating a new Lanczos recurrence starting

[8] In practice, we may also include a preset limit $q_{\max}$ on the number of vectors we are willing to store and orthogonalize. In such cases, one might turn to restarting (as described in the next section) to refine the quality of the low-rank approximation.

with the updated $\mathbf{\Omega}$. This process can then be repeated. Ideally p is large on the desirable eigenvalues of $\mathbf{A}$ and small elsewhere; i.e., p acts as a filtering polynomial. A number of techniques for choosing such polynomials and performing the update step have been studied [1, 50].

If filter polynomials $\{p^{(i)}\}_{i=1}^{r}$ are used, then the final Krylov subspace generated is

$$\mathcal{K}_{q+1}(\mathbf{A}, p^{(r)}(\mathbf{A}) \cdots p^{(1)}(\mathbf{A})\mathbf{\Omega}).$$

While $p^{(r-1)}(\mathbf{A}) \cdots p^{(1)}(\mathbf{A})\mathbf{\Omega}$ could be computed explicitly, the advantage of breaking it into phases is that the process can adapt to the information gained at each step, i.e., the number of restarts r and the polynomials $\{p^{(i)}\}_{i=1}^{r}$ do not have to be chosen in advance. This allows the process to be terminated after a sufficiently desirable approximation is obtained.

One might hope to apply the techniques from subsection 4.1 in order to derive a stopping criterion for the restarting stage of Algorithm 4.2. The main difference is that in many of the cases we envision for using Algorithm 4.2, the rank of the approximation space $\bar{\mathbf{Q}}_{q+1}$ will be fixed. In such cases, the minimum possible value of $\|\widetilde{\mathbf{B}}\|_\mathsf{F}^2 = \|(\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^\mathsf{T}) f(\mathbf{A})(\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^\mathsf{T})\|_\mathsf{F}^2$ will be limited by the quality of the best rank $(q+1)b$ approximation to $f(\mathbf{A})$. Thus, it would be more appropriate to terminate the restarting procedure after the quality of approximation is not improved. This might be done by observing the quantity

$$\operatorname{tr}\left([f(\mathbf{T}_{q+1})]_{1:(q+1)b,1:(q+1)b}\right)$$

before deciding on whether to repeat lines 4 and 5 or proceed to line 7.

One can of course use Lemma 4.1 to estimate $\|\widetilde{\mathbf{B}}\|_\mathsf{F}^2$ in order to determine the number of samples to be used in the second stage of the algorithm. However, the spread of the output of the second stage of the algorithm can be easily controlled by relatively simple statistical methods for *scalar* random variables such as observing the sample variance, bootstrapping, or jackknife.

While we believe that a practical version of our algorithm for application to real problems should incorporate both adaptive parameter selection and restarting, a good implementation is necessarily dependent on the problem at hand and the computing

Algorithm 4.2 Low-memory Krylov-aware stochastic trace estimation.

```
 1: procedure RESTART-TRACE($\mathbf{A}, f, r, \{p^{(i)}\}, b, q, m, n$)
 2:   Sample Gaussian matrices $\mathbf{\Omega} \in \mathbb{R}^{d \times b}$ and $\mathbf{\Psi} \in \mathbb{R}^{d \times m}$
 3:   for $i = 1, 2, \ldots, r$ do
 4:     $\mathbf{T}_q, \bar{\mathbf{Q}}_{q+1} =$ BLOCK-LANCZOS($\mathbf{A}, \mathbf{\Omega}, q$)
 5:     $\mathbf{\Omega} = \bar{\mathbf{Q}}_q [p(\mathbf{T}_q)]_{:,1:b}\mathbf{R}_1$                    ▷ $\mathbf{\Omega} = p^{(i)}(\mathbf{A})\mathbf{\Omega}$
 6:   $\mathbf{T}_{q+n}, \bar{\mathbf{Q}}_{q+1} =$ BLOCK-LANCZOS($\mathbf{A}, \mathbf{\Omega}, q+n$)
 7:   $t_{\text{defl}} = \operatorname{tr}\left([f(\mathbf{T}_{q+n})]_{1:(q+1)b,1:(q+1)b}\right)$     ▷ $\operatorname{tr}(\bar{\mathbf{Q}}_{q+1}^\mathsf{T} f(\mathbf{A})\bar{\mathbf{Q}}_{q+1})$
 8:   $\mathbf{Y} = (\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^\mathsf{T})\mathbf{\Psi}$                    ▷ $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$
 9:   for $i = 1, 2, \ldots, m$ do
10:     $\mathbf{T}_n^{(i)} =$ BLOCK-LANCZOS($\mathbf{A}, \mathbf{y}_i, n$)
11:     $t_{\text{rem}} = t_{\text{rem}} + \frac{d-(q+1)b}{m}[f(\mathbf{T}_n^{(i)})]_{1,1}$     ▷ $\frac{d-(q+1)b}{m}\mathbf{y}_i^\mathsf{T}\mathbf{B}\mathbf{y}_i/\mathbf{y}_i^\mathsf{T}\mathbf{y}_i$
12:   return $t_{\text{defl}} + t_{\text{rem}}$
```

system to be used. Thus, for the sake of clarity, we do not provide a description which combines both adaptive parameter selection and restarting.

5. Numerical experiments.

5.1. Quantum spin systems.

In this example, we consider the task of computing the partition function

$$Z(\beta) = \operatorname{tr}(\exp(-\beta \mathbf{A}))$$

for the isotropic XY Heisenberg spin chain with a magnetic field of strength h pointed in the z-direction [58, 48, 13, 7]. Specifically, the Heisenberg Hamiltonian for such a chain with N spins of spin number $s = 1/2$ is given by

$$\mathbf{A} = 2 \sum_{i=1}^{N-1} \left(\mathbf{s}_i^{\mathrm{x}} \mathbf{s}_{i+1}^{\mathrm{x}} + \mathbf{s}_i^{\mathrm{y}} \mathbf{s}_{i+1}^{\mathrm{y}} \right) + h \sum_{i=1}^{N} \mathbf{s}_i^{\mathrm{z}}.$$

Here $\mathbf{s}_i^{\mathrm{x/y/z}} \in \mathbb{C}^{(2s+1)^N \times (2s+1)^N}$ is defined by

$$\mathbf{s}_i^{\mathrm{x/y/z}} = \underbrace{\mathbf{I} \otimes \cdots \otimes \mathbf{I}}_{i-1 \text{ terms}} \otimes \mathbf{s}^{\mathrm{x/y/z}} \otimes \underbrace{\mathbf{I} \otimes \cdots \otimes \mathbf{I}}_{N-i \text{ terms}},$$

where

$$\mathbf{s}^{\mathrm{x}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{s}^{\mathrm{y}} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \mathbf{s}^{\mathrm{z}} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

We set $h = 0.3$ and $N = 20$ so that $d = 2^{20} = 1048576$ and apply Algorithms 3.1 or 4.2 with several different choices of parameters to compute $Z(\beta)$ for a range of β. For each parameter choice shown in Table 5.1, we run the algorithm independently 100 times and compare it to the true value of $Z(\beta)$, which can be computed analytically for $s = 1/2$ using the standard "Bethe ansatz" [33]. In all cases, n is fixed to be large enough that the matrix exponential is applied accurately. The 90th percentile of the relative errors are reported in Figure 5.1, i.e., errors were better than reported in the figure in 90% of the trials.

