



Estimation of spectral gaps for sparse symmetric matrices

Michele Benzi¹ · Michele Rinelli² · Igor Simunec³

Received: 22 October 2024 / Revised: 8 September 2025 / Accepted: 17 January 2026 /
Published online: 26 February 2026
© The Author(s) 2026, modified publication 2026

Abstract

In this paper we propose and analyze an algorithm for identifying spectral gaps of a real symmetric matrix A by simultaneously approximating the traces of spectral projectors associated with multiple different spectral slices. Our method utilizes Hutchinson’s stochastic trace estimator together with the Lanczos algorithm to approximate quadratic forms involving spectral projectors. Instead of focusing on determining the gap between two particular consecutive eigenvalues of A , we aim to find all gaps that are wider than a specified threshold. By examining the problem from this perspective, and thoroughly analyzing both the Hutchinson and the Lanczos components of the algorithm, we obtain error bounds that allow us to determine the number of Hutchinson’s sample vectors and Lanczos iterations needed to ensure the detection of all gaps above the target width with high probability. In particular, we conclude that the most efficient strategy is to always use a single random sample vector for Hutchinson’s estimator and concentrate all computational effort in the Lanczos algorithm. Our numerical experiments demonstrate the efficiency and reliability of this approach.

Mathematics Subject Classification 65F50 · 65F15

1 Introduction

We consider the problem of locating gaps in the spectrum of an $n \times n$ real symmetric matrix A , i.e., finding intervals that do not contain any eigenvalue of A . A variant of this problem that is more commonly encountered in the literature is the following:

✉ Michele Benzi
michele.benzi@sns.it

Michele Rinelli
michele.rinelli@kuleuven.be

Igor Simunec
igor.simunec@epfl.ch

¹ Faculty of Science, Scuola Normale Superiore, Piazza dei Cavalieri, 7, 56126 Pisa, Italy

² Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

³ Institute of Mathematics, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

given an integer k with $1 \leq k < n$, find a real number μ such that exactly k of the eigenvalues of A (counted with their multiplicities) are strictly below μ . This problem can be equivalently restated as finding μ such that the spectral projector P_μ onto the subspace spanned by the eigenvectors of A associated with eigenvalues strictly smaller than μ satisfies $\text{rank}(P_\mu) = k$; since for a projector the rank is equal to the trace, we can also write this requirement as $\text{tr}(P_\mu) = k$. If we denote the eigenvalues of A by λ_i and we label them in non-decreasing order, this problem has a solution only if $\lambda_k < \lambda_{k+1}$ (non-degeneracy assumption). In this case, any μ lying in the open interval $(\lambda_k, \lambda_{k+1})$ is a solution. Instead of looking for the gap for a specific value of k , the approach that we present here aims to find all gaps in the spectrum with width above a certain threshold, and in addition we obtain an estimate of the number of eigenvalues below each gap. In practical cases where it is of interest to find μ in the gap between λ_k and λ_{k+1} , the gap $(\lambda_k, \lambda_{k+1})$ is often relatively large, so we expect to be able to find it by looking for all gaps above a certain width.

There are several applications where it is required to locate gaps in the spectrum of a given symmetric matrix. In Kohn–Sham Density Functional Theory for electronic structure computations [1], at each step of a self-consistent field iteration it is necessary to compute either the eigenvectors of a symmetric matrix (the linearized discrete Kohn–Sham Hamiltonian) associated with eigenvalues below the *Fermi level* (or *chemical potential*, usually denoted by μ), corresponding to the so-called *occupied states* of the system, or equivalently the corresponding spectral projector P_μ . In the case of insulators at zero electronic temperature, the linearized Hamiltonians exhibit a gap separating the first k eigenvalues from the rest of the spectrum, where k is the number of electrons, and locating this gap is equivalent to estimating the Fermi level μ . In particular, in linear scaling electronic structure computations the spectral projector is approximated either by polynomials or by rational functions that act as filters, i.e., approximate the step function $h_\mu(\lambda)$ that takes the value 1 for $\lambda < \mu$ and 0 for $\lambda > \mu$ (see, e.g., [2]). Clearly, in order to use this approach it is first necessary to estimate the Fermi level μ , which is a nontrivial task.

Another potential area of application arises in solid state physics, where the location of gaps in the energy spectrum of quantum-mechanical systems described by Schrödinger operators with periodic potentials is an important problem; see, e.g., [3], as well as [4].

An additional possible area of application is *spectral clustering*, which uses the eigenvectors of the graph Laplacian to partition a graph, see [5] and references therein. The stability of the partitioning under small perturbations of the weights is governed by the gaps in the spectrum of the graph Laplacian; in particular, relatively large gaps correspond to stable partitions. Hence, identifying sufficiently large gaps in the spectrum of the Laplacian matrix is an important preliminary step for any spectral clustering algorithm.

Finally, as a further application we mention estimating the stability of invariant subspaces under perturbations. It is well known in numerical linear algebra that individual eigenvectors of symmetric matrices corresponding to a cluster of tightly spaced eigenvalues will be highly ill-conditioned. On the other hand, if the cluster is well-separated from the rest of the spectrum, the invariant subspace spanned by those eigenvectors is well-conditioned and can be determined with high accuracy; furthermore, iterative

methods designed to approximate the invariant subspace (or, equivalently, the orthogonal projector onto it) will typically converge rapidly. Hence, being able to efficiently identify gaps in the spectrum of a symmetric matrix A provides a way to estimate the conditioning of invariant subspaces. Moreover, when A is banded, or sparse, the invariant space is “localized”, an important property that can reduce computational costs considerably; see, e.g., [6] and also [2, Sec. 11.4] and [7].

In the literature, a few approaches have been proposed to estimate values of μ lying within the target gap. It is well known that for any real symmetric matrix $A - \mu I$ there exists a unit lower triangular matrix L and a block diagonal matrix D with 1×1 and 2×2 blocks such that $A - \mu I = LDL^T$; see for example [8, Chap. 1.3.4]. Since A and D are congruent, they have the same inertia, i.e., the same number of positive, negative, and possibly zero eigenvalues (the latter can occur only when μ happens to be an eigenvalue of A , an extremely unlikely occurrence in practice). Hence, if A is not too large, an LDL^T factorization of $A - \mu I$ can be used to count the number of eigenvalues smaller than μ . If this number is less than k (larger than k) then μ is increased (respectively, decreased) and the procedure is repeated until the correct value is found. When A is sparse, one can make use of symmetric row and column permutations to reduce the fill-in in L (and therefore the arithmetic and storage costs) while preserving sparsity, see [9, Chap. 7]. A major drawback of this approach is its cost, and especially the fact that having computed the LDL^T factorization of $A - \mu I$ for a certain μ is of no help in computing the factorization for a different value of μ , hence using matrix factorizations in a trial-and-error fashion can be prohibitively expensive, even assuming that a factorization can be computed at all.

Different techniques have been used in the context of linear scaling methods. In these methods, an initial guess $\mu_0 \approx \mu$ is used in constructing a polynomial approximation to the projector P_{μ_0} by an iterative process known as *purification*, essentially a Newton-type method. The value of μ is adjusted in the course of the iterative process until the prescribed value k of the trace is obtained; hence, the determination of μ and the approximation of the projector P_μ are interweaved. The main cost in this approach is the need to repeatedly solve $n \times n$ linear systems with variable coefficient matrix at each iteration. We refer to [10, 11] for details on these adaptive procedures.

Yet another approach is the one that combines stochastic trace estimators with the Lanczos algorithm to compute an approximation of $\text{tr}(P_\mu)$. An advantage of this approach is that we can exploit the shift-invariance property of Krylov subspaces (see, e.g., [12]) to efficiently approximate the trace of the spectral projectors P_μ associated with many different values of μ at the same time. The combination of Hutchinson’s estimator and the Lanczos algorithm has also appeared in literature on related problems that exploit the trace of spectral projectors, such as the estimation of the number of eigenvalues of a matrix A contained in a given interval $[a, b]$ or the estimation of spectral densities, see, for instance, [13–16]. In the context of spectral density approximation, a similar approach is the kernel polynomial method, which was introduced in [17, 18] and uses a polynomial expansion based on Chebyshev polynomials.

Here we focus on the approach that combines Hutchinson’s trace estimator and the Lanczos algorithm to estimate the trace of the spectral projectors P_μ . When using this kind of approach, it is essential to have a criterion to choose the parameters of the algorithm, namely the number of random vectors x_i for Hutchinson’s estimator

and the number of Lanczos iterations, in order to minimize the computational cost and guarantee the correctness of the results. However, although there exist theoretical results on the convergence speed of this approach in the literature, they are mainly focused on the global approximation of the spectral density and they are not satisfactory when applied to gap detection; in particular, they are not useful to determine how the parameters should be selected. The main purpose of this work is to answer this question for the problem of finding gaps in the spectrum of a matrix. By changing the point of view from finding μ such that the number of eigenvalues below μ is exactly k to finding all the gaps in the spectrum with width above a certain threshold, we are able to provide a rigorous analysis that allows us to choose the input parameters to ensure that all the target gaps are found, and at the same time minimize the computational cost. Our analysis will determine that for this problem the most efficient choice is to use a single quadratic form $\mathbf{x}^T P_\mu \mathbf{x}$ to approximate $\text{tr}(P_\mu)$, but in order to reach this conclusion we first have to analyze the more general algorithm that uses Hutchinson's stochastic trace estimator. This analysis will result in an algorithm that simultaneously computes upper and lower bounds to $\mathbf{x}^T P_\mu \mathbf{x}$ for all considered values of μ , and exploits them to find intervals that contain no eigenvalues of A with high probability.

The rest of the paper is organized as follows. In Sect. 2 we describe a basic version of the algorithm that we are going to use to detect the gaps in the spectrum of A . Each component of the algorithm is then analyzed in detail in Sect. 3, with a focus on bounding the error. Then, in Sect. 4 we put together the analysis to determine the best choice of input parameters and we obtain an efficient and robust algorithm that can give guarantees on its output. In Sect. 5 we conduct some numerical experiments to evaluate the performance and reliability of our approach. Finally, Sect. 6 contains some concluding remarks.

2 Algorithm overview

We start by introducing some notation to establish the framework for our algorithm. Given $\mu \in \mathbb{R}$, let $h_\mu : \mathbb{R} \rightarrow \mathbb{R}$ denote the Heaviside function

$$h_\mu(x) = \begin{cases} 1 & \text{for } x < \mu, \\ 1/2 & \text{for } x = \mu, \\ 0 & \text{for } x > \mu. \end{cases}$$

The step function h_μ is closely related to the sign function, indeed we have $h_\mu(x) = -\frac{1}{2} \text{sign}(x - \mu) + \frac{1}{2}$. We denote by λ_i the eigenvalues of A in non-decreasing order, counted according to their multiplicities, and by $\{\mathbf{u}_i\}_{i=1}^n \subset \mathbb{R}^n$ an orthonormal basis of eigenvectors, where \mathbf{u}_i is an eigenvector associated with λ_i . Assuming that μ does not coincide with any of the eigenvalues of the matrix A , we can define the spectral projector

$$P_\mu := h_\mu(A) = \sum_{i: \lambda_i < \mu} \mathbf{u}_i \mathbf{u}_i^T.$$

Note that although we can define $h_\mu(A)$ for all μ , since $h_\mu(\mu) = 1/2$ this definition only yields a spectral projector if $\mu \neq \lambda_i$ for all $i = 1, \dots, n$, so in the following we are always going to assume that this condition holds. Since the eigenvalues of P_μ are all either 0 or 1, we have

$$\text{tr}(P_\mu) = \text{rank}(P_\mu) = \#\{i : \lambda_i < \mu\} =: n_e(\mu),$$

where $n_e(\mu)$ denotes the number of eigenvalues of A that are strictly smaller than μ . The problem of finding gaps in the spectrum of A can be formulated as finding intervals $[a_j, b_j]$ such that $\text{tr}(P_\mu)$ is constant for $\mu \in [a_j, b_j]$. We are going to determine gaps in the spectrum of A by approximating $\text{tr}(P_\mu)$ for several different values of μ , using Hutchinson’s trace estimator combined with the Lanczos method for the approximation of quadratic forms. We first focus on the approximation of $\text{tr}(P_\mu)$ for a fixed value of μ .

2.1 Hutchinson’s trace estimator

Stochastic trace estimators approximate the trace of a matrix B that is accessible via matrix–vector products by making use of the fact that, for any random vector x such that $\mathbb{E}[xx^T] = I$, we have $\mathbb{E}[x^T B x] = \text{tr}(B)$, where \mathbb{E} denotes the expected value. Hutchinson’s trace estimator [19] is a simple stochastic estimator that generates s vectors x_1, \dots, x_s with i.i.d. random entries, that are usually chosen as either Gaussian $\mathcal{N}(0, 1)$ or Rademacher (uniform ± 1), and approximates $\text{tr}(B)$ with

$$\text{tr}_s^H(B) := \frac{1}{s} \sum_{i=1}^s x_i^T B x_i. \tag{2.1}$$

In order to attain an absolute accuracy ε on $\text{tr}(B)$, Hutchinson’s trace estimator requires $O(\|B\|_F^2/\varepsilon^2)$ sample vectors [20, 21]. More precisely, the error from the Hutchinson’s trace estimator when B is symmetric can be bounded by using the following result from [20].

Proposition 2.1 [20, Theorem 1] *Let B be a nonzero symmetric matrix, and let $\text{tr}_s^H(B)$ denote the Hutchinson trace approximation obtained using s i.i.d. Gaussian $\mathcal{N}(0, 1)$ random vectors. If*

$$s \geq \frac{4}{\varepsilon^2} (\|B\|_F^2 + \varepsilon \|B\|_2) \log(2/\delta),$$

then

$$\mathbb{P}(|\text{tr}(B) - \text{tr}_s^H(B)| \geq \varepsilon) \leq \delta.$$

If we use vectors with Rademacher instead of Gaussian entries, the convergence rate of Hutchinson’s estimator depends on $\|B - \text{diag}(B)\|_F$ and $\|B - \text{diag}(B)\|_2$ instead of $\|B\|_F$ and $\|B\|_2$, so in certain situations it can be faster than with Gaussian vectors, for instance when B is diagonally dominant; see [20, Corollary 1] for a precise statement.

The $O(1/\varepsilon^2)$ dependence on ε of the number of sample vectors s makes it very expensive to obtain highly accurate approximations with stochastic trace estimators.

However, this is not an issue for a setting such as ours, where an absolute accuracy $\varepsilon = \frac{1}{2}$ is enough to exactly determine $\text{tr}(P_\mu)$, which is an integer. In contrast, the dependence on $\|P_\mu\|_F^2$ will make it quite difficult to accurately estimate the number of eigenvalues below μ , since $\|P_\mu\|_F^2 = n_e(\mu)$ can be quite large. Note that when μ is close to the right endpoint of the spectrum, one can consider $I - P_\mu$ instead of P_μ in order to reduce its norm, since $\text{tr}(I - P_\mu) = n - n_e(\mu)$. However, this does not handle the case where μ is near the center of the spectrum. To overcome this issue, we will have to slightly change our perspective in order to develop an efficient algorithm. This topic will be discussed in greater detail in Sect. 3.2.

Remark 2.2 Since $\text{rank}(P_\mu) = n_e(\mu)$ and all the eigenvalues of P_μ are either 0 or 1, for this setting it usually does not make much sense to use the improved stochastic trace estimators such as Hutch++ [22, 23] or XTrace [24], which also employ a low-rank matrix approximation. Indeed, there is no gradual decay in the eigenvalues of P_μ and hence there is no advantage in computing a low-rank approximation of P_μ , unless $n_e(\mu)$ is very small compared to the matrix size.

Remark 2.3 When the matrix A is sparse, we could alternatively approximate the trace of the matrix function $P_\mu = h_\mu(A)$ with a probing method (see, for instance, [25–27]). The probing approach has a faster convergence when the function $h_\mu(A)$ is well-approximated by polynomials on the spectrum of A [25, Theorem 4.4], i.e., it converges similarly to the Lanczos method for the computation of $h_\mu(A)\mathbf{x}$. We will see in Sect. 3.3 that this convergence rate is heavily dependent on the value of μ , so we prefer to use a stochastic trace estimator in order to eliminate this dependence.

2.2 Lanczos approximation of quadratic forms

Within Hutchinson’s estimator we have to compute the quadratic forms $\mathbf{x}_i^T P_\mu \mathbf{x}_i$ for $i = 1, \dots, s$, which can be done efficiently by using the Lanczos method. For simplicity, consider a single quadratic form $\mathbf{x}^T h_\mu(A)\mathbf{x}$. Let us denote by $\mathcal{K}_m(A, \mathbf{x}) = \text{span}\{\mathbf{x}, A\mathbf{x}, \dots, A^{m-1}\mathbf{x}\}$ the Krylov subspace of dimension m associated to A and \mathbf{x} , and let $V_m \in \mathbb{R}^{n \times m}$ be an orthonormal basis of $\mathcal{K}_m(A, \mathbf{x})$ constructed using the Lanczos algorithm [28, Algorithm 6.15]. The Lanczos algorithm also constructs a tridiagonal matrix $\underline{T}_m \in \mathbb{R}^{(m+1) \times m}$ that satisfies the Arnoldi relation

$$AV_m = V_{m+1}\underline{T}_m.$$

The quadratic form $\mathbf{x}^T P_\mu \mathbf{x}$ can then be approximated with

$$q_m := \|\mathbf{x}\|_2^2 \mathbf{e}_1^T h_\mu(\underline{T}_m) \mathbf{e}_1, \tag{2.2}$$

where $T_m = V_m^T AV_m$ is the leading principal $m \times m$ block of \underline{T}_m . The convergence rate of the approximation in (2.2) depends on how well the function f can be approximated with polynomials over the spectrum of A ; see Sect. 3.3. The Lanczos approximation of $\mathbf{x}^T P_\mu \mathbf{x}$ can also be interpreted and analyzed in terms of Gaussian quadrature rules [29].

2.3 Simultaneous computation for different μ

The approach described above can be easily modified in order to compute approximations to $\text{tr}(P_\mu)$ simultaneously for several μ , for a cost that is only slightly higher than the cost of the approximation for a single μ . The key observation is that for a fixed matrix A and vector \mathbf{x} , we can extract approximations to the quadratic forms $\mathbf{x}^T f_j(A)\mathbf{x}$ for several different functions f_j from a single Krylov subspace $\mathcal{K}_m(A, \mathbf{x})$. Since the cost of computing the approximation (2.2) is dominated by the cost of constructing the orthonormal basis V_m via the Lanczos algorithm, the additional operations required to compute approximations for different functions represent only a minor part of the overall cost. More precisely, let us consider N_f different functions $f_j, j = 1, \dots, N_f$. For a fixed Krylov subspace dimension m , we can approximate each quadratic form $\mathbf{x}^T f_j(A)\mathbf{x}$ with

$$\alpha_{m,j} := \|\mathbf{x}\|_2^2 \mathbf{e}_1^T f_j(T_m)\mathbf{e}_1, \quad j = 1, \dots, N_f.$$

The quantities $\alpha_{m,j}$ can be computed efficiently in the following way. Obtain first an eigenvalue decomposition $T_m = U_m D_m U_m^T$ and compute $f_j(D_m)$ for $j = 1, \dots, N_f$ by applying the functions f_j to the diagonal entries of D_m . The Lanczos approximations to the quadratic forms can now be computed with the identity

$$\alpha_{m,j} = \mathbf{w}_m^T f_j(D_m)\mathbf{w}_m, \quad j = 1, \dots, N_f,$$

where $\mathbf{w}_m := \|\mathbf{x}\|_2 U_m^T \mathbf{e}_1$ is the same for all j and needs to be computed only once. If we assume that the small matrix function $f(T_m)$ is computed via an eigenvalue decomposition also in the case of a single function f , the only additional operations that we have to do when we have N_f different functions are the computation of the diagonal matrix functions $f_j(D_m)$ and the computation of the final approximations $\alpha_{m,j} = \mathbf{w}_m^T f_j(D_m)\mathbf{w}_m$ for all j , for a total cost of $O(mN_f)$. This additional cost is negligible compared to the $O(m \text{nnz}(A))$ cost for the matrix–vector products with A in the construction of the Krylov subspace $\mathcal{K}_m(A, \mathbf{x})$, unless the number of functions N_f is extremely large.

In total, the cost of computing approximations to $\mathbf{x}^T f_j(A)\mathbf{x}$ for $j = 1, \dots, N_f$ with the Lanczos algorithm adds up to $O(m \text{nnz}(A) + nm + m^2 + mN_f)$, where the $O(nm)$ term is the cost of orthogonalization in the Lanczos algorithm using the three-term recurrence; if full reorthogonalization is used instead to increase stability, the orthogonalization cost would increase to $O(nm^2)$. The $O(m^2)$ term corresponds to the computation of eigenvalues and eigenvectors¹ of the tridiagonal matrix T_m , and the $O(mN_f)$ term corresponds to the computation of $\alpha_{m,j}$ for all j . Note that the convergence rate of the approximation of the quadratic form $\mathbf{x}^T f_j(A)\mathbf{x}$ can vary significantly depending on the function f_j , so this approach may construct approximations with considerably different errors for different functions, since we are using

¹ The $O(m^2)$ cost for computing eigenvectors of a tridiagonal matrix assumes that the MRRR algorithm [30, 31] is used, which is now the state of the art in SCALAPACK and other software packages. We note that this algorithm might fail in some extreme cases; however, remedies are known [32].

the same number of iterations m for all functions. We will focus on analyzing this aspect in Sect. 3.3.

Remark 2.4 A similar approach could be used to approximate quadratic forms with several different functions using rational Krylov methods [33], which depend on a certain sequence of poles $\xi = [\xi_1, \dots, \xi_{m-1}]$ and employ as an approximation space the rational Krylov subspace $\mathcal{Q}_m(A, \mathbf{x}, \xi) = q_{m-1}(A)^{-1} \mathcal{K}_m(A, \mathbf{b})$, with $q_{m-1}(z) := \prod_{j=1}^{m-1} (z - \xi_j)$. However, the convergence of a rational Krylov method is heavily dependent on the poles ξ , which should be chosen depending on the function. In particular, in the case of $f_j = h_{\mu_j}$ for different parameters μ_j , a sequence of poles that produces an accurate approximation for a certain μ_j will give poor approximations for other parameters, especially if they are far from μ_j . For this reason, the Lanczos method is better suited for this scenario. A rational Krylov subspace method could be useful to compute a more accurate approximation for the quadratic form $\mathbf{x}^T h_\mu(A) \mathbf{x}$ associated with a specific parameter μ , especially in situations in which the Lanczos method converges very slowly.

Returning now to the approximation of $\text{tr}(P_\mu)$ with Hutchinson's estimator, let us consider N_f different parameters $\{\mu_j\}_{j=1}^{N_f} \subset \mathbb{R}$ and define the functions $f_j = h_{\mu_j}$. Provided that we use the same random sample vectors $\{\mathbf{x}_i\}_{i=1}^s$ for the approximation of $\text{tr}(P_{\mu_j})$ via Hutchinson's estimator for all j , we can use the same Krylov subspace $\mathcal{K}_m(A, \mathbf{x}_i)$ to extract approximations to $\mathbf{x}_i^T P_{\mu_j} \mathbf{x}_i$ for all μ_j at the same time, with essentially the same cost of approximating a quadratic form for a single parameter μ . This feature of this approach represents a considerable computational advantage over other methods, such as those based on the LDL^T factorization, which are only able to test a single candidate μ at a time.

A high-level algorithmic description of the procedure outlined in this section is given in Algorithm 1. This algorithm returns an approximation to $\text{tr}(P_{\mu_j})$ for all $j = 1, \dots, N_f$. The level of accuracy of these approximations, as well as whether they can be exploited to detect gaps in the spectrum of A , is going to be thoroughly investigated in the sections that follow. We are also going to identify criteria for selecting the input parameters of the method, namely the number of Hutchinson sample vectors s and Lanczos iterations m . The product of this analysis will be an algorithm that, given as input a failure probability δ and a relative gap width θ , aims to find all the gaps in the spectrum whose relative width is larger than θ ; each one of these gaps is found with probability at least $1 - \delta$. In addition, all the gaps found by the algorithm are certified, in the sense that the probability of finding an eigenvalue of A within a gap is smaller than δ .

Remark 2.5 We mention that Algorithm 1 has been extensively used in the literature for related problems, such as spectral density estimation and estimation of eigenvalue counts in an interval [13, 15, 16], and it is also known as "stochastic Lanczos quadrature" [16]. The main novelty of our work is in the theoretical analysis contained in the following sections, which provides guarantees for the detection of spectral gaps, and allows us to select the parameters s and m in order to minimize the computational cost. We are going to provide a more detailed comparison of our contributions with those found in the literature in Sect. 4.5.

Algorithm 1 Eigenvalue gap finder – Prototype

Input: Symmetric matrix $A \in \mathbb{R}^{n \times n}$, number of Hutchinson samples s , Krylov subspace dimension m , vector of test parameters $\mu = [\mu_1, \dots, \mu_{N_f}]$
Output: $\tau \tau_j \approx \text{tr}(P_{\mu_j})$ for $j = 1, \dots, N_f$
1: **for** $i = 1, \dots, s$ **do**
2: Draw a vector $\mathbf{x}_i \in \mathbb{R}^n$ with random i.i.d. Gaussian $\mathcal{N}(0, 1)$ entries
3: Construct an orthonormal basis $V_m \in \mathbb{R}^{n \times m}$ of the Krylov space $\mathcal{K}_m(A, \mathbf{x}_i)$ with the Lanczos algorithm and compute the associated tridiagonal matrix $T_m \in \mathbb{R}^{(m+1) \times m}$
4: Compute the eigenvalue decomposition $T_m = U_m D_m U_m^T$ and $\mathbf{w}_m = \|\mathbf{x}_i\|_2 U_m^T \mathbf{e}_1$
5: **for** $j = 1, \dots, N_f$ **do**
6: Compute $\alpha_{m,j}^{(i)} = \mathbf{w}_m^T h_{\mu_j}(D_m) \mathbf{w}_m$
7: **end for**
8: **end for**
9: **for** $j = 1, \dots, N_f$ **do**
10: Compute the trace approximation $\tau \tau_j = s^{-1} \sum_{i=1}^s \alpha_{m,j}^{(i)}$
11: **end for**

3 Algorithm analysis

In this section we examine some general properties of the trace approximations computed by Algorithm 1, and we analyze in detail the errors that occur in the Hutchinson and Lanczos parts of the approximation.