The results for the first four parameter choices are shown in the left panel of Figure 5.1 and illustrate the benefit of our algorithm over approaches based solely on low-rank approximation or quadratic trace estimation. Parameter choice (i) is a low-rank approximation type approach with $m = 0$. This produces a highly accurate approximation for large β, but a very inaccurate approximation for smaller β.

Table 5.1

Choices of parameters for Figure 5.1. *In the second-to-last column, an entry such as* $1200 + 300 = 1500$ *indicates that* 1200 *matvecs were devoted to deflation and* 300 *to estimating the trace of the remainder, for a total of* 1500.

	r	b	q	m	n	# matvecs	cols($\bar{\mathbf{Q}}_{q+1}$)
(i)	0	8	30	0	50	$640 + 0 = 640$	248
(ii)	0	0	0	13	50	$0 + 650 = 650$	0
(iii)	0	8	30	13	50	$640 + 650 = 1290$	248
(iv)	0	4	30	6	50	$320 + 300 = 650$	124
(v)	0	4	10	6	50	$240 + 300 = 540$	44
(vi)	2	4	10	6	50	$720 + 300 = 1020$	44
(vii)	4	4	10	6	50	$1200 + 300 = 1500$	44

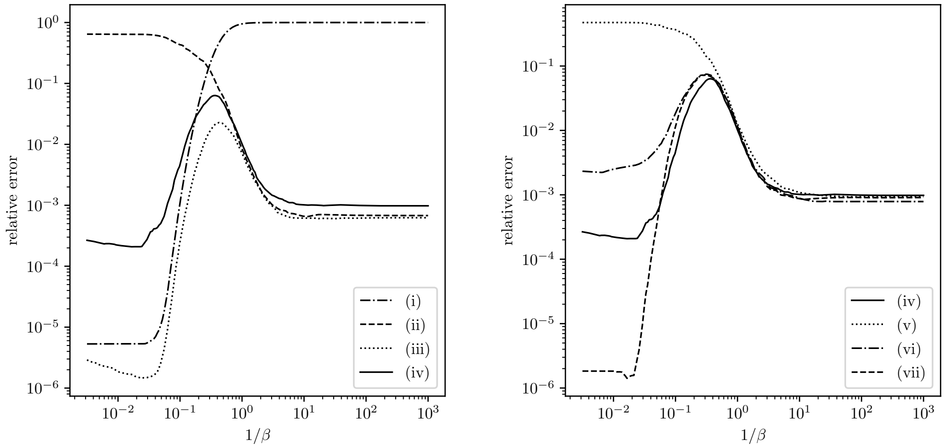

FIG. 5.1. *90th percentile of relative errors for Algorithm* 3.1 *(left) and Algorithm* 4.2 *(right) used to estimate* $\operatorname{tr}(\exp(-\beta\mathbf{A}))$ *for a spin chain. Curve* (i) *corresponds to low-rank approximation only, curve* (ii) *corresponds to quadratic trace estimation only, curve* (iii) *corresponds to a combination whose total cost is the cost of* (i)+(ii)*, and curve* (iv) *corresponds to a combination whose total cost is the same as each of* (i) *and* (ii)*. Curve* (v) *corresponds to zero restarts, curve* (vi) *to two restarts, and curve* (vii) *to four restarts. Exact parameter choices are given in Table* 5.1*.*

Parameter choice (ii) is a pure quadratic trace estimation approach with $b = q = 0$. This performs most accurately at small β but less accurately at large β. The next two parameter choices combine both approaches. Parameter choice (iii) uses the same nonzero values of q, b, and m from choices (i) and (ii) and, unsurprisingly, performs better than the first two approaches. Of course, the total number of matvecs is the sum of the first two approaches. Parameter choice (iv) is around the same cost as the first two approaches, and while it performs somewhat worse than either of the original approaches at extreme values of β, it performs better than these approaches for intermediate values of β. More importantly, the approach produces an approximation which is more uniformly good over the entire range of β encountered.

We remark that the fact that the quality of the approximations improve as $\beta \to 0$ is due to the normalization. If we do not use this normalization factor, then the error stagnates for β small. See [12, section 4.2] for more experiments regarding the use of normalization.

The results for the final three parameter choices are shown in the right panel of Figure 5.1 and illustrate the potential effectiveness of restarting. We take the filter polynomials to be the Chebyshev interpolants to $x \mapsto \exp(-\beta_0 x)$ on an interval containing the eigenvalues of $\bar{\mathbf{T}}_q$, where β_0 is the largest of the β used in the experiment. As expected, restarting allows a better low-rank approximation while limiting the storage and organization costs. With sufficient restarts, we are able to attain an approximation of quality comparable or even better to the results of parameter choice (iv).

5.2. Parameter selection for the inverse. In this example, we explore how the block size b, Krylov depth q, and number of restart cycles r impacts the quality of the basis $\bar{\mathbf{Q}}_{q+1}$. Specifically, we compute

$$\|(\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}})f(\mathbf{A})(\mathbf{I} - \bar{\mathbf{Q}}_{q+1}\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}})\|_{\mathsf{F}}/\|f(\mathbf{A})\|_{\mathsf{F}}. \tag{5.1}$$

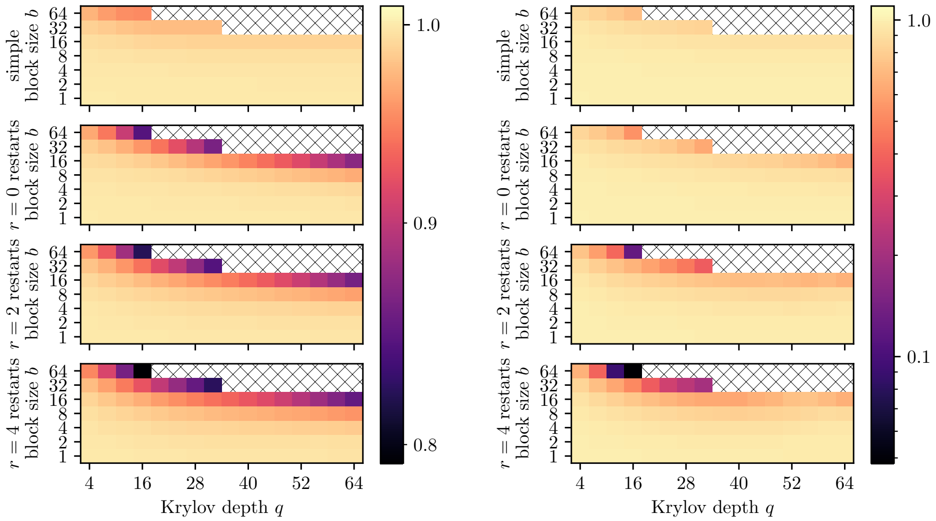

FIG. 5.2. *Quality of projection subspace* $\bar{\mathbf{Q}}_{q+1}$ *as in* (5.1) *(smaller is better) for* $f(\mathbf{A})$ *with algebraic decay* (5.2) *with* $n = 2000$ *and* $\kappa = 1000$. *Values for which* $qb > 1024$ *were not computed. For reference, we also show the space obtained by Algorithm* 2.3 *as "simple."*

Assuming n is at least large enough that $f(\mathbf{A})$ is applied accurately, this quantity is directly proportional to the variance of the quadratic trace estimator used in the second stage of our algorithm.