3.1 General properties

A simple but essential observation is that the approximations to $\text{tr}(P_{\mu})$ computed by Algorithm 1 are increasing in μ . Let us start by looking at the Hutchinson approximation. Denoting by $(\lambda_j, \mathbf{u}_j)_{j=1}^n$ the normalized eigenpairs of A , so that $A = \sum_{j=1}^n \lambda_j \mathbf{u}_j \mathbf{u}_j^T$, we can write the trace approximation obtained with Hutchinson’s estimator as

$$\text{tr}_s^H(P_{\mu}) = \frac{1}{s} \sum_{i=1}^s \mathbf{x}_i^T P_{\mu} \mathbf{x}_i = \frac{1}{s} \sum_{i=1}^s \mathbf{x}_i^T \left(\sum_{j: \lambda_j < \mu} \mathbf{u}_j \mathbf{u}_j^T \right) \mathbf{x}_i = \frac{1}{s} \sum_{i=1}^s \sum_{j: \lambda_j < \mu} |\mathbf{u}_j^T \mathbf{x}_i|^2.$$

This shows that the Hutchinson trace approximation is monotonic in μ . In addition, the above identity also highlights the fact that Hutchinson’s approximation of $\text{tr}(P_{\mu})$ is a piecewise constant function, with jumps when μ coincides with an eigenvalue of A . The height of the jump at λ_k is given by $\frac{1}{s} \sum_{i=1}^s |\mathbf{u}_k^T \mathbf{x}_i|^2$.

Let us now consider the Lanczos approximation of the quadratic form $\mathbf{x}_i^T P_{\mu} \mathbf{x}_i$. Letting $T_m^{(i)}$ be the projection of A onto $\mathcal{K}_m(A, \mathbf{x}_i)$ computed by the Lanczos algorithm, and denoting by $(\lambda_j^{(i)}, \mathbf{u}_j^{(i)})_{j=1}^m$ the normalized eigenpairs of $T_m^{(i)}$, we have

$$\mathbf{x}_i^T P_{\mu} \mathbf{x}_i \approx \alpha_{m,j}^{(i)} := \|\mathbf{x}_i\|_2^2 \mathbf{e}_1^T h_{\mu}(T_m^{(i)}) \mathbf{e}_1 = \|\mathbf{x}_i\|_2^2 \sum_{j: \lambda_j^{(i)} < \mu} |\mathbf{e}_1^T \mathbf{u}_j^{(i)}|^2.$$

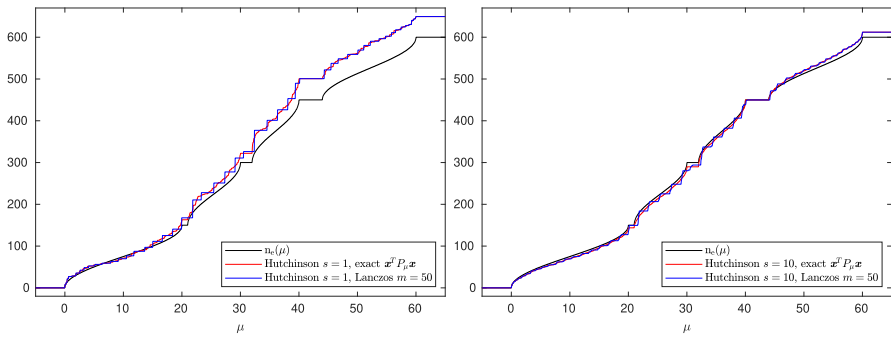


Fig. 1 Approximations to $\text{tr}(P_\mu)$ obtained with Algorithm 1 compared with exact $\text{tr}(P_\mu)$ and Hutchinson’s approximation $\text{tr}_s^H(P_\mu)$ with exact quadratic forms, for several different μ . Left: $s = 1, m = 50$. Right: $s = 10, m = 50$

Again, this is an increasing function of μ , with jumps whenever μ coincides with an eigenvalue of $T_m^{(i)}$. It follows that the approximation of $\text{tr}(P_\mu)$ computed by Algorithm 1 using Hutchinson’s estimator and the Lanczos method is an increasing function of μ , with jumps when μ is equal to an eigenvalue of $T_m^{(i)}$ for some i .

Example 3.1 To illustrate the behavior of the approximation of $\text{tr}(P_\mu)$ computed by Algorithm 1 on a simple example, we consider a symmetric matrix A of size $n = 600$, with eigenvalues contained in the interval $[0, 60]$ and three gaps given by the intervals $[20, 21]$, $[30, 32]$ and $[40, 44]$. In Fig. 1 we compare the approximation from Algorithm 1 with $m = 50$ Lanczos iterations, Hutchinson’s estimator $\text{tr}_s^H(P_\mu)$ with exact quadratic forms, and the exact trace $\text{tr}(P_\mu)$, which coincides with $n_e(\mu)$, for several different values of μ . The number of random vectors is either $s = 1$ or $s = 10$. Figure 1 confirms the behavior described above. In particular, we see that Hutchinson’s estimator with exact quadratic forms has a jump whenever μ coincides with an eigenvalue of A , so it correctly detects all the gaps in the spectrum, although the value of the approximation of $\text{tr}(P_\mu)$ can be quite off, especially with $s = 1$. On the other hand, Hutchinson’s estimator combined with the Lanczos algorithm for computing quadratic forms, which is the approximation that we are actually able to compute, has jumps when μ corresponds to an eigenvalue of one of the projected matrices $T_m^{(i)}$. We see that the Lanczos approximation is able to detect the largest gap quite well, but the other two gaps do not stand out from the other steps in the staircase-like curve. With $s = 10$, the Lanczos curve looks like a staircase with somewhat jagged steps, because the projected matrices $T_m^{(i)}$ have different eigenvalues for different random vectors x_i , and the curve is obtained by averaging the approximations $\alpha_{m,j}^{(i)}$ associated with each $T_m^{(i)}$. Note that the curves in Fig. 1 are approximations of the (unnormalized) spectral density of A , which coincide with the approximations shown, for instance, in [16, Fig. 1].

3.2 Trace estimator analysis

In this section we focus on analyzing the approximation of $\text{tr}(P_\mu)$ using Hutchinson’s estimator. The error from the stochastic trace estimator can be bounded with Theorem 2.1. Since all the eigenvalues of P_μ are either 0 or 1, we have $\|P_\mu\|_2 = 1$ and $\|P_\mu\|_F^2 = n_e(\mu)$. Therefore, in order to have $\mathbb{P}(|\text{tr}(P_\mu) - \text{tr}_p^H(P_\mu)| \geq \varepsilon) \leq \delta$ with Gaussian vectors, we would need to take

$$s \geq \frac{4}{\varepsilon^2} (n_e(\mu) + \varepsilon) \log(2/\delta). \tag{3.1}$$

Since the exact $\text{tr}(P_\mu)$ only takes integer values, in order to compute it correctly it is sufficient to have a final absolute accuracy smaller than $\frac{1}{2}$ on the trace approximation; if we also account for the error in the Lanczos approximation, it is enough to take $\varepsilon = \frac{1}{4}$. However, a key issue with this approach is the fact that the required number of sample vectors s grows proportionally to the number $n_e(\mu)$ of eigenvalues below the level μ . This means that in a situation in which a gap is not close to the extrema of the spectrum, such as if $n_e(\mu) = n/c$ for some moderate constant $c > 1$, we would have to take $s = O(n)$, which is too expensive to be practical.

Remark 3.1 In alternative to Gaussian random vectors, we can also use Rademacher vectors, which have random ± 1 entries. As we mentioned in Sect. 2.1, the convergence rate of Hutchinson’s estimator with Rademacher random vectors depends on $\|B - \text{diag}(B)\|_F^2$ instead of $\|B\|_F^2$, so it can be faster than with Gaussian vectors in certain scenarios, such as when the entries of B decay away from the diagonal. Here we prefer to employ Gaussian vectors because of the rotational invariance of their distribution, which will prove useful in the following analysis.

In order to circumvent the growth of the number of sample vectors with $n_e(\mu)$, we slightly change our point of view. Instead of considering the approximation of $\text{tr}(P_\mu)$, let us take two different values $\mu < \mu'$, and consider the difference $\text{tr}(P_{\mu'}) - \text{tr}(P_\mu)$. If we use the same sample vectors in Hutchinson’s estimator for both $\text{tr}(P_\mu)$ and $\text{tr}(P_{\mu'})$, the difference $\text{tr}_s^H(P_{\mu'}) - \text{tr}_s^H(P_\mu)$ coincides with the Hutchinson approximation of $\text{tr}(P_{\mu'} - P_\mu)$. Since we have $\|P_{\mu'} - P_\mu\|_F^2 = n_e(\mu') - n_e(\mu)$, this means that Hutchinson’s estimator can easily approximate the differences in the traces of projectors when μ and μ' are close enough that there are only a few eigenvalues of A in the interval $[\mu, \mu']$, even if the exact values of $\text{tr}(P_\mu)$ and $\text{tr}(P_{\mu'})$ are quite far from the approximations computed by the stochastic estimator. In particular, the heights of the jumps in $\text{tr}_s^H(P_\mu)$ that occur when μ coincides with an eigenvalue of A should be approximated quite accurately even with a small number of sample vectors.

Let us analyze this phenomenon in more detail. Recall that

$$\text{tr}_s^H(P_\mu) = \frac{1}{s} \sum_{i=1}^s \sum_{j: \lambda_j < \mu} |\mathbf{u}_j^T \mathbf{x}_i|^2,$$

so if we take μ and μ' such that $\lambda_{k-1} < \mu < \lambda_k < \mu' < \lambda_{k+1}$ for some k , we have

$$\text{tr}_s^H(P_{\mu'}) - \text{tr}_s^H(P_{\mu}) = \frac{1}{s} \sum_{i=1}^s |\mathbf{u}_k^T \mathbf{x}_i|^2.$$

Since the random vectors \mathbf{x}_i are independent of the eigenvector \mathbf{u}_k , we have that $\{\mathbf{u}_k^T \mathbf{x}_i\}_{i=1}^s$ are i.i.d. $\mathcal{N}(0, 1)$ random variables, due to the rotational invariance of Gaussian vectors. As a consequence, the random variable $y_s = \sum_{i=1}^s |\mathbf{u}_k^T \mathbf{x}_i|^2$ has a χ^2 distribution with s degrees of freedom, i.e., it is the sum of the squares of s independent unit normal random variables.

We have $\mathbb{E}[y_s] = s$, which means that the expected height of each jump in $\text{tr}_s^H(P_{\mu})$ is equal to 1. Intuitively, the height of a jump in $\text{tr}_s^H(P_{\mu})$ is small when the vectors \mathbf{x}_i are all approximately orthogonal to the eigenvector \mathbf{u}_k , an event that has a small chance of occurring. Since small jumps in $\text{tr}_s^H(P_{\mu})$ are more difficult to detect when approximating the quadratic forms in Hutchinson’s estimator via the Lanczos method, it will be useful to have an upper bound on the probability of small jumps. The lemma that follows provides us with such a bound. In the following, we say that the jump in $\text{tr}_s^H(P_{\mu})$ at the k -th eigenvalue of A is ε -small if its height is smaller than ε , i.e., if $y_s \leq s\varepsilon$.

Lemma 3.2 *Given $\delta \in (0, 1)$ and $\varepsilon \in (0, 1)$, if the number of samples s satisfies*

$$s \geq \left\lceil \frac{2 \log(1/\delta)}{\log(1/\varepsilon) + \varepsilon - 1} \right\rceil,$$

then we have

$$\mathbb{P}(y_s \leq s\varepsilon) \leq \delta,$$

i.e., the jump at λ_k is ε -small with probability at most δ .

Proof To prove this, we use a Chernoff bound on the lower tail of the probability distribution of y_s . For any $\lambda > 0$, we have

$$\mathbb{P}(y_s \leq s\varepsilon) = \mathbb{P}(e^{-\lambda y_s} \geq e^{-\lambda s\varepsilon}) \leq \frac{\mathbb{E}[e^{-\lambda y_s}]}{e^{-\lambda s\varepsilon}}, \tag{3.2}$$

where the last inequality follows from Markov’s inequality. Since y_s is a χ^2 random variable with s degrees of freedom, we can write $y_s = X_1^2 + \dots + X_s^2$, where $\{X_j\}_{j=1}^s$ are i.i.d. $\mathcal{N}(0, 1)$ random variables. We have

$$\mathbb{E}[e^{-\lambda X_j^2}] = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-\lambda z^2} e^{-\frac{1}{2}z^2} dz = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{1}{\sqrt{1+2\lambda}} e^{-\frac{1}{2}y^2} dy = \frac{1}{\sqrt{1+2\lambda}},$$

where we used the change of variables $y = z \sqrt{1+2\lambda}$. It follows that

$$\mathbb{E}[e^{-\lambda y_s}] = \mathbb{E}\left[\prod_{j=1}^s e^{-\lambda X_j^2}\right] = \mathbb{E}[e^{-\lambda X_1^2}]^s = (1 + 2\lambda)^{-s/2},$$

and by combining this identity with (3.2) we obtain

$$\mathbb{P}(y_s \leq s\varepsilon) \leq e^{\lambda s\varepsilon} (1 + 2\lambda)^{-s/2} \quad \forall \lambda > 0. \tag{3.3}$$

Taking $\lambda = \frac{1}{2\varepsilon} - \frac{1}{2}$, we get

$$\mathbb{P}(y_s \leq s\varepsilon) \leq (\varepsilon e^{1-\varepsilon})^{s/2}.$$

With some simple algebraic manipulations, we conclude that $\mathbb{P}(y_s \leq s\varepsilon) \leq \delta$ provided that we take

$$s \geq \frac{2 \log(1/\delta)}{\log(1/\varepsilon) + \varepsilon - 1}.$$

This concludes the proof, since the number of samples s is an integer. □

Note that $\mathbb{P}(y_s \leq s\varepsilon)$ is the probability that a jump in $\text{tr}_s^H(P_\mu)$ corresponding to a certain fixed eigenvalue of A is ε -small, and we can use Lemma 3.2 to obtain conditions on s that bound the probability of having ε -small jumps associated with any set of eigenvalues of A . This probability can be either bounded with a union bound argument, or computed directly by observing that the heights of different jumps in $\text{tr}_s^H(P_\mu)$ are independent, because the quantities $\{\mathbf{u}_k^T \mathbf{x}_i\}_{i,k}$ are all independent due to the rotational invariance of the \mathbf{x}_i and the orthogonality of the \mathbf{u}_k . The number of sample vectors s required to have an ε -small jump probability lower than δ reduces as ε decreases. At the same time, the number of Lanczos iterations required to approximate the quadratic forms with accuracy of order ε (and thus manage to detect the jumps in $\text{tr}_s^H(P_\mu)$) increases as ε becomes small. In Sect. 3.3 we obtain some bounds on the Lanczos error, which we will later compare with the ε -small jump probability bound from Lemma 3.2 in order to determine the best choice of parameters for the algorithm.

3.3 Lanczos approximation analysis

In this section we discuss the error in the Lanczos approximation of a quadratic form $\mathbf{x}^T h_\mu(A)\mathbf{x}$. Recall that we denote by

$$E_k(f, \mathcal{S}) = \min_{\deg p \leq k} \max_{z \in \mathcal{S}} |f(z) - p(z)|$$

the best uniform norm error in the approximation of a continuous function f on the compact set $\mathcal{S} \subset \mathbb{R}$ with polynomials of degree up to k . If \mathfrak{Q}_m denotes the Lanczos approximation to $\mathbf{x}^T f(A)\mathbf{x}$ from the Krylov subspace $\mathcal{K}_m(A, \mathbf{x})$, we have

$$|\mathbf{x}^T f(A)\mathbf{x} - \mathfrak{Q}_m| \leq 2\|\mathbf{x}\|_2^2 E_{2m-1}(f, \Lambda(A) \cup \Lambda(T_m)), \tag{3.4}$$

see, for instance, [34, eq. (1.2)]. Since the function $f = h_\mu$ is discontinuous at μ , we have that $E_{2m-1}(h_\mu, [a, b])$ is always at least $\frac{1}{2}$ if $\mu \in [a, b]$, so we cannot derive

meaningful error bounds if we simply consider an interval $[a, b]$ that contains the whole spectrum of A . On the other hand, we can obtain bounds on the polynomial approximation error of h_μ on two disjoint intervals, one to the left and one to the right of μ . We start by stating a result for the polynomial approximation of the sign function on the union of two symmetric intervals $[-b, -a] \cup [a, b]$.

Proposition 3.3 [35, Theorem 1] *Let $0 < a < b$. We have*

$$\lim_{m \rightarrow \infty} \sqrt{m} \left(\frac{b+a}{b-a} \right)^m E_{2m+1}(\text{sign}, [-b, -a] \cup [a, b]) = \frac{b-a}{\sqrt{\pi ab}}.$$

The statement of Proposition 3.3 implies that there exists $C > 0$ such that

$$E_{2m+1}(\text{sign}, [-b, -a] \cup [a, b]) \leq \frac{C}{\sqrt{m}} \left(\frac{b-a}{b+a} \right)^m,$$

and for m large enough we can take

$$C = \frac{b-a}{\sqrt{\pi ab}} + 1.$$

Remark 3.4 In a situation in which the spectrum of A is not symmetric with respect to the origin, using the bound in Proposition 3.3 to bound the error of the Lanczos method for the sign function can yield overly pessimistic results, since it would require us to consider intervals that are much wider than $\Lambda(A)$. In such a case, it would be more convenient to consider the polynomial approximation of the sign function on a nonsymmetric set. For instance, the asymptotics of $E_{2k+1}(\text{sign}, [-a, -1] \cup [1, b])$ for $a \neq b$ have been studied in [36], where the authors generalize Proposition 3.3 to [36, Theorem 1.1]. The result in [36] could be used to obtain more refined error bounds for the Lanczos approximation error, but the complexity of the statement, which involves Green’s functions and integral representations, makes it significantly more difficult to employ in our setting.

We can easily turn Proposition 3.3 into a bound for the step function h_μ for any $\mu \in \mathbb{R}$. Let us define

$$b_1 := \lambda_1 - \mu, \quad a_1 := \lambda_{n_c(\mu)} - \mu, \quad a_2 := \lambda_{n_c(\mu)+1} - \mu, \quad b_2 := \lambda_n - \mu,$$

and let

$$a := \min\{|a_1|, |a_2|\} \quad \text{and} \quad b := \max\{|b_1|, |b_2|\}.$$

Note that $\Lambda(A) \subset [\lambda_1, \lambda_{n_c(\mu)}] \cup [\lambda_{n_c(\mu)+1}, \lambda_n]$ and recall that $h_\mu(x) = -\frac{1}{2} \text{sign}(x - \mu) + \frac{1}{2}$, so we have

$$\begin{aligned} E_{2m+1}(h_\mu, [\lambda_1, \lambda_{n_c(\mu)}] \cup [\lambda_{n_c(\mu)+1}, \lambda_n]) &= E_{2m+1}(h_0, [b_1, a_1] \cup [a_2, b_2]) \\ &= \frac{1}{2} E_{2m+1}(\text{sign}, [b_1, a_1] \cup [a_2, b_2]) \end{aligned}$$

$$\begin{aligned} &\leq \frac{1}{2} E_{2m+1}(\text{sign}, [-b, -a] \cup [a, b]) \\ &\leq \frac{C}{2\sqrt{m}} \left(\frac{b-a}{b+a} \right)^m, \end{aligned}$$

with $C = 1 + (b - a)/\sqrt{\pi ab}$, where we used Proposition 3.3 for the last inequality. Denoting the relative spectral gap associated to μ by $\theta := a/b$, we obtain

$$E_{2m+1}(h_\mu, [\lambda_1, \lambda_{n_e(\mu)}] \cup [\lambda_{n_e(\mu)+1}, \lambda_n]) \leq \frac{C}{2\sqrt{m}} \left(\frac{1-\theta}{1+\theta} \right)^m. \tag{3.5}$$

We can combine this result with (3.4) to bound the error in the Lanczos approximation of quadratic forms with $P_\mu = h_\mu(A)$, provided that no eigenvalue of T_m lies within the gap $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$. Unfortunately, we cannot exclude a priori that an eigenvalue of T_m lies within the gap, and indeed this often happens in practice; however, since there must be at least one eigenvalue of A in the open interval between two consecutive Ritz values [37, Sect. 4], there can be at most one eigenvalue of T_m in the interval $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$. This fact can be proved using the theory of orthogonal polynomials, see, for instance, [38, Theorem 1.21], using the fact that the Ritz values are the zeros of an orthogonal polynomial associated with a Gauss quadrature rule; see, e.g., [29, Theorem 6.2]. In the analysis that follows, we assume for simplicity that there are no Ritz values in the gap $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$. We discuss how we can modify our approach to deal with this issue in Remark 3.7.

Remark 3.5 In the literature, several techniques have been used to obtain approximations to the sign matrix function that circumvent the problem of Ritz values getting arbitrarily close to zero [39], for instance by writing $\text{sign}(z) = z \cdot (z^2)^{-1/2}$ and constructing an approximation of $(A^2)^{-1/2} \mathbf{x}$ using the Krylov subspace $\mathcal{K}_m(A^2, \mathbf{x})$, which ensures that the eigenvalues of the projected matrix are bounded away from 0, since A^2 is positive definite as long as A is nonsingular. Alternatively, one can construct an approximation using harmonic Ritz values [40], which are always outside of the interval between the largest negative and the smallest positive eigenvalue of A , so they avoid the discontinuity of the sign function. Since in this work our goal is to approximate $\mathbf{x}^T h_\mu(A) \mathbf{x}$ for several μ , we opted to use the standard Lanczos approximation (2.2), for which it is very cheap and straightforward to compute approximations for many different functions.

Combining (3.4) with (3.5), we obtain the following bound on the Lanczos approximation error.

Proposition 3.6 *Let \mathcal{Q}_m be the Lanczos approximation (2.2) to $\mathbf{x}^T h_\mu(A) \mathbf{x}$ from the Krylov subspace $\mathcal{K}_m(A, \mathbf{x})$, and assume that there is no eigenvalue of T_m in the interval $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$. We have*

$$|\mathbf{x}^T h_\mu(A) \mathbf{x} - \mathcal{Q}_m| \leq \frac{C_\theta \|\mathbf{x}\|_2^2}{\sqrt{m-1}} \left(\frac{1-\theta}{1+\theta} \right)^{m-1}, \tag{3.6}$$

where θ is defined above and $C_\theta = 1 + (1 - \theta)/\sqrt{\pi\theta}$.

Note that if we take a different value of μ in the interval $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$, the error $|\mathbf{x}^T h_\mu(A)\mathbf{x} - \alpha_m|$ remains the same, because the function h_μ takes the same values on the eigenvalues of A and T_m and hence the matrix functions $h_\mu(A)$ and $h_\mu(T_m)$ do not change. On the other hand, the bound in Proposition 3.6 depends on θ , which in turn depends on μ . It is quite easy to see that the sharpest bound in (3.6) is obtained when θ is as large as possible, which occurs when μ is precisely in the middle of the gap, i.e. $\mu = \frac{1}{2}(\lambda_{n_e(\mu)} + \lambda_{n_e(\mu)+1})$.

The convergence predicted by Proposition 3.6 is at least exponential with rate $(1 - \theta)/(1 + \theta)$, with in addition an $O(1/\sqrt{m})$ factor which is independent of the gap width. Note that C_θ grows as the gap becomes smaller; see Proposition 3.3. In order to have an error $|\mathbf{x}^T h_\mu(A)\mathbf{x} - \alpha_m| \leq \varepsilon$, the number of Lanczos iterations m should satisfy

$$m + \frac{\log(m - 1)}{2 \log((1 + \theta)/(1 - \theta))} \geq 1 + \frac{\log(C_\theta \|\mathbf{x}\|_2^2/\varepsilon)}{\log((1 + \theta)/(1 - \theta))}. \tag{3.7}$$

Assuming that the relative gap θ is large enough, we do not lose much by discarding the $\log(m - 1)$ term, and we can take

$$m \geq 1 + \frac{\log(C_\theta \|\mathbf{x}\|_2^2/\varepsilon)}{\log((1 + \theta)/(1 - \theta))}. \tag{3.8}$$

On the other hand, in a situation where the spectral gap is very small, the Lanczos method effectively converges like $O(1/\sqrt{m})$. Note that if \mathbf{x} is a Gaussian random vector we have $\mathbb{E}[\|\mathbf{x}\|_2^2] = n$, so with a very small gap for which $\theta \approx 0$ we would need to take

$$m \gtrsim \frac{C_\theta^2 n^2}{\varepsilon^2},$$

which is definitely unfeasible. This means that when using the Lanczos method we can only hope to approximate $\mathbf{x}^T h_\mu(A)\mathbf{x}$ accurately when μ is inside a relatively large gap in the spectrum of A , and we should instead expect to have quite a large error whenever μ is close to an eigenvalue of A .

A crucial requirement for using the bound in Proposition 3.6 is the knowledge of the relative gap width θ . Since the gaps in the spectrum are not known to us in advance, we cannot use this result to check if the error of the Lanczos method is below a certain tolerance ε . On the other hand, Proposition 3.6 can be useful to choose the number m of Lanczos iterations at the start of the algorithm. Indeed, if our goal is to find all the gaps in the spectrum of A with relative width at least θ , for any given ε we can select m using (3.8) to ensure that the error $|\mathbf{x}^T h_\mu(A)\mathbf{x} - \alpha_m| \leq \varepsilon$ whenever μ is inside a gap with width at least θ . However, we are still unable to determine for which values of μ this condition is satisfied. The final component that we need to make the algorithm work is an a posteriori bound on the error, which will allow us to detect when $|\mathbf{x}^T h_\mu(A)\mathbf{x} - \alpha_m| \leq \varepsilon$, without knowing the gaps in the spectrum in advance. For this purpose we can use an a posteriori error bound from [26], which we adapt to the step function h_μ in Sect. 3.4.

Remark 3.7 If there is an eigenvalue of T_m inside the gap $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$, the results above do not hold. However, it is still possible to detect that there is a gap in the spectrum of A . Indeed, let us assume that the gap $(\lambda_{n_e(\mu)}, \lambda_{n_e(\mu)+1})$ has relative width θ and that there is exactly one eigenvalue τ of T_m in the gap. Then, the gap is split into the two smaller gaps $(\lambda_{n_e(\mu)}, \tau)$ and $(\tau, \lambda_{n_e(\mu)+1})$, and one of them has a relative width at least $\theta/2$. In other words, there is an interval of relative width $\theta/2$ where there are no eigenvalues of A or T_m , and hence when μ is inside that interval the Lanczos approximation error satisfies the bound (3.6) with θ replaced by $\theta/2$. In order to ensure that this gap is detected, it is then enough to use $\theta/2$ instead of θ as a target gap width for our algorithm. We remark that with this approach we are only able to detect the larger portion of the gap, and not the whole gap in $\Lambda(A)$; however, the accuracy in the approximation of the gap can be refined for a small computational cost by simply running a few extra Lanczos iterations, which would cause the eigenvalue of T_m to move around the gap or, even better, to disappear from the gap.

3.4 An a posteriori error bound for the Lanczos algorithm

In this section we adapt an a posteriori error bound given in [26] for the approximation of $\mathbf{x}^T f(A)\mathbf{x}$ with a rational Krylov method to the case of the Lanczos approximation of $f = h_\mu$. The result given in [26, Theorem 4.8] employs the Cauchy integral formula and the residue theorem to obtain upper and lower bounds on the error in the approximation of $\mathbf{x}^T f(A)\mathbf{x}$ with a rational Krylov method. This result can be easily used to bound the error in the Lanczos approximation for $f = h_\mu$, but it requires some caution since the function h_μ is discontinuous at μ . In the following, we briefly retrace the proof of [26, Theorem 4.8] and adapt it to the current setting.