In order to test our algorithm, we use $f(x) = 1/x$ and choose two spectra defined, for $i = 1, 2, \ldots, d$, by

$$(5.2) \qquad f(\lambda_i^{\text{slow}}) = 1 + \left(\frac{i-1}{d-1}\right)^2 (\kappa - 1), \qquad f(\lambda_i^{\text{fast}}) = 1 + \left(\frac{i-1}{d-1}\right)(\kappa - 1)\rho^{d-i}.$$

These, respectively, correspond to algebraic and geometric decay in the eigenvalues of the *matrix function* $f(\mathbf{A})$. We then use the first stage of Algorithms 3.1 and 4.2 to compute $\bar{\mathbf{Q}}_{q+1}$ for a range of q, b, and r.

The results of our experiments are reported in Figure 5.2. As expected, when q, b, and r are larger, the quality of the approximation improves. In the case of slow (algebraic) decay, restarting is only mildly effective due to the fact that there are many eigenvalues with similar magnitude to the top eigenvalues. Moreover, as expected, the reduction in the value of (5.1) is not substantial. On the other hand, in the case of fast (geometric) decay, restarting allows higher quality approximations.

In addition, we show the quality of the projection used by Algorithm 2.3, which is essentially Hutch++ with matvec products with $f(\mathbf{A})$ computed via a black-box Krylov subspace method. As expected, after q becomes sufficiently large such that the Lanczos approximation to $f(\mathbf{A})\mathbf{\Omega}$ is reasonably accurate, this approach no longer improves with q. Moreover, for any fixed values of q and b, the approximation is worse than than our algorithm, due to the fact that the projection space is a strict subspace of $\bar{\mathbf{Q}}_{q+1}$.

5.3. Adaptive algorithm. We test Algorithm 4.1 on several problems, one with synthetic data and two using matrices from the SuiteSparse Matrix Collection [9]. On each of these problems we compare its performance to that of A-Hutch++ [43]. These algorithms differ only in their method for producing a set of vectors $\bar{\mathbf{Q}}_{q+1}$ for deflation and estimating $\operatorname{tr}(\bar{\mathbf{Q}}_{q+1}^{\mathsf{T}} f(\mathbf{A}) \bar{\mathbf{Q}}_{q+1})$; in our implementation, the procedures for estimating the remainder (lines 5–15 of Algorithm 4.1) are identical. Thus for a fixed input (ϵ,δ) we expect the algorithms to have similar accuracy. The primary differences will be the number of vectors devoted to deflation and the cost of the deflation step.

5.3.1. Estrada index. For our first test case we estimated the Estrada index $\mathrm{tr}(\exp(\mathbf{A}))$ of the Roget's Thesaurus graph, a 1022×1022 matrix with 7297 nonzero elements after symmetrization. We ran the adaptive algorithms to a relative error of 2^{-p} for $2\leq p\leq 7$ with failure probability $\delta=0.05$ and with 100 trials for each value of p. Matvecrix-vector products with $\exp(\mathbf{A})$ were estimated using $n=30$ steps of the Lanczos process, and the block-Lanczos routine in Algorithm 4.1 used block size $b=2$.

Results are shown in Figure 5.3 and Table 5.2. From the table in particular, we make the following observations:

- Algorithm 4.1 used less than half as many matvecs as A-Hutch++ for $p=2$. As the desired relative error decreased, our algorithm's comparative advantage increased to nearly a factor of 7 for $p=7$.
- Algorithm 4.1 used about 10 times as many vectors for deflation regardless of ϵ. Even so, it required many fewer matvecs for the deflation step.
- Because Algorithm 4.1 used more vectors for deflation, it did not need nearly as many samples to estimate the remainder.

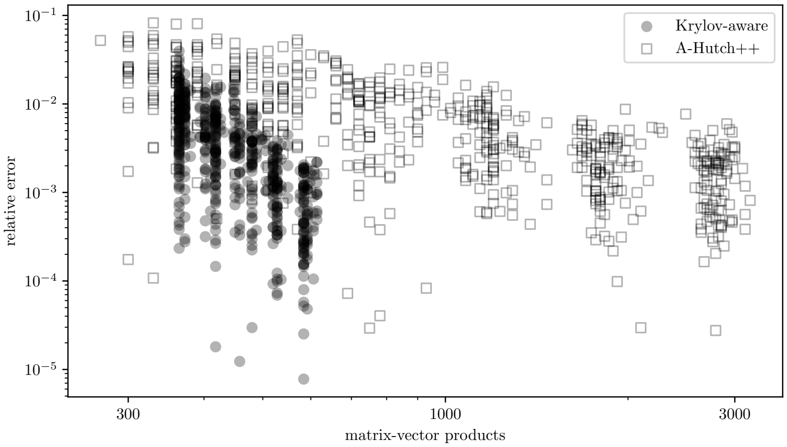

FIG. 5.3. *Costs associated with estimating* $\mathrm{tr}(\exp(\mathbf{A}))$ *for the Roget's Thesaurus graph.*

TABLE 5.2
Costs associated with estimating $\mathrm{tr}(\exp(\mathbf{A}))$ *for the Roget's Thesaurus graph. Values are the average over* 100 *trials rounded to the nearest integer.*

	Algorithm 4.1			A-Hutch++		
p	$\mathrm{cols}(\bar{\mathbf{Q}}_{q+1})$	m	# matvecs	$\mathrm{cols}(\bar{\mathbf{Q}}_{q+1})$	m	# matvecs
2	72	2	$304+60=364$	3	5	$208+160=368$
3	80	2	$312+74=386$	5	8	$278+253=531$
4	96	3	$328+93=421$	6	14	$390+408=798$
5	130	4	$362+106=469$	10	20	$609+586=1195$
6	174	4	$406+118=524$	15	31	$916+936=1851$
7	233	4	$465+125=590$	24	45	$1445+1354=2799$