If $\Gamma \subset \mathbb{C}$ is a simple closed curve that surrounds the eigenvalues of A below μ , we have

$$P_\mu = h_\mu(A) = \frac{1}{2\pi i} \int_\Gamma (zI - A)^{-1} dz.$$

If the eigenvalues of T_m that are smaller than μ also lie in the interior of Γ , we have

$$h_\mu(A)\mathbf{x} - \|\mathbf{x}\|_2 V_m h_\mu(T_m)\mathbf{e}_1 = \frac{1}{2\pi i} \int_\Gamma (zI - A)^{-1} \mathbf{r}_m(z) dz,$$

where

$$\mathbf{r}_m(z) = \mathbf{x} - (zI - A)\mathbf{x}_m(z), \quad \text{with} \quad \mathbf{x}_m(z) = \|\mathbf{x}\|_2 V_m (zI - T_m)^{-1} \mathbf{e}_1,$$

that is, $\mathbf{r}_m(z)$ is the residual of a shifted linear system after m iterations of the Lanczos method. With the same algebraic manipulations used in [26, Sect. 4.3], we obtain

$$\mathbf{x}^T h_\mu(A)\mathbf{x} - \|\mathbf{x}\|_2^2 \mathbf{e}_1^T h_\mu(T_m)\mathbf{e}_1 = \frac{\|\mathbf{x}\|_2^2}{2\pi i} \int_\Gamma \varphi_m(z)^2 \mathbf{v}_{m+1}^T (zI - A)^{-1} \mathbf{v}_{m+1} dz,$$

with

$$\varphi_m(z) = t_{m+1,m} \mathbf{e}_m^T (zI - T_m)^{-1} \mathbf{e}_1,$$

where $t_{m+1,m}$ denotes the element in position $(m + 1, m)$ of the matrix T_m . Using an eigenvalue decomposition $T_m = U_m D_m U_m^T$, with U_m orthogonal and $D_m = \text{diag}(\theta_1, \dots, \theta_m)$, we can define

$$[\alpha_1, \dots, \alpha_m] := t_{m+1,m} \mathbf{e}_m^T U_m, \quad [\beta_1, \dots, \beta_m]^T := U_m^T \mathbf{e}_1$$

and

$$\gamma_j := \sum_{k:k \neq j} \alpha_k \beta_k \frac{1}{\theta_j - \theta_k}, \quad j = 1, \dots, m.$$

With these definitions, the function $\varphi_m(z)^2$ can be rewritten in the form

$$\varphi_m(z)^2 = \sum_{j=1}^m \alpha_j^2 \beta_j^2 \frac{1}{(z - \theta_j)^2} + 2 \sum_{j=1}^m \alpha_j \beta_j \gamma_j \frac{1}{z - \theta_j}.$$

Using the residue theorem, with the same procedure used in [26, Sect. 4.3] we obtain

$$\mathbf{x}^T h_\mu(A) \mathbf{x} - \|\mathbf{x}\|_2^2 \mathbf{e}_1^T h_\mu(T_m) \mathbf{e}_1 = \|\mathbf{x}\|_2^2 \mathbf{v}_{m+1}^T g_m(A) \mathbf{v}_{m+1}, \tag{3.9}$$

with

$$g_m(z) := \begin{cases} \sum_{j:\theta_j > \mu} \alpha_j^2 \beta_j^2 \frac{1}{(z - \theta_j)^2} + 2\alpha_j \beta_j \gamma_j \frac{1}{z - \theta_j} & \text{if } z < \mu, \\ - \sum_{j:\theta_j < \mu} \alpha_j^2 \beta_j^2 \frac{1}{(z - \theta_j)^2} - 2\alpha_j \beta_j \gamma_j \frac{1}{z - \theta_j} & \text{if } z > \mu. \end{cases} \tag{3.10}$$

We can use (3.9) to obtain the following upper bound for the Lanczos approximation error, which corresponds to the upper bound in [26, Theorem 4.8].

Proposition 3.8 *Let \mathfrak{q}_m be the Lanczos approximation (2.2) to $\mathbf{x}^T h_\mu(A) \mathbf{x}$ from the Krylov subspace $\mathcal{K}_m(A, \mathbf{x})$. We have*

$$|\mathbf{x}^T h_\mu(A) \mathbf{x} - \mathfrak{q}_m| \leq \|\mathbf{x}\|_2^2 \max_{z \in \Lambda(A)} |g_m(z)| \leq \|\mathbf{x}\|_2^2 \max_{z \in [\lambda_{\min}, \lambda_{\max}]} |g_m(z)|,$$

where g_m is defined in (3.10).

In contrast with what happens when the function f is continuous over the whole spectral interval of A , in this case the function $g_m(z)$ is discontinuous at μ and the lower bound from [26, Theorem 4.8] no longer holds. Indeed, if we denote by $(\lambda_k, \mathbf{u}_k)$ the normalized eigenpairs of the matrix A , we can write

$$\mathbf{v}_{m+1}^T g_m(A) \mathbf{v}_{m+1} = \sum_{k=1}^m g_m(\lambda_k) (\mathbf{u}_k^T \mathbf{v}_{m+1})^2, \quad \text{with} \quad \sum_{k=1}^m (\mathbf{u}_k^T \mathbf{v}_{m+1})^2 = 1.$$

This expression shows that the quadratic form $\mathbf{v}_{m+1}^T g_m(A) \mathbf{v}_{m+1}$ can potentially take any value in the convex hull of the set $\{g_m(\lambda_k)\}_{k=1}^m$, depending on the coefficients $(\mathbf{u}_k^T \mathbf{v}_{m+1})^2$. In the case that $g_m(\lambda_k)$ takes both positive and negative values, it is possible to have $\mathbf{v}_{m+1}^T g_m(A) \mathbf{v}_{m+1} = 0$, so the lower bound

$$|\mathbf{x}^T h_\mu(A) \mathbf{x} - \|\mathbf{x}\|_2^2 \mathbf{e}_1^T h_\mu(T_m) \mathbf{e}_1| \geq \|\mathbf{x}\|_2^2 \min_{z \in [\lambda_{\min}, \lambda_{\max}]} |g_m(z)|$$

may not hold. Note that this argument is not in contradiction with the statement of [26, Theorem 4.8] when the function g_m is continuous over the whole spectral interval of A , because if $g_m(\lambda_k)$ takes both negative and positive values there must be a point $z \in [\lambda_{\min}, \lambda_{\max}]$ such that $g_m(z) = 0$ by continuity of g_m .

Example 3.2 In this example we compare the error of the Lanczos approximation of $\mathbf{x}^T h_\mu(A) \mathbf{x}$ with the a priori error bound from Proposition 3.6, the a posteriori error bound from Proposition 3.8, and the simple error estimate $|\mathbf{x}^T h_\mu(A) \mathbf{x} - \alpha_m| \approx |\alpha_m - \alpha_{m+1}|$. We consider a symmetric matrix A of size $n = 1000$ with $\Lambda(A) \subset [0, 40]$, and gaps in the intervals $[10, 11]$ and $[20, 25]$. We plot the convergence of the Lanczos method in Fig. 2, where on the left plot $\mu = 22$ is contained in the larger gap, and on the right plot $\mu = 10.6$ is inside the smaller gap. In the legend, the ‘‘central μ ’’ a priori bound refers to the bound obtained by taking μ in the middle of the gap, as mentioned below Proposition 3.6.

Note that the convergence of the Lanczos method is much slower when μ is inside the smaller gap, and in both cases the convergence behavior is irregular and oscillating. The ‘‘central μ ’’ is the most accurate between the two a priori bounds, but it does not capture the correct convergence rate for $\mu = 10.6$, which is further from the middle of the spectrum; in this case, a better a priori bound would be obtained by using a convergence result with nonsymmetric intervals, such as [36, Theorem 1.1]. The a posteriori bound is also quite irregular, but it is able to capture the convergence rate of the Lanczos approximation correctly in both cases, although it is off by a constant factor. Surprisingly, the error estimate $|\mathbf{x}^T h_\mu(A) \mathbf{x} - \alpha_m| \approx |\alpha_m - \alpha_{m+1}|$ is very accurate, and it seems to almost never underestimate the error. This behavior can be explained in the following way: the error estimate based on consecutive differences satisfies the triangle inequality

$$|\alpha_m - \alpha_{m+1}| \geq \left| |\mathbf{x}^T h_\mu(A) \mathbf{x} - \alpha_m| - |\mathbf{x}^T h_\mu(A) \mathbf{x} - \alpha_{m+1}| \right|,$$

and due to the oscillations in the convergence of the Lanczos approximation, the two right-hand side terms $|\mathbf{x}^T h_\mu(A) \mathbf{x} - \alpha_m|$ and $|\mathbf{x}^T h_\mu(A) \mathbf{x} - \alpha_{m+1}|$ often have significantly different moduli, so we have

$$\left| |\alpha - \alpha_m| - |\alpha - \alpha_{m+1}| \right| \geq c \max \{ |\alpha - \alpha_m|, |\alpha - \alpha_{m+1}| \}, \quad \alpha := \mathbf{x}^T h_\mu(A) \mathbf{x}, \tag{3.11}$$

for a constant c that is close to 1. For instance, if $|\alpha - \alpha_m| > |\alpha - \alpha_{m+1}|$, we can take

$$c = 1 - \frac{|\alpha - \alpha_{m+1}|}{|\alpha - \alpha_m|}.$$

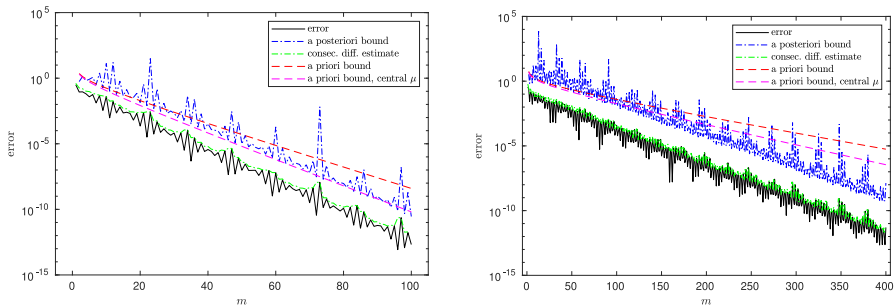


Fig. 2 Approximation of $\mathbf{x}^T h_\mu(A)\mathbf{x}$ with the Lanczos algorithm, and comparison with the a priori error bound from Proposition 3.6, the a posteriori error bound from Proposition 3.8 and the consecutive difference estimate. The spectrum of A has gaps in the intervals $[10, 11]$ and $[20, 25]$. Left: $\mu = 22$. Right: $\mu = 10.6$

The inequality (3.11) explains why the consecutive difference estimate is so often an upper bound when the Lanczos method has oscillating convergence, as in the current setting.

4 Final algorithm

In this section we put together all the theoretical results obtained in Sect. 3 to improve Algorithm 1 and obtain an efficient algorithm that also provides some guarantees on its output.

4.1 Choosing the input parameters

First of all, we need a criterion to decide the number s of Hutchinson sample vectors and the number m of Lanczos iterations. For this purpose, given a failure probability δ and a target relative gap width θ , we set the goal to find all gaps in the spectrum of A with relative width greater than or equal to θ , with probability $1 - \delta$.

How should we choose s and m so that all these gaps are found? Let us consider a candidate gap $[\mu_1, \mu_2] \subset \mathbb{R}$ and fix a parameter $\varepsilon > 0$, and say that we take m such that the Lanczos approximation error for each quadratic form $\mathbf{x}_i^T P_\mu \mathbf{x}_i$ is smaller than $\varepsilon/2$ for all μ inside the gap, so that $\text{tr}_s^H(P_\mu)$ is contained in an interval of width smaller than ε for all $\mu \in [\mu_1, \mu_2]$. Then there are two possible scenarios: either there are no eigenvalues of A in the interval $[\mu_1, \mu_2]$, or all the jumps in $\text{tr}_s^H(P_\mu)$ associated to eigenvalues inside that interval are ε -small (more precisely, the sum of the heights of these jumps is smaller than ε). This second event has a small probability of occurring, which can be bounded using Lemma 3.2, so we can conclude that $[\mu_1, \mu_2]$ is a gap in the spectrum of A with high probability.

The probability that all jumps in $\text{tr}_s^H(P_\mu)$ associated to eigenvalues in $[\mu_1, \mu_2]$ are ε -small is smaller than the probability of a single jump being ε -small, so we can impose

$$\mathbb{P}(y_s \leq s\varepsilon) \leq \delta,$$

where y_s is a χ^2 distribution with s degrees of freedom, see Sect. 3.2. By Lemma 3.2, this condition is satisfied provided that

$$s \geq \frac{2 \log(1/\delta)}{\log(1/\varepsilon) + \varepsilon - 1}.$$

If the relative width of $[\mu_1, \mu_2]$ is at least θ , by Proposition 3.6 we have

$$\left| \text{tr}_s^H(P_\mu) - \frac{1}{s} \sum_{i=1}^s \mathfrak{Q}_m^{(i)} \right| \leq \frac{C_\theta \sum_{i=1}^s \|\mathbf{x}_i\|_2^2}{s\sqrt{m-1}} \left(\frac{1-\theta}{1+\theta} \right)^{m-1},$$

where we denote by $\mathfrak{Q}_m^{(i)}$ the approximation of $\mathbf{x}_i^T P_\mu \mathbf{x}_i$ after m Lanczos iterations. By (3.8), the above bound is smaller than $\varepsilon/2$ if we take

$$m \geq 1 + \frac{\log(2C_\theta \sum_{i=1}^s \|\mathbf{x}_i\|_2^2 / s\varepsilon)}{\log((1+\theta)/(1-\theta))}, \quad C_\theta = \frac{1-\theta}{\sqrt{\pi\theta}} + 1. \tag{4.1}$$

Note that although the a priori bound in Proposition 3.6 allows us to say that the Lanczos error is smaller than $\varepsilon/2$ when μ is inside a gap wider than θ , we cannot use it to determine for which values of μ this actually happens. To detect the gaps in practice, we are going to use either the a posteriori bound from Proposition 3.8 or the error estimate based on consecutive differences, and assume that it is at least as accurate as the a priori bound, i.e., that it is also smaller than $\varepsilon/2$ inside a gap with relative width θ . This assumption is usually satisfied, as shown in Example 3.2, at least in the central portion of each gap. Recalling the discussion in Example 3.7, we can replace θ with $\theta/2$ to ensure that all gaps of width θ are detected even when there is a Ritz value within a gap.

We have obtained bounds on s and m that depend on δ, θ and ε , where the latter is a parameter that can be freely chosen. In order to obtain an algorithm that is as efficient as possible, we should aim to minimize the term that dominates the computational cost of the algorithm, which is the $O(sm \text{ nnz}(A))$ term corresponding to the sm matrix–vector products with A . We have

$$sm \geq \frac{2 \log(2/\delta)}{\log(1/\varepsilon) + \varepsilon - 1} \cdot \left(1 + \frac{\log(2C_\theta \sum_{i=1}^s \|\mathbf{x}_i\|_2^2 / s\varepsilon)}{\log((1+\theta)/(1-\theta))} \right),$$

and ε should be chosen in order to minimize the right hand side. With some simple computations one can show that the function

$$f(\varepsilon) = \frac{1}{\log(1/\varepsilon) + \varepsilon - 1} \cdot \left(1 + \frac{\log(2C_\theta n/\varepsilon)}{\log((1+\theta)/(1-\theta))} \right)$$

is increasing for $\varepsilon \in (0, 1)$ since $2C_\theta n \geq 1$. Recalling that $\mathbb{E}[\|\mathbf{x}_i\|^2] = n$ for all i , this implies that we should take ε as small as possible, and since s must be an integer, in

practice the most efficient choice is to take ε so that the probability of a jump being ε -small is lower than δ already when $s = 1$, i.e.,

$$\varepsilon \leq e^{-1} \delta^2. \quad (4.2)$$

In other words, we should simply use a single quadratic form $\mathbf{x}^T P_\mu \mathbf{x}$ to estimate $\text{tr}(P_\mu)$, instead of using the more general stochastic trace estimator with s different random vectors.

We mention that bounds on the number of Hutchinson sample vectors required to achieve a certain accuracy have been obtained in [15, 16] in the context of spectral density estimation. For instance, it is shown in [16, Theorem 1] that, in order to obtain an accuracy of order t on the normalized spectral density in the Wasserstein distance, it is enough to take roughly $\lceil 1/nt^2 \rceil$ sample vectors. When t is not too small, this implies that it is sufficient to use a single vector in Hutchinson's estimator, a result that matches the conclusion reached in this section. However, note that the result from [16] is not applicable in our setting: indeed, an absolute error ε on $n_e(\mu)$ corresponds to an error $t = \varepsilon/n$ on the normalized spectral density, so that the number of samples predicted by [16, Theorem 1] would be roughly $\lceil n/\varepsilon^2 \rceil$, which grows linearly with n , matching our initial observations in Sect. 3.2. Another key difference is that the theoretical results for spectral density estimation are usually given in terms of the Wasserstein distance or some other global measure for the quality of approximation. On the other hand, we are only interested in approximating the spectral density close to large gaps, so it is reasonable to expect that better results can be obtained with a different kind of analysis. We refer to Sect. 4.5 for further discussion and comparison of our analysis with the existing literature.

4.2 Detecting gaps in the spectrum

To detect the gaps in the spectrum, we use an a posteriori error bound to determine when the error of the Lanczos approximation is smaller than $\varepsilon/2$. We focus on the case $s = 1$, with a single random Gaussian vector $\mathbf{x} \in \mathbb{R}^n$, since we concluded that it is the most efficient choice in Sect. 4.1. More precisely, for each μ we perform m iterations of the Lanczos method to obtain $\mathfrak{a}_m(\mu) \approx \mathbf{x}^T P_\mu \mathbf{x}$ with the approximation (2.2), and we use either Proposition 3.8 or the estimate based on consecutive differences to obtain

$$|\mathbf{x}^T P_\mu \mathbf{x} - \mathfrak{a}_m(\mu)| \leq \text{bound}_m(\mu),$$

where either

$$\text{bound}_m(\mu) = \|\mathbf{x}\|_2^2 \max_{z \in [\lambda_{\min}, \lambda_{\max}]} |g_m(z)| \quad \text{with } g_m(z) \text{ as defined in (3.10),}$$

or $\text{bound}_m(\mu) = c|\mathfrak{a}_m(\mu) - \mathfrak{a}_{m+1}(\mu)|$, with a safety factor $c \geq 1$. As shown in Example 3.2, the consecutive difference error estimate is usually more accurate than the a posteriori error bound from Proposition 3.8, and very often we can also expect it to be an actual upper bound due to the oscillation in the convergence of the Lanczos

algorithm. In the discussion that follows, we are going to assume that $\text{bound}_m(\mu)$ is always an upper bound for the error $|\mathbf{x}^T P_\mu \mathbf{x} - \alpha_m(\mu)|$; since there is no guarantee that this is true for the consecutive difference estimate, we will discuss some ways to make it more robust in Example 4.1. For all μ , letting

$$\begin{aligned} \alpha_m^U(\mu) &:= \alpha_m(\mu) + \text{bound}_m(\mu), \\ \alpha_m^L(\mu) &:= \alpha_m(\mu) - \text{bound}_m(\mu), \end{aligned}$$

we have $\mathbf{x}^T P_\mu \mathbf{x} \in [\alpha_m^L(\mu), \alpha_m^U(\mu)]$.

Recalling that the quadratic form $\mathbf{x}^T P_\mu \mathbf{x}$ is an increasing function of μ , we can refine the upper and lower bounds $\alpha_m^U(\mu)$ and $\alpha_m^L(\mu)$. Indeed, we can also take the upper and lower bounds to be increasing in μ : specifically, for any $\mu < \mu'$ we must have

$$\mathbf{x}^T P_\mu \mathbf{x} \leq \mathbf{x}^T P_{\mu'} \mathbf{x} \leq \alpha_m^U(\mu') \quad \text{and} \quad \mathbf{x}^T P_{\mu'} \mathbf{x} \geq \mathbf{x}^T P_\mu \mathbf{x} \geq \alpha_m^L(\mu),$$

so we can replace the bounds $\alpha_m^U(\mu)$ and $\alpha_m^L(\mu)$ with, respectively,

$$\widehat{\alpha}_m^U(\mu) := \min_{\mu' \geq \mu} \alpha_m^U(\mu') \quad \text{and} \quad \widehat{\alpha}_m^L(\mu) := \max_{\mu' \leq \mu} \alpha_m^L(\mu'). \tag{4.3}$$

Note that the bounds $\widehat{\alpha}_m^U(\mu)$ and $\widehat{\alpha}_m^L(\mu)$ are by construction increasing functions of μ . We can further improve the bounds by taking the best ones over several values of m . Indeed, given a set of positive integers $\mathcal{M} \subset \mathbb{N}$, we can define

$$\widehat{\alpha}_{\mathcal{M}}^U(\mu) := \min_{m \in \mathcal{M}} \widehat{\alpha}_m^U(\mu) \quad \text{and} \quad \widehat{\alpha}_{\mathcal{M}}^L(\mu) := \max_{m \in \mathcal{M}} \widehat{\alpha}_m^L(\mu), \tag{4.4}$$

and we still have $\mathbf{x}^T P_\mu \mathbf{x} \in [\widehat{\alpha}_{\mathcal{M}}^L(\mu), \widehat{\alpha}_{\mathcal{M}}^U(\mu)]$. Using a set \mathcal{M} with more than a single value of m can be useful to improve the bounds in the case when there is an eigenvalue of T_m inside one of the gaps. In such a situation, by considering bounds for $\mathcal{M} = \{m - d, \dots, m - 1, m\}$ there is a higher chance that one of the matrices T_{m-j} , $j = 0, \dots, d$ has no eigenvalues in the gap, so the bounds $\widehat{\alpha}_{\mathcal{M}}^U(\mu)$ and $\widehat{\alpha}_{\mathcal{M}}^L(\mu)$ are likely going to be better than $\widehat{\alpha}_m^U(\mu)$ and $\widehat{\alpha}_m^L(\mu)$.

After incorporating these improvements, we can conclude that if ε is chosen according to (4.2) and for some $\mu_1 < \mu_2$ we have $|\widehat{\alpha}_{\mathcal{M}}^U(\mu_2) - \widehat{\alpha}_{\mathcal{M}}^L(\mu_1)| \leq \varepsilon$, then there are no eigenvalues of A in the interval $[\mu_1, \mu_2]$ with probability at least $1 - \delta$.

4.3 Algorithm summary

The analysis in the previous sections leads to the following algorithm, whose pseudocode is given in Algorithm 2. Given an input failure probability δ and target relative gap width θ , we first compute ε according to (4.2) such that the probability of having an ε -small jump at an eigenvalue in Hutchinson’s trace estimator is less than δ when using a single random vector (i.e., with $s = 1$). We then use Proposition 3.6 to select

Algorithm 2 Eigenvalue gap finder

Input: Symmetric matrix $A \in \mathbb{R}^{n \times n}$, failure probability δ , target relative gap size θ , vector of test parameters $\boldsymbol{\mu} = [\mu_1, \dots, \mu_{N_f}]$

Output: $\text{tr}_j \approx \text{tr}(P_{\mu_j})$ for $j = 1, \dots, N_f$, intervals $[a_i, b_i]$ that contain no eigenvalues of A with probability $1 - \delta$ each

- 1: Compute ε according to (4.2), ensuring that the probability of a jump being smaller than ε with $s = 1$ Hutchinson vectors is smaller than δ .
- 2: Draw a random Gaussian vector $\mathbf{x} \in \mathbb{R}^n$.
- 3: Compute m according to (4.1), ensuring that the Lanczos approximation (2.2) to $\mathbf{x}^T P_{\mu} \mathbf{x}$ has an error smaller than $\varepsilon/2$ if μ is within a gap of relative width θ or larger.
- 4: Select a set \mathcal{M} of integers smaller than or equal to m , such as $\mathcal{M} = \{m - 4, \dots, m - 1, m\}$.
- 5: Compute V_m and T_m by running m iterations of the Lanczos algorithm with the matrix A and the vector \mathbf{x} .
- 6: Compute the eigendecomposition $T_m = U_m D_m U_m^T$ and $\mathbf{w}_m = \|\mathbf{x}\|_2 U_m^T \mathbf{e}_1$.
- 7: **for** $j = 1, \dots, N_f$ **do**
- 8: Compute $\mathbf{a}_{m,j} = \mathbf{w}_m^T h_{\mu_j}(D_m) \mathbf{w}_m$.
- 9: **for** $m' \in \mathcal{M}$ **do**
- 10: Compute the monotone upper and lower bounds $\widehat{\mathbf{q}}_{m'}^U(\mu_j)$ and $\widehat{\mathbf{q}}_{m'}^L(\mu_j)$ using (4.3), where $\text{bound}_{m'}(\mu_j)$ is obtained either with Proposition 3.8 or by running Lanczos for one additional iteration to compute $|\mathbf{a}_{m',j} - \mathbf{a}_{m'+1,j}|$.
- 11: **end for**
- 12: Compute the bounds $\widehat{\mathbf{q}}_{\mathcal{M}}^U(\mu_j) = \min_{m' \in \mathcal{M}} \widehat{\mathbf{q}}_{m'}^U(\mu_j)$ and $\widehat{\mathbf{q}}_{\mathcal{M}}^L(\mu_j) = \max_{m' \in \mathcal{M}} \widehat{\mathbf{q}}_{m'}^L(\mu_j)$.
- 13: **end for**
- 14: Compute $\widehat{\mathbf{q}}_{\mathcal{M}}^U(\boldsymbol{\mu}) - \widehat{\mathbf{q}}_{\mathcal{M}}^L(\boldsymbol{\mu})$ and determine the largest intervals $[a_i, b_i]$ such that $|\widehat{\mathbf{q}}_{\mathcal{M}}^U(b_i) - \widehat{\mathbf{q}}_{\mathcal{M}}^L(a_i)| \leq \varepsilon$.

the number m of Lanczos iterations that ensures that the error of the Lanczos approximation to the quadratic form $\mathbf{x}^T h_{\mu}(A) \mathbf{x}$ is smaller than $\varepsilon/2$ whenever μ is inside a gap with relative width at least θ . As an alternative, we could fix the number of Lanczos iterations m a priori, but by doing so we would lose the guarantee that all the gaps with width larger than θ are located. Then, we run m iterations of the Lanczos method to compute the approximation (2.2) and the upper and lower a posteriori bounds $\widehat{\mathbf{q}}_{\mathcal{M}}^U(\mu)$ and $\widehat{\mathbf{q}}_{\mathcal{M}}^L(\mu)$ defined in (4.4), for all input values of μ and a set \mathcal{M} of integers smaller than or equal to m , such as $\{m - d, \dots, m - 1, m\}$, with, e.g., $d = 2$. All the intervals $[\mu_1, \mu_2]$ such that $|\widehat{\mathbf{q}}_{\mathcal{M}}^U(\mu_2) - \widehat{\mathbf{q}}_{\mathcal{M}}^L(\mu_1)| \leq \varepsilon$ are hence gaps in the spectrum of A with probability at least $1 - \delta$.