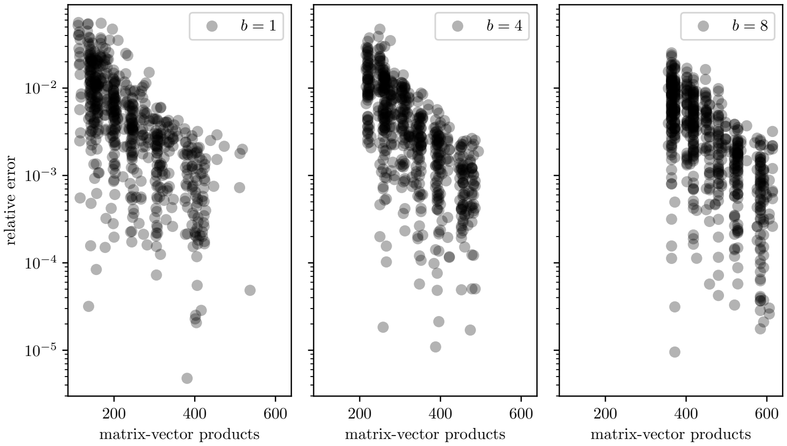

FIG. 5.4. *Costs associated with estimating* tr(exp($\mathbf{A}$)) *for the Roget's Thesaurus graph with different block sizes.*

TABLE 5.3
Costs associated with estimating tr(exp($\mathbf{A}$)) *for the Roget's Thesaurus graph with Algorithm* 4.1 *using different block sizes. The number of vectors used for deflation is* $\text{cols}(\bar{\mathbf{Q}}_{q+1}) = (q+1)b$. *Values are the average over* 100 *trials rounded to the nearest integer.*

	$b = 1$			$b = 4$			$b = 8$		
p	$(q+1)b$	m	# mv	$(q+1)b$	m	# mv	$(q+1)b$	m	# mv
2	23	3	140	44	2	230	72	2	364
3	36	3	163	56	3	261	81	2	386
4	51	4	202	73	4	298	96	3	421
5	92	4	253	113	4	342	130	4	469
6	124	5	316	157	4	398	174	4	523
7	160	7	408	216	4	465	233	4	589

With the caveat that counting matvecs alone does not account for the cost of orthogonalizing and storing $\bar{\mathbf{Q}}_{q+1}$, it is apparent that drawing the deflation vectors from a block-Krylov space and taking advantage of that structure has the potential to greatly reduce the cost of trace estimation problems.

We also ran the same set of experiments using block sizes $b \in \{1, 4, 8\}$. Results are shown in Figure 5.4 and Table 5.3. Our algorithm (unsurprisingly) used fewer matvecs when the block size was smaller, although the difference became less pronounced for smaller error tolerances ϵ. Smaller block sizes also tended to use fewer vectors for deflation. So at least to the extent that the number of matvecs is a reasonable proxy for the computational cost, we recommend using smaller block sizes over larger ones.

5.3.2. Log determinant. For our second test case we estimated the log determinant $\text{tr}(\log(\mathbf{A}))$ of the matrix `thermomech_TC`, a 102158×102158 matrix with 711558 nonzero elements. We ran the adaptive algorithms to a relative error of 2^{-p} for $5 \leq p \leq 9$ with failure probability $\delta = 0.05$ and with 50 trials for each value of p. Matvec products with $\log(\mathbf{A})$ were estimated using $n = 35$ steps of the Lanczos process, and the block-Lanczos routine in Algorithm 4.1 used block size $b = 2$. Figure 5.5 shows the relative error and cost in matvecs from each individual trial, and Table 5.4 shows the average cost over the trials for each value of p.

The `thermomech_TC` matrix is very well-conditioned with $\kappa(\mathbf{A}) \approx 70$, so not many samples are needed to get an accurate estimate and the variance reduction step is not

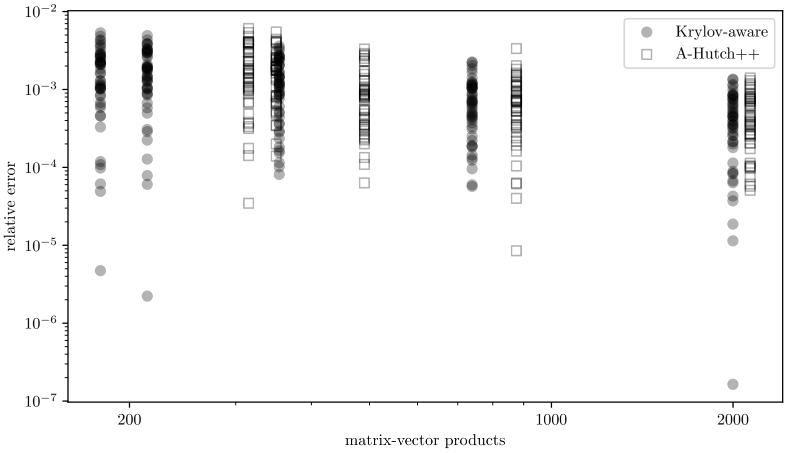

FIG. 5.5. *Costs asscoiated with estimating* $\operatorname{tr}(\log(\mathbf{A}))$ *for thermomechTC.*

TABLE 5.4

Costs associated with estimating $\operatorname{tr}(\log(\mathbf{A}))$ *for thermomechTC. Values are the average over* 50 *trials rounded to the nearest integer.*

	Algorithm 4.1			A-Hutch++		
p	cols($\bar{\mathbf{Q}}_{q+1}$)	m	# matvecs	cols($\bar{\mathbf{Q}}_{q+1}$)	m	# matvecs
5	6	3	$74+105=179$	3	3	$210+105=315$
6	6	4	$74+140=214$	3	4	$210+140=350$
7	6	8	$74+280=354$	3	8	$210+280=490$
8	6	19	$74+665=739$	3	19	$210+665=875$
9	6	55	$74+1925=1999$	3	55	$210+1925=2135$

particularly useful. Consequently, Algorithm 4.1 and A-Hutch++ have very similar performances Both terminate the variance reduction step almost immediately; the only difference is that the overhead cost is lower for our method.

5.3.3. Nuclear norm. For our final test case we estimated $\operatorname{tr}(\mathbf{A}^{1/2})$ of a synthetic 2500×2500 matrix $\mathbf{A} = \operatorname{diag}(1^{-c}, 2^{-c}, \ldots, 2500^{-c})$, with $c = 1.5$. We ran the adaptive algorithms to a relative error of 2^{-p} for $2 \leq p \leq 7$ with failure probability $\delta = 0.05$ and with 100 trials for each value of p. Matvec products with $\mathbf{A}^{1/2}$ were estimated using $n = 50$ steps of the Lanczos process, and the block-Lanczos routine in Algorithm 4.1 used block size $b = 2$.

Results are shown in Figure 5.6 and Table 5.5. The general pattern is similar to that of our experiments for the Roget's Thesaurus graph in subsection 5.3.1: Algorithm 4.1 devoted more vectors to deflation (about 20 times as many as A-Hutch++) and fewer toward estimating the remainder. It was more efficient overall, and the improvement increased from about a factor of 2 for $p = 2$ to a factor of 7 for $p = 7$.

6. Proofs.

In this section, we provide proofs of the theoretical results stated above.

Proof of Lemma 2.1. Note that $\bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T}$ is the orthogonal projector onto $\mathcal{K}_q(\mathbf{A}, \mathbf{Z})$, and for any $j \leq q-1$, $\mathbf{A}^j \mathbf{Z} \in \mathcal{K}_q(\mathbf{A}, \mathbf{Z})$. Thus, for any $k \leq q-1$,

$$\begin{aligned}\mathbf{A}^k \mathbf{Z} = \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{A}^k \mathbf{Z} &= \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{A} \mathbf{A}^{k-1} \mathbf{Z} \\ &= \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{A} \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{A}^{k-1} \mathbf{Z} \\ &= \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{A} \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{A} \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \cdots \bar{\mathbf{Q}}_q \bar{\mathbf{Q}}_q^\mathsf{T} \mathbf{Z}.\end{aligned}$$

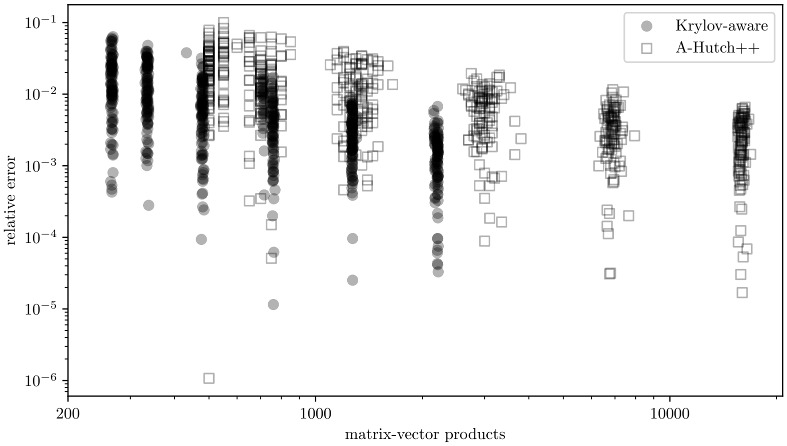

FIG. 5.6. *Estimating* $\operatorname{tr}(\mathbf{A}^{1/2})$ *for a synthetic matrix.*

TABLE 5.5

Costs associated with estimating $\operatorname{tr}(\mathbf{A}^{1/2})$ *for a synthetic matrix. Values are the average over* 100 *trials rounded to the nearest integer.*

	Algorithm 4.1			A-Hutch++		
p	$\operatorname{cols}(\bar{\mathbf{Q}}_{q+1})$	m	# matvecs	$\operatorname{cols}(\bar{\mathbf{Q}}_{q+1})$	m	# matvecs
2	18	3	$116+150=266$	3	4	$300+216=516$
3	37	4	$135+200=335$	3	8	$300+419=719$
4	82	6	$180+300=479$	4	19	$360+962=1322$
5	162	10	$260+488=747$	7	46	$726+2286=3012$
6	320	17	$418+852=1270$	18	102	$1801+5080=6881$
7	816	26	$914+1286=2199$	44	232	$4354+11587=15941$

From (2.5) and the orthogonality of $\bar{\mathbf{Q}}_{q+1}$, we have that $\bar{\mathbf{Q}}_q^\mathsf{T}\mathbf{A}\bar{\mathbf{Q}}_q = \mathbf{T}_q$. Thus since $\bar{\mathbf{Q}}_q^\mathsf{T}\mathbf{Z} = \mathbf{E}_1\mathbf{R}_1$, we find

$$\mathbf{A}^j\mathbf{Z} = \bar{\mathbf{Q}}_q[\mathbf{T}_q^j]_{:,1:b}\mathbf{R}_1.$$

By linearity, (2.6) is exact for any f that is a polynomial of degree up to $q-1$.

As for (2.7), let $k \leq 2q-1$ be partitioned as $k = 2j+\ell$, where $j \leq q-1$ and $\ell \in \{0,1\}$. Recalling that $\bar{\mathbf{Q}}_q^\mathsf{T}\mathbf{A}\bar{\mathbf{Q}}_q = \mathbf{T}_q$, the previous result implies that

$$\begin{aligned}
\mathbf{Z}^\mathsf{T}\mathbf{A}^k\mathbf{Z} &= (\mathbf{A}^j\mathbf{Z})^\mathsf{T}\mathbf{A}^\ell(\mathbf{A}^j\mathbf{Z}) \\
&= (\bar{\mathbf{Q}}_q(\mathbf{T}_q)^j\mathbf{E}_1\mathbf{R}_1)^\mathsf{T}\mathbf{A}^\ell(\bar{\mathbf{Q}}_q(\mathbf{T}_q)^j\mathbf{E}_1\mathbf{R}_1) \\
&= ((\mathbf{T}_q)^j\mathbf{E}_1\mathbf{R}_1)^\mathsf{T}(\bar{\mathbf{Q}}_q^\mathsf{T}\mathbf{A}^\ell\bar{\mathbf{Q}}_q)((\mathbf{T}_q)^j\mathbf{E}_1\mathbf{R}_1) \\
&= \mathbf{R}_1^\mathsf{T}\mathbf{E}_1^\mathsf{T}(\mathbf{T}_q)^{2j+\ell}\mathbf{E}_1\mathbf{R}_1. \\
&= \mathbf{R}_1^\mathsf{T}\mathbf{E}_1^\mathsf{T}(\mathbf{T}_q)^{k}\mathbf{E}_1\mathbf{R}_1.
\end{aligned}$$

Again, by linearity, (2.7) is exact for any f that is a polynomial of degree up to $2q-1$. ∎

Proof of Lemma 2.2. Given $\mathbf{A}$ and $\mathbf{Z}$, define for convenience

$$\mathsf{lan}_q(f) \equiv \bar{\mathbf{Q}}_q\,[f(\mathbf{T}_q)]_{:,1:b}\,\mathbf{R}_1$$

as the approximation (2.6) to $f(\mathbf{A})\mathbf{Z}$. Let p be any polynomial with $\deg(p) \leq q-1$. Then from the first part of Lemma 2.1, the equality $\|\mathbf{R}_1\| = \|\mathbf{Z}\|$, and the fact that the eigenvalues of $\mathbf{T}_q$ interlace those of $\mathbf{A}$, it holds that

$$\begin{aligned}\|f(\mathbf{A})\mathbf{Z} - \mathsf{lan}_q(f)\|_2 &\leq \|f(\mathbf{A})\mathbf{Z} - p(\mathbf{A})\mathbf{Z}\|_2 + \|\mathsf{lan}_q(p) - \mathsf{lan}_q(f)\|_2 \\ &\leq \|\mathbf{Z}\|_2\left(\|f(\mathbf{A}) - p(\mathbf{A})\|_2 + \|f(\mathbf{T}_q) - p(\mathbf{T}_q)\|_2\right) \\ &= \|\mathbf{Z}\|_2\left(\max_{x\in\Lambda(\mathbf{A})}|f(x)-p(x)| + \max_{x\in\Lambda(\mathbf{T}_q)}|f(x)-p(x)|\right) \\ &\leq 2\|\mathbf{Z}\|_2 \max_{x\in[\lambda_{\min},\lambda_{\max}]}|f(x)-p(x)|.\end{aligned}$$