Note that m is selected so that the a priori error bound on the approximation of $\mathbf{x}^T h_{\mu}(A) \mathbf{x}$ from Proposition 3.6 is smaller than $\varepsilon/2$ whenever μ is inside a gap with relative width at least θ , so, assuming that the a posteriori error bound or estimate is at least as accurate as the a priori one, we expect that all gaps of relative width θ or larger are detected by Algorithm 2. Numerical evidence shows that this assumption is generally satisfied when μ is not too close to the extrema of the gap and ε is not too large. We also mention that, although Algorithm 2 cannot explicitly find the gap between the k -th and the $(k + 1)$ -th eigenvalue for a given k , it still computes estimates for $\text{tr}(P_{\mu_j})$ for all $\mu_j \in \boldsymbol{\mu}$, so we know the approximate number of eigenvalues below each detected gap and thus we can roughly determine if one of the gaps found by the algorithm is likely to be the one between λ_k and λ_{k+1} (of course, we can only expect that Algorithm 2 will detect this gap if we assume that it is wide enough). If

more accuracy on the number of eigenvalues below each gap is required, then one should use a larger number s of sample vectors for Hutchinson’s trace estimator. Alternatively, if it is feasible to compute an LDL^T factorization, we can use a single factorization to compute the exact number of eigenvalues below a certain gap in order to verify if it corresponds to the searched gap between the eigenvalues λ_k and λ_{k+1} . Observe that while computing such a factorization can be quite expensive, it would still be significantly cheaper than trying to find the gap between λ_k and λ_{k+1} by factorizing shifted matrices $A - \mu I$ in a trial-and-error approach, which would require the computation of several LDL^T factorizations.

Remark 4.1 Recall that $\mathbf{x}^T P_\mu \mathbf{x}$ is a staircase-like piecewise constant function of μ , and in particular it must be constant in any interval that contains no eigenvalues of A . This means that given a candidate gap $[\mu_1, \mu_2]$, in addition to the condition $|\widehat{q}_{\mathcal{M}}^U(\mu_2) - \widehat{q}_{\mathcal{M}}^L(\mu_1)| \leq \varepsilon$, we would also need to check that $\widehat{q}_{\mathcal{M}}^L(\mu_2) < \widehat{q}_{\mathcal{M}}^U(\mu_1)$. If this additional condition is not satisfied, it is impossible for $\mathbf{x}^T P_\mu \mathbf{x}$ to be a constant function for $\mu \in [\mu_1, \mu_2]$, and hence we can conclude with certainty that there must be at least one eigenvalue of A in the interval $[\mu_1, \mu_2]$. This seems to cause a contradiction when the condition $|\widehat{q}_{\mathcal{M}}^U(\mu_2) - \widehat{q}_{\mathcal{M}}^L(\mu_1)| \leq \varepsilon$ is verified but $\widehat{q}_{\mathcal{M}}^L(\mu_2) < \widehat{q}_{\mathcal{M}}^U(\mu_1)$ is not, since it would imply that all the eigenvalues of A contained in $[\mu_1, \mu_2]$ have an associated ε -small jump, which is a low-probability event. However, it simply means that this situation is very unlikely to happen in a practical scenario, and it would probably not be encountered if the algorithm was run on the same problem for a second time. In our implementation, we also check if the condition $\widehat{q}_{\mathcal{M}}^L(\mu_2) < \widehat{q}_{\mathcal{M}}^U(\mu_1)$ holds, even though in practice we never encountered a situation in which it was not satisfied.

Remark 4.2 We mention that one could implement a variant of Algorithm 2 which does not require as input a vector of parameters $\boldsymbol{\mu} = [\mu_1, \dots, \mu_{N_f}]$, but instead detects gaps by sequentially evaluating the upper and lower bounds $\widehat{q}_{\mathcal{M}}^U(\mu)$ and $\widehat{q}_{\mathcal{M}}^L(\mu)$ on suitably chosen points. Such an algorithm would likely employ fewer than N_f values of μ to reach the same accuracy. However, since the bulk of the computational cost is in the construction of the Krylov subspace, the number of points N_f has a negligible impact on performance (see Sects. 2.3 and 4.4) and thus we did not explore this approach in detail, opting instead for a more straightforward strategy with a sufficiently dense grid of points. We note that in order to detect all target gaps with our approach, N_f must be chosen so that there are at least two μ_j that lie in each gap with width greater than θ .

4.4 Computational cost

The most expensive operations in Algorithm 2 are the m matrix–vector products with A required to construct the Krylov basis in the Lanczos algorithm, which cost $O(m \text{nnz}(A))$ for a sparse matrix A . The additional operations done for the computation of the quadratic forms $\mathbf{x}^T P_{\mu_j} \mathbf{x}$ with N_f different parameters μ_j add up to $O(nm + m^2 + mN_f)$, as discussed in Sect. 2.3. If we estimate the error with the consecutive difference estimate, we have to run the Lanczos algorithm for one additional iteration, and compute the differences $|\mathfrak{q}_{m',j} - \mathfrak{q}_{m'+1,j}|$ for all $m' \in \mathcal{M}$. If $|\mathcal{M}| = d$

and the elements of \mathcal{M} are consecutive integers, this amounts to computing $d + 1$ quadratic forms $\mathfrak{q}_{m',j}$ for all j , so the additional cost is $O(m^2d + mN_f d)$.

On the other hand, if we use the a posteriori error bound from Proposition 3.8, we have to evaluate the coefficients α_j, β_j and γ_j for all μ_j , which can be done with $O(m^2)$ operations assuming that the eigenvalue decomposition of T_m is available, and compute the maximum of $|g_m(z)|$ over $[\lambda_{\min}, \lambda_{\max}]$, where g_m is defined in (3.10). The evaluation of $g_m(z)$ for all μ_j at a single point z requires $O(mN_f)$ operations, and if a discretization of $[\lambda_{\min}, \lambda_{\max}]$ with L points is used, we can approximately evaluate the bound in Proposition 3.8 for all μ_j with $O(m^2 + mN_f L)$ operations. If a set \mathcal{M} with d elements is used to compute the more robust bounds $\widehat{\mathfrak{q}}_{\mathcal{M}}^U(\mu_j)$ and $\widehat{\mathfrak{q}}_{\mathcal{M}}^L(\mu_j)$, the cost for the computation of the bounds is roughly multiplied by a factor d and becomes $O(m^2d + mN_f Ld)$, where we have also included the cost of computing the eigenvalues and eigenvectors of $T_{m'}$ for $m' \in \mathcal{M}$. Note that given $\mathfrak{q}_m^U(\mu_j)$ and $\mathfrak{q}_m^L(\mu_j)$, the refined bounds $\widehat{\mathfrak{q}}_m^U(\mu_j)$ and $\widehat{\mathfrak{q}}_m^L(\mu_j)$ described in Sect. 4.2 can be computed for all μ_j with $O(N_f)$ operations. Finally, once $\widehat{\mathfrak{q}}_{\mathcal{M}}^U(\mu_j)$ and $\widehat{\mathfrak{q}}_{\mathcal{M}}^L(\mu_j)$ have been computed, the gaps can be located with a cost of $O(N_f)$. The total computational cost of Algorithm 2 is therefore $O(m \text{ nnz}(A) + nm + m^2d + mN_f d)$ with the consecutive difference estimate, and $O(m \text{ nnz}(A) + nm + m^2d + mN_f Ld)$ with the a posteriori bound from Proposition 3.8.

Note that all operations associated with different values of μ are completely independent, so the algorithm can be easily parallelized, for instance by assigning different slices of the spectral interval to different processors. This parallelization is not particularly helpful when the matrix size n is large enough, since the dominant part of the computation would still be the $O(m \text{ nnz}(A))$ term associated with matrix–vector products with A ; however, parallelizing the computations associated with different parameters μ can be helpful for mitigating the $O(mN_f Ld)$ cost of computing the a posteriori error bounds from Proposition 3.8, which can become quite high if both N_f and L are large, see, for instance, the experiment in Sect. 5.3.

If our goal is to detect all gaps with relative width larger than θ with probability at least $1 - \delta$, we should take m according to (4.1) with $\varepsilon \leq e^{-1}\delta^2$, i.e.,

$$m \geq 1 + \frac{1 + \log(2C_\theta \|\mathbf{x}\|_2^2/\delta^2)}{\log((1 + \theta)/(1 - \theta))}, \quad C_\theta = \frac{1 - \theta}{\sqrt{\pi\theta}} + 1, \tag{4.5}$$

where we recall that $\mathbb{E}[\|\mathbf{x}\|_2^2] = n$. By looking at the behavior of the right-hand side of (4.5) as $\theta \rightarrow 0$, using the fact that

$$\log\left(\frac{1 + \theta}{1 - \theta}\right) \approx \frac{1}{2\theta},$$

we conclude that for small θ we should take

$$m \approx \log\left(\frac{2\|\mathbf{x}\|_2^2}{\delta^2\sqrt{\pi\theta}}\right) \frac{1}{2\theta}.$$

Although the dependence of m on the failure probability δ and the matrix size n is not particularly severe since they appear inside a logarithm, the main drawback of this approach is the $O(1/\theta)$ dependence on the relative gap width θ , which causes Algorithm 2 to quickly become more expensive as the target gap width is reduced. This is an inherent limitation of the Lanczos method for the computation of $\mathbf{x}^T P_\mu \mathbf{x}$, and more precisely of the polynomial approximation of the sign function (see Proposition 3.3). As we mentioned in Sect. 3.3, Proposition 3.3 can be improved by considering two nonsymmetric intervals as in [36], and this would in turn lead to an improved estimate on the required number of Lanczos iterations m ; however, in general we do not expect to find an asymptotic estimate that is better than $m = O(1/\theta)$. Indeed, for a gap located near the center of $\Lambda(A)$, the spectrum of A is contained in two intervals that are approximately symmetric with respect to the gap, and there would not be any significant advantage in using the improved result [36, Theorem 1.1].

Example 4.1 In this example we show how the bounds described in Sect. 4.2 perform on the simple test problem from Example 3.1. The matrix A is 600×600 , with $\Lambda(A) \subset [0, 60]$ and three gaps in the spectrum given by the intervals $[20, 21]$, $[30, 32]$ and $[40, 44]$. We use a single Gaussian vector $\mathbf{x} \in \mathbb{R}^n$ for Hutchinson’s estimator and $m = 50$ Lanczos iterations. We compare the exact $\mathbf{x}^T P_\mu \mathbf{x}$ with the upper and lower bounds $\widehat{q}_{\mathcal{M}}^U(\mu)$ and $\widehat{q}_{\mathcal{M}}^L(\mu)$ obtained both with Proposition 3.8 and with the consecutive difference error estimate. In particular, for the latter we use the error estimate $\text{bound}_m(\mu_j) = c|\alpha_{m,j} - \alpha_{m+1,j}|$, with a safety factor $c = 2$. In both cases, we take $\mathcal{M} = \{m - d + 1, \dots, m\}$ with $d = 3$. Since $c|\alpha_{m,j} - \alpha_{m+1,j}|$ is not guaranteed to be an upper bound for $|\mathbf{x}^T P_{\mu_j} \mathbf{x}|$, the inequalities $\widehat{q}_{\mathcal{M}}^L(\mu) \leq \mathbf{x}^T P_\mu \mathbf{x} \leq \widehat{q}_{\mathcal{M}}^U(\mu)$ do not necessarily hold for all μ , especially after the two minimizations and maximizations in (4.3) and (4.4). To overcome this issue, we also propose to use a “safe” variant of the upper and lower estimates for $\mathbf{x}^T P_\mu \mathbf{x}$, defined as

$$\widehat{q}_{\mathcal{M}}^{U*}(\mu) := \max_{m \in \mathcal{M}} \widehat{q}_m^U(\mu) \quad \text{and} \quad \widehat{q}_{\mathcal{M}}^{L*}(\mu) := \min_{m \in \mathcal{M}} \widehat{q}_m^L(\mu), \tag{4.6}$$

which correspond to taking the worst of the bounds associated with \mathcal{M} instead of the best ones. This variant significantly increases the chance that $\widehat{q}_{\mathcal{M}}^{U*}(\mu)$ and $\widehat{q}_{\mathcal{M}}^{L*}(\mu)$ are an upper and lower bound for $\mathbf{x}^T P_\mu \mathbf{x}$, respectively, although it reduces the overall accuracy of the estimates.