The first result follows by optimizing over polynomials p. The second result is proved in a similar fashion using the second part of Lemma 2.1. ∎

Proof of Lemma 2.3. By the triangle inequality and basic properties of the Frobenius norm,

$$\begin{aligned}\|\mathbf{P_Q}f(\mathbf{A})\mathbf{P_Q} - \mathbf{P_V}f(\mathbf{A})\mathbf{P_V}\|_{\mathsf{F}} &\leq \|\mathbf{P_Q}f(\mathbf{A})\mathbf{P_Q} - \mathbf{P_Q}f(\mathbf{A})\mathbf{P_V}\|_{\mathsf{F}} \\ &\quad + \|\mathbf{P_Q}f(\mathbf{A})\mathbf{P_V} - \mathbf{P_V}f(\mathbf{A})\mathbf{P_V}\|_{\mathsf{F}} \\ &\leq 2\|f(\mathbf{A})\|_2\|\mathbf{P_Q} - \mathbf{P_V}\|_{\mathsf{F}}.\end{aligned}$$

Using the assumption that $\mathbf{V}$ and $\mathbf{Q}$ are of equal rank, [51, Theorems 2.3, 2.4] assert that

$$\|\mathbf{P_Q} - \mathbf{P_V}\|_2 \leq \|(f(\mathbf{A})\mathbf{\Omega})^\dagger\|_2\|f(\mathbf{A})\mathbf{\Omega} - \mathsf{lan}_q(f)\|_2.$$

Since $f(\mathbf{A})$ is square, $\sigma_{\min}(f(\mathbf{A})\mathbf{\Omega}) \geq \sigma_{\min}(\mathbf{\Omega})\sigma_{\min}(f(\mathbf{A}))$. Thus,

$$\|(f(\mathbf{A})\mathbf{\Omega})^\dagger\| = \sigma_{\min}(f(\mathbf{A})\mathbf{\Omega})^{-1} \leq \sigma_{\min}(\mathbf{\Omega})^{-1}\sigma_{\min}(f(\mathbf{A}))^{-1}.$$

Using this and the fact that $\operatorname{rank}(\mathbf{P_Q} - \mathbf{P_V}) \leq 2b$,

$$\begin{aligned}\|\mathbf{P_Q}f(\mathbf{A})\mathbf{P_Q} - \mathbf{P_V}f(\mathbf{A})\mathbf{P_V}\|_{\mathsf{F}} &\leq 2\|f(\mathbf{A})\|_2\|\mathbf{P_Q} - \mathbf{P_V}\|_{\mathsf{F}} \\ &\leq 2\sqrt{2b}\|f(\mathbf{A})\|_2\|\mathbf{P_Q} - \mathbf{P_V}\|_2 \\ &\leq 2\sqrt{2b}\|f(\mathbf{A})\|_2\|(f(\mathbf{A})\mathbf{\Omega})^\dagger\|_2\|f(\mathbf{A})\mathbf{\Omega} - \mathsf{lan}_q(f)\|_2 \\ &\leq 2\sqrt{2b}\,\frac{\sigma_{\max}(f(\mathbf{A}))}{\sigma_{\min}(f(\mathbf{A}))}\frac{\sigma_{\max}(\mathbf{\Omega})}{\sigma_{\min}(\mathbf{\Omega})}\Delta.\end{aligned}$$ ∎

Proof of Lemma 3.1. The proof is essentially identical to the proof of Lemma 2.1 after relabeling $\bar{\mathbf{Q}}_q \to \bar{\mathbf{Q}}_{q+n+1}$, $\mathbf{Z} \to \bar{\mathbf{Q}}_q$, and $q \to n$. Since $\bar{\mathbf{Q}}_q$ has orthogonal columns, the analogue of $\mathbf{R}_1$ from Lemma 2.1 is just the identity.

Proof of Theorem 3.2. Define

$$\mathsf{est} \equiv \operatorname{tr}(\hat{\mathbf{Q}}^\mathsf{T} f(\mathbf{A})\hat{\mathbf{Q}}) + \frac{d-\hat{b}}{m}\sum_{i=1}^{m}\frac{\mathbf{y}_i^\mathsf{T} f(\mathbf{A})\mathbf{y}_i}{\mathbf{y}_i^\mathsf{T}\mathbf{y}_i}.$$

Recall that $\mathbf{y}_i = (\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^\mathsf{T})\boldsymbol{\psi}_i$, where $\boldsymbol{\psi}_i$ is a Gaussian vector, so $\mathbf{y}_i/\|\mathbf{y}_i\|_2$ has a uniform distribution on the unit hypersphere defined on the complement of the column span of $\hat{\mathbf{Q}}$ which has dimension $d-\hat{b}$. Thus $\mathbb{E}[\mathsf{est}] = \operatorname{tr}(f(\mathbf{A}))$, and by the law of total variance and the variance formula for trace estimation using random vectors from the real hypersphere given in [21],

$$\begin{aligned}\mathbb{V}\big[\mathsf{est}\big] &= \mathbb{E}\big[\mathbb{V}[\mathsf{est}|\boldsymbol{\Omega}]\big] + \mathbb{V}\big[\mathbb{E}[\mathsf{est}|\boldsymbol{\Omega}]\big] \\ &= \mathbb{E}\left[\frac{2(d-\hat{b})}{m(d-\hat{b}+2)}\left(\|\mathbf{F}\|_{\mathsf{F}}^2 - \frac{\operatorname{tr}(\mathbf{F})^2}{d-\hat{b}}\right)\right] + \underbrace{\mathbb{V}\big[\operatorname{tr}(f(\mathbf{A}))\big]}_{=0} \\ &= \frac{2(d-\hat{b})}{m(d-\hat{b}+2)}\left(\mathbb{E}\big[\|\mathbf{F}\|_{\mathsf{F}}^2\big] - \frac{\mathbb{E}\big[\operatorname{tr}(\mathbf{F})^2\big]}{d-\hat{b}}\right).\end{aligned}$$

Algorithms 2.3 and 3.1 compute the quantities

$$t_{\text{defl}} = \operatorname{tr}\left([f(\mathbf{T}_{q+n})]_{1:\hat{b},1:\hat{b}}\right), \qquad t_{\text{rem}} = \frac{d-\hat{b}}{m}\sum_{i=1}^{m}[f(\mathbf{T}_n^{(i)})]_{1,1},$$

which, respectively, approximate the first and second terms of est. Since $\hat{\mathbf{Q}}$ has orthonormal columns, $\|\hat{\mathbf{Q}}\|_2^2 = 1$. Thus, by the definition of Δ,

$$\left\|\hat{\mathbf{Q}}^{\mathsf{T}} f(\mathbf{A})\hat{\mathbf{Q}} - [f(\mathbf{T}_{q+n})]_{1:\hat{b},1:\hat{b}}\right\|_2 \leq \Delta, \qquad \left|\frac{\mathbf{y}_i^{\mathsf{T}} f(\mathbf{A})\mathbf{y}_i}{\mathbf{y}_i^{\mathsf{T}}\mathbf{y}_i} - [f(\mathbf{T}_n^{(i)})]_{1,1}\right| \leq \Delta.$$

Using the fact that $|\operatorname{tr}(\mathbf{E})| \leq \hat{b}\,\|\mathbf{E}\|_2$ for any $\mathbf{E} \in \mathbb{R}^{\hat{b}\times\hat{b}}$,

$$|\mathsf{est} - (t_{\text{defl}} + t_{\text{rem}})| \leq \hat{b}\,\Delta + (d-\hat{b})\Delta = d\,\Delta.$$

Taking expected values gives the bound for the expectation.

We can bound the variance by

$$\mathbb{V}\big[t_{\text{defl}} + t_{\text{rem}}\big] \leq \left(\sqrt{\mathbb{V}\big[\mathsf{est}\big]} + \sqrt{\mathbb{V}\big[\mathsf{est} - (t_{\text{defl}} + t_{\text{rem}})\big]}\right)^2.$$

Since $\mathbb{V}\big[\mathsf{est} - (t_{\text{defl}} + t_{\text{rem}})\big] \leq \mathbb{E}\big[|\mathsf{est} - (t_{\text{defl}} + t_{\text{rem}})|^2\big] \leq \mathbb{E}\big[(d\,\Delta)^2\big] = d^2\,\mathbb{E}\big[\Delta^2\big]$, we get the variance bound.

Finally, if the same $\boldsymbol{\Omega}$ is used in both algorithms, the column span of $\mathbf{Q}$ is that of $\bar{\mathbf{Q}}_q\,[f(\mathbf{T}_q)]_{:,1:b}\,\mathbf{R}_1$, which is clearly contained in $\bar{\mathbf{Q}}_{q+1}$. Thus, $\mathbf{P}_{\bar{\mathbf{Q}}_q} = \mathbf{P}_{\bar{\mathbf{Q}}_q}\mathbf{P}_{\mathbf{Q}}$, so basic properties of the Frobenius norm, and the fact that $\|\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}}\|_2 \leq 1$, imply that

$$\|\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}} f(\mathbf{A})\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}}\|_{\mathsf{F}}^2 = \|\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}}\mathbf{P}_{\mathbf{Q}} f(\mathbf{A})\mathbf{P}_{\mathbf{Q}}\mathbf{P}_{\bar{\mathbf{Q}}_{q+1}}\|_{\mathsf{F}}^2 \leq \|\mathbf{P}_{\mathbf{Q}} f(\mathbf{A})\mathbf{P}_{\mathbf{Q}}\|_{\mathsf{F}}^2.$$

Taking expectations on both sides preserves this inequality. □

7. Conclusions and future work.

From our analysis and experiments it is clear that exploiting the structure of block-Krylov subspaces can significantly reduce the number of matvec products required for randomized trace estimation. The effect on the computational time is less clear, and depends both on the cost of maintaining the orthonormal basis $\bar{\mathbf{Q}}_q$ and on how efficiently matvecs with $\mathbf{A}$ can be computed in parallel. A high-quality practical implementation for large-scale problems will likely require further study in order to more effectively balance the true costs of the algorithm. For such a setting, we believe that the restarted variant (Algorithm 4.2), in particular, merits further study.

It is also worth considering the role of the approximation degree n in further detail. For example, if the deflation step has proved useful and significantly reduced the Frobenius norm of the remainder, then it may be possible to estimate the trace of the remainder using a smaller value of n. We are not aware of any formal results on this topic. Informally, the paper [28] proposes a trace estimator that uses multiple different degrees n, and in doing so hedges against overestimating n.

Acknowledgments. The authors thank the referees and editor for their comments which improved the presentation of the paper. We also thank David Persson for helpful discussions relating to the proof.

REFERENCES

[1] J. Baglama, D. Calvetti, and L. Reichel, *IRBL: An implicitly restarted block-Lanczos method for large-scale Hermitian eigenproblems*, SIAM J. Sci. Comput., 24 (2003), pp. 1650–1677.

[2] Z. Bai, G. Fahey, and G. Golub, *Some large-scale matrix computation problems*, J. Comput. Appl. Math., 74 (1996), pp. 71–89.

[3] Z. Bai and G. Golub, *Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices*, Ann. Numer. Math., 4 (1996), pp. 29–38.

[4] R. A. Baston and Y. Nakatsukasa, *Stochastic Diagonal Estimation: Probabilistic Bounds and an Improved Algorithm*, preprint, arXiv:2201.10684, 2022.

[5] C. Bekas, E. Kokiopoulou, and Y. Saad, *An estimator for the diagonal of a matrix*, Appl. Numer. Math., 57 (2007), pp. 1214–1229.

[6] T. Chen, A. Greenbaum, C. Musco, and C. Musco, *Error bounds for Lanczos-based matrix function approximation*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 787–811.

[7] T. Chen, T. Trogdon, and S. Ubaru, *Randomized Matrix-free Quadrature for Spectrum and Spectral Sum Approximation*, preprint, arXiv:2204.01941, 2022.

[8] A. Cortinovis and D. Kressner, *On randomized trace estimates for indefinite matrices with an application to determinants*, Found. Comput. Math., 22 (2022), pp. 875–903.

[9] T. A. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25.

[10] P. Drineas, I. C. F. Ipsen, E.-M. Kontopoulou, and M. Magdon-Ismail, *Structural convergence results for approximation of dominant subspaces from block Krylov spaces*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 567–586.

[11] V. L. Druskin and L. A. Knizhnerman, *Error bounds in the simple Lanczos procedure for computing functions of symmetric matrices and eigenvalues*, Comput. Math. Math. Phys., 31 (1991), pp. 20–30.

[12] E. N. Epperly, J. A. Tropp, and R. J. Webber, *Xtrace: Making the Most of Every Sample in Stochastic Trace Estimation*, preprint, arXiv:2301.07825, 2023.

[13] M. Faridfar and J. Vahedi, *Thermodynamic behavior of spin-*1 *Heisenberg chain: A comparative study*, J. Supercond. Novel Magn., 35 (2021), pp. 519–528.

[14] A. Frommer, S. Güttel, and M. Schweitzer, *Convergence of restarted Krylov subspace methods for Stieltjes functions of matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1602–1624.

[15] A. Frommer, K. Kahl, Th. Lippert, and H. Rittich, 2*-norm error bounds and estimates for Lanczos approximations to linear systems and rational matrix functions*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1046–1065.

[16] A. Frommer and M. Schweitzer, *Error bounds and estimates for Krylov subspace approximations of Stieltjes matrix functions*, BIT, 56 (2015), pp. 865–892.

[17] A. Frommer and V. Simoncini, *Matrix functions*, in Mathematics in Industry, Springer, Berlin, 2008, pp. 275–303.

[18] A. Frommer and V. Simoncini, *Stopping criteria for rational matrix functions of Hermitian and symmetric matrices*, SIAM J. Sci. Comput., 30 (2008), pp. 1387–1412.

[19] A. Frommer and V. Simoncini, *Error bounds for Lanczos approximations of rational functions of matrices*, in Numerical Validation in Current Hardware Architectures, Springer, Berlin, 2009, pp. 203–216.

[20] A. S. Gambhir, A. Stathopoulos, and K. Orginos, *Deflation as a method of variance reduction for estimating the trace of a matrix inverse*, SIAM J. Sci. Comput., 39 (2017), pp. A532–A558.

[21] D. Girard, *Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille*, Technical report 669-M, IMAG, Grenoble, France, 1987.

[22] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, NJ, 2009.

[23] G. H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software, Academic, New York, 1977, pp. 361–377.

[24] S. Gratton and D. Titley-Peloquin, *Improved bounds for small-sample estimation*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 922–931.

[25] S. Güttel and M. Schweitzer, *A comparison of limited-memory Krylov methods for Stieltjes functions of Hermitian matrices*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 83–107.

[26] N. Halko, P.-G. Martinsson, Y. Shkolnisky, and M. Tygert, *An algorithm for the principal component analysis of large data sets*, SIAM J. Sci. Comput., 33 (2011), pp. 2580–2594.

[27] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[28] E. Hallman and D. Troester, *A multilevel approach to stochastic trace estimation*, Linear Algebra Appl., 638 (2022), pp. 125–149.

[29] I. Han, D. Malioutov, H. Avron, and J. Shin, *Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations*, SIAM J. Sci. Comput., 39 (2017), pp. A1558–A1585.

[30] N. J. Higham, *Functions of Matrices*, SIAM, Philadelphia, 2008.

[31] M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Comm. Statist. Simulation Comput., 18 (1989), pp. 1059–1076.

[32] M. D. Ilic, I. W. Turner, and D. P. Simpson, *A restarted Lanczos approximation to functions of a symmetric matrix*, IMA J. Numer. Anal., 30 (2009), pp. 1044–1061.

[33] M. Karabach, G. Müller, H. Gould, and J. Tobochnik, *Introduction to the Bethe ansatz* I, Computers Phys., 11 (1997), p. 36–43.

[34] L. A. Knizhnerman, *The simple Lanczos procedure: Estimates of the error of the Gauss quadrature formula and their applications*, Comput. Math. Math. Phys., 36 (1996), pp. 1481–1492.

[35] M. Ledoux, *The Concentration of Measure Phenomenon,* Math. Surveys Monogr. 89, American Mathematical Society, Providence, RI, 2001.

[36] H. Li and Y. Zhu, *Randomized block Krylov subspace methods for trace and log-determinant estimators*, BIT, 61 (2021), pp. 911–939.

[37] L. Lin, *Randomized estimation of spectral densities of large matrices made accurate*, Numer. Math., 136 (2017), pp. 183–213.

[38] P.-G. Martinsson and J. A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572.

[39] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff, *Hutch++: Optimal stochastic trace estimation*, in Symposium on Simplicity in Algorithms (SOSA), SIAM, Philadelphia, 2021, pp. 142–155.

[40] K. Morita and T. Tohyama, *Finite-temperature properties of the Kitaev-Heisenberg models on kagome and triangular lattices studied by improved finite-temperature Lanczos methods*, Phys. Rev. Res., 2 (2020), 013205.

[41] C. Musco and C. Musco, *Randomized block Krylov methods for stronger and faster approximate singular value decomposition*, in Advances in Neural Information Processing Systems, Curran Associates, Red Hook, NY, 28 2015, pp. 1396–1404.

[42] C. Musco, C. Musco, and A. Sidford, *Stability of the Lanczos method for matrix function approximation*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2018, pp. 1605–1624.

[43] D. Persson, A. Cortinovis, and D. Kressner, *Improved variants of the Hutch++ algorithm for trace estimation*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 1162–1185.

[44] D. Persson and D. Kressner, *Randomized low-rank approximation of monotone matrix functions*, SIAM J. Matrix Anal. Appl. 44 (2023), pp. 894–918.

[45] S. Popescu, A. J. Short, and A. Winter, *Entanglement and the foundations of statistical mechanics*, Nature Phys., 2 (2006), pp. 754–758.

[46] F. Roosta-Khorasani, G. J. Székely, and U. M. Ascher, *Assessing stochastic algorithms for large scale nonlinear least squares problems using extremal probabilities of linear combinations of gamma random variables*, SIAM/ASA J. Uncertain. Quantif., 3 (2015), pp. 61–90.

[47] A. K. Saibaba, A. Alexanderian, and I. C. Ipsen, *Randomized matrix-free trace and log-determinant estimators*, Numer. Math., 137 (2017), pp. 353–395.

[48] J. Schnack, J. Richter, and R. Steinigeweg, *Accuracy of the finite-temperature Lanczos method compared to simple typicality-based estimates*, Phys. Rev. Res., 2 (2020), 013186.

[49] J. Skilling, *The eigenvalues of mega-dimensional matrices*, in Maximum Entropy and Bayesian Methods, Springer, Dordrecht, Netherlands, 1989, pp. 455–466.

[50] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[51] G. W. Stewart, *On the perturbation of pseudo-inverses, projections and linear least squares problems*, SIAM Rev., 19 (1977), pp. 634–662.

[52] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 601–614.

[53] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*, SIAM, Philadelphia, 2019.

[54] S. Ubaru, J. Chen, and Y. Saad, *Fast estimation of* $\operatorname{tr}(f(A))$ *via stochastic Lanczos quadrature*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1075–1099.

[55] S. Ubaru and Y. Saad, *Applications of trace estimation techniques*, in International Conference on High Performance Computing in Science and Engineering, Springer, Cham, Switzerland, 2017, pp. 19–33.

[56] R. Vershynin, *Introduction to the non-asymptotic analysis of random matrices*, in Compressed Sensing, Cambridge University Press, Cambridge, 2012, pp. 210–268.

[57] S. Wang, Z. Zhang, and T. Zhang, *Improved Analyses of the Randomized Power Method and Block Lanczos Method*, preprint, arXiv:1508.06429, 2015.

[58] A. Weisse, G. Wellein, A. Alvermann, and H. Fehske, *The kernel polynomial method*, Rev. Mod. Phys., 78 (2006), pp. 275–306.

[59] K. Wu and H. Simon, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.

[60] L. Wu, J. Laeuchli, V. Kalantzis, A. Stathopoulos, and E. Gallopoulos, *Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse*, J. Comput. Phys., 326 (2016), pp. 828–844.

[61] Q. Xu and T. Chen, *A Posteriori Error Bounds for the Block-Lanczos Method for Matrix Function Approximation*, preprint, arXiv:2211.15643, 2022.

[62] Y. Zhou and Y. Saad, *Block Krylov-Schur method for large symmetric eigenvalue problems*, Numer. Algorithms, 47 (2008), pp. 341–359.