The bounds are shown in Fig. 3, including the safe variants $\widehat{q}_{\mathcal{M}}^{U*}(\mu)$ and $\widehat{q}_{\mathcal{M}}^{L*}(\mu)$ of the bounds obtained by using consecutive differences. Notice that the upper and lower estimates $\widehat{q}_{\mathcal{M}}^U(\mu)$ and $\widehat{q}_{\mathcal{M}}^L(\mu)$ computed with consecutive differences are much closer to $\mathbf{x}^T P_\mu \mathbf{x}$ compared to the bounds obtained with Proposition 3.8, but they sometimes fail to satisfy $\widehat{q}_{\mathcal{M}}^L(\mu) \leq \mathbf{x}^T P_\mu \mathbf{x} \leq \widehat{q}_{\mathcal{M}}^U(\mu)$: see for instance the right plot in Fig. 3, which is a close-up of the larger gap $[40, 44]$. On the other hand, the safe variants $\widehat{q}_{\mathcal{M}}^{U*}(\mu)$ and $\widehat{q}_{\mathcal{M}}^{L*}(\mu)$ are more robust, while still being more accurate than the a posteriori bounds from Proposition 3.8. To provide a quantitative comparison, for the plot in Fig. 3 we used 4000 different shift parameters μ_j , and the inequality $\mathbf{x}^T P_{\mu_j} \mathbf{x} \leq \widehat{q}_{\mathcal{M}}^U(\mu_j)$ was not satisfied for 251 of them, while $\mathbf{x}^T P_{\mu_j} \mathbf{x} \leq \widehat{q}_{\mathcal{M}}^{U*}(\mu_j)$ was not satisfied for only 35 values of j . Note that the fraction of values of μ for which

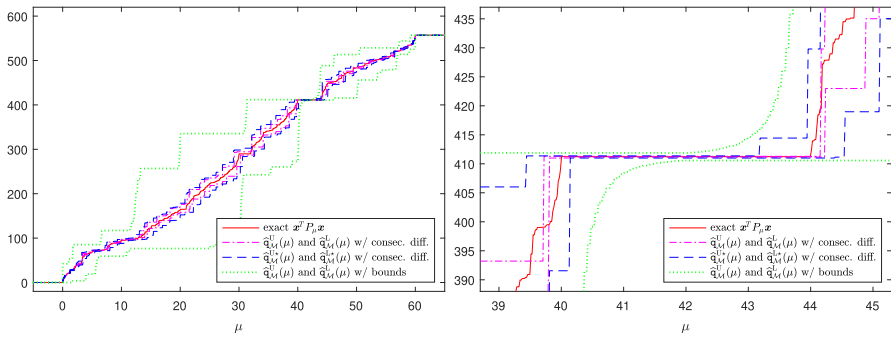


Fig. 3 Upper and lower bounds for $\mathbf{x}^T P_\mu \mathbf{x}$ obtained with the Lanczos algorithm with $m = 50$ iterations. The right plot is a close-up of the left plot around the larger gap [40, 44]

the safe upper estimate $\widehat{q}_{\mathcal{M}}^{U*}(\mu)$ fails to be an upper bound is smaller than 10^{-2} , which is the failure probability that we use for the experiments in Sect. 5.

4.5 Literature comparison

As we already mentioned in some of the previous sections, the combination of Hutchinson’s trace estimator with the Lanczos algorithm has been used in several works on related problems, for instance in [13, 15, 16]. The main aspect that sets our work apart from the rest of the literature is the focus on the detection of gaps, and the consequent difference in the theoretical analysis that we perform. Indeed, most results in the literature on spectral density approximation are concerned with global approximation of the spectral density, such as in the Wasserstein distance [16, Theorem 1], so they are unable to exploit the faster convergence of the Lanczos method for $\mathbf{x}^T h_\mu(A)\mathbf{x}$ when μ is located inside a large gap (see Proposition 3.6). Moreover, convergence results found in the literature are usually based on the approximation error for $\text{tr}(P_\mu)$ with Hutchinson’s estimator, using results such as Theorem 2.1. In our context, we require an absolute error on $n_c(\mu) = \text{tr}(P_\mu)$ smaller than $\frac{1}{2}$, which corresponds to a relative error of $\frac{1}{2n}$ on the normalized spectral density; as we observed at the end of Sect. 4.1, achieving this level of accuracy with a bound such as [16, Theorem 1] would require us to take $s = O(n)$ vectors for Hutchinson’s estimator, which are too many to make this approach feasible.

On the other hand, our analysis based on the height of the jumps in $\mathbf{x}^T h_\mu(A)\mathbf{x}$ and $\text{tr}_s^H(P_\mu)$ bypasses the need to obtain a highly accurate approximation of the trace and allows us to use significantly fewer matrix–vector products. The main tradeoff is that we do not obtain the precise number of eigenvalues below each detected gap, but only an estimate. To the best of our knowledge, this perspective and the resulting approach for the theoretical analysis have not appeared before in the literature. Although we have not explored this direction, it is possible that some of the techniques used in this work may also be applicable to related research areas in which the combination of Hutchinson and Lanczos is employed, such as spectral density estimation.

5 Numerical experiments

In this section we run some experiments to investigate the performance of Algorithm 2. For all the experiments, we use Algorithm 2 with a failure probability $\delta = 10^{-2}$ and a number of Lanczos iterations m that is either fixed in advance or computed according to (4.5) in order to find all gaps with relative width larger than a certain target θ . When not stated differently, for the Lanczos algorithm we employ the error estimate $\text{bound}_m(\mu_j) = c|\alpha_{m,j} - \alpha_{m,j+1}|$ with a safety factor $c = 2$, and we use the more robust upper and lower estimates $\hat{\alpha}^{\text{U}\star}(\mu)$ and $\hat{\alpha}^{\text{L}\star}(\mu)$ defined in (4.6), with $\mathcal{M} = \{m - d + 1, \dots, m\}$ with $d = 3$. These estimates are cheaper to compute compared to the a posteriori error bound from Proposition 3.8, and they are also quite reliable, as shown in Example 4.1. Even though the estimate $c|\alpha_{j,m} - \alpha_{j,m+1}|$ does not have the same theoretical guarantees as the a posteriori error bound, we have seen in Example 3.2 that it is usually much closer to the actual error. All the experiments have been done using MATLAB R2024b on a laptop running Ubuntu 24.04, with 32 GB of RAM and an Intel Core Ultra 9 185H CPU with base clock rate 2.3 GHz. The code to reproduce the experiments in this section is available on Github at <https://github.com/simunek/eigenvalue-gap-finder>.

5.1 Scaling with the gap width

We start by examining the behavior of Algorithm 2 with respect to the relative gap width θ .

We consider a symmetric tridiagonal matrix A of size $n = 30,000$ obtained as a sum $A = D + T$, where T is a random symmetric tridiagonal matrix with $\mathcal{N}(0, 1)$ entries and D is a diagonal matrix. The eigenvalues of D are chosen so that the spectrum of A has a gap with relative width approximately θ . Specifically, the eigenvalues of D are contained in the intervals $[1, 10^3] \cup [10^3 + x, 10^4]$, where x satisfies

$$\frac{x/2}{10^4 - 10^3 - x/2} = \theta,$$

so that the relative width of the gap $[10^3, 10^3 + x]$ is θ , and hence the spectrum of the perturbed matrix $A = D + T$ also has a gap with relative width approximately θ . We consider a matrix D with 20,000 logspaced eigenvalues in $[1, 10^3]$ and 10,000 logspaced eigenvalues in $[10^3 + x, 10^4]$.

We estimate this gap using Algorithm 2, with m computed according to (4.5), so that we can expect it to successfully find all gaps with relative width larger than θ ; we use $N_f = 10,000$ parameters μ , logarithmically spaced on the interval $[1, 10^4]$. For each θ , in Table 1 we show the true gap computed by diagonalizing A , the gap estimated by Algorithm 2 and the approximate number of eigenvalues below the gap, obtained by rounding the trace approximation $\mathbf{x}^T P_\mu \mathbf{x}$ for μ equal to the left endpoint of the estimated gap, as well as the number of Lanczos iterations m , the execution time of the algorithm and the time required for the computation of the eigenvalues of A . We see that the gap is found successfully for all θ , with the estimated gap being

slightly larger than the true gap only for $\theta = 0.1$ and $\theta = 0.05$. The estimated number of eigenvalues below the gap roughly approximates the exact value 20,000, but it is still relatively far from the correct number since we are using only one sample vector for Hutchinson's estimator. Note that the number of Lanczos iterations scales approximately as $O(1/\theta)$, but the execution time increases more than linearly in the number of Lanczos iterations m ; this is likely due to the $O(m^2)$ cost for the computation of eigenvalues and eigenvectors of the projected matrix T_m becoming an increasingly dominant part of the computation when the subspace dimension m is large. On the other hand, the execution time for eig remains constant, so when the gap width θ becomes sufficiently small it would be more efficient to directly diagonalize the matrix instead of using Algorithm 2. Of course, this would not be the case if the matrix dimension n increases, as we show in the following experiment.

5.2 Scaling with the matrix size

Let us now consider a sequence of matrices with increasing size and approximately constant gap width. We use a setup similar to Sect. 5.2, with fixed relative gap width $\theta = 0.01$ and increasing matrix size n . We take $A = D + T$, where T is symmetric tridiagonal with random $\mathcal{N}(0, 1)$ entries and D is diagonal with $\Lambda(D) \subset [1, 10^3] \cup [10^3 + x, 10^4]$, with $n/2$ logspaced eigenvalues below the gap and $n/2$ logspaced eigenvalues above the gap. The value of x is chosen so that A has relative gap width approximately θ , as in Sect. 5.2.

We use Algorithm 2 to estimate the gap, with the number of Lanczos iterations m computed according to (4.5) and $N_f = 10,000$ parameters μ , logarithmically spaced over the interval $[1, 10^4]$. The results for a matrix dimension that increases from $n = 5000$ to $n = 80,000$ are shown in Table 2. In this case, the gap is found correctly for all matrix sizes, with none of the estimated gaps being larger than the true gap computed via a diagonalization. Note that the number of Lanczos iterations increases very slowly with the matrix dimension, in accordance with the discussion in Sect. 4.4. Hence, the execution time of the algorithm increases roughly linearly, since with a large matrix size n and moderate subspace dimension m the leading term in the computational cost is given by the m matrix–vector products with A in the construction of the Lanczos basis, which cost $O(mn)$ for a tridiagonal matrix. On the other hand, the time required for computing the eigenvalues of the tridiagonal matrix A scales like $O(n^2)$, so it quickly becomes more expensive than Algorithm 2. Note that if A was a sparse matrix that is not tridiagonal, the diagonalization time would scale like $O(n^3)$, making the difference between the two approaches even more evident.

It is interesting to observe that Algorithm 2 can be also used to cheaply obtain a rough estimate of the gap by taking a larger failure probability δ and fixing the number of Lanczos iterations m in advance instead of computing it with (4.5). For instance, we show in Table 3 the performance of Algorithm 2 with $\delta = 0.5$ and $m = 250$. We see that the gap in the spectrum is still located successfully, although the accuracy is generally lower and sometimes the estimated gap is larger than the true gap, for instance in the case $n = 5000$. Note that the estimated number of eigenvalues below the gap coincides with the corresponding number from Table 2, even if the number

Table 1 Performance of Algorithm 2 with $n = 30,000$ and varying gap width θ

θ	True gap	Est. gap	Est. $n_e(\mu)$	m	Time (s)	Time eig (s)
0.1	[1001.80, 2633.87]	[1016.48, 2637.19]	19,637	112	0.103	2.318
0.05	[1001.65, 1856.42]	[999.77, 1856.64]	19,776	226	0.174	2.212
0.025	[1001.15, 1435.81]	[1001.61, 1434.55]	20,198	456	0.312	2.293
0.01	[1001.17, 1175.30]	[1001.61, 1174.65]	20,027	1156	0.789	2.077
0.005	[1000.69, 1085.83]	[1000.69, 1085.18]	19,862	2342	1.894	2.308
0.0025	[1003.69, 1042.75]	[1004.38, 1042.08]	19,709	4745	5.537	2.311

of Lanczos iterations is different: this happens because we used the same random vector \mathbf{x} for Hutchinson's trace estimator, and in both cases we are approximating the same quadratic form $\mathbf{x}^T P_\mu \mathbf{x}$ (the values of μ in the two cases may be different, but as long as they are both inside the gap the corresponding values of $\mathbf{x}^T P_\mu \mathbf{x}$ coincide). In particular, this also confirms that the error in the estimated number of eigenvalues below the gap is essentially only due to the error in the stochastic trace estimator, and not to the error in the approximation of the quadratic form via the Lanczos algorithm.

5.3 Examples from applications

In this section, we demonstrate the effectiveness of Algorithm 2 on two illustrative examples arising from practical applications.

5.3.1 Dirac comb Hamiltonian

As a first model problem, we consider the one-dimensional discrete Hamiltonian operator

$$A = -L + V,$$

where L is a finite difference discretization of the second derivative on $[0, N]$ with periodic boundary conditions using a uniform grid with k points on each unit interval, for a total of $n = kN$ points, and V is a diagonal matrix representing a discrete *Dirac comb* periodic potential, which takes the value k^2 at the points $x = 1, 2, \dots, N - 1$, and is zero otherwise. This problem is a toy model that mimics the properties of a Hamiltonian of a one-dimensional physical system with a periodic potential, whose spectrum is typically characterized by several gaps, representing forbidden energy levels for the electrons in the system; see for instance [3, Chap. 7] and [41] for more details.

We use a discretization with $N = 2000$ and $k = 5$ for this example, for a total of $n = 10,000$ discretization points. The spectrum of the resulting symmetric tridiagonal matrix $A \in \mathbb{R}^{n \times n}$, shown in Fig. 4, has five bands, separated by four relatively large gaps, each containing a single isolated eigenvalue, which splits it into two parts; we will refer to them respectively as the lower and upper part of the gap. We shift and scale A so that its spectrum is contained in the interval $[0, 10]$, to facilitate visualization of the gaps. We run Algorithm 2 to detect the gaps, both with the robust consecutive difference estimates (4.6) and with the error bounds from Proposition 3.8, using $m = 150$ Lanczos iterations and $d = 3$. In Fig. 5 we show the approximations to $\mathbf{x}^T P_\mu \mathbf{x}$ and the upper and lower bounds obtained with the two approaches, where \mathbf{x} is the single random vector used in Hutchinson's estimator; we use the same vector \mathbf{x} for the two variants of Algorithm 2. Note that the bounds from Proposition 3.8 are significantly less accurate than the estimates (4.6) in regions far from the gaps, but both approaches are extremely accurate inside the gaps. With the chosen parameters, Algorithm 2 successfully detects both parts of the first gap, the lower half of the second and third gaps, and both parts of the fourth gap; by increasing the number of Lanczos iterations to $m = 250$, all gaps are successfully detected.

Table 2 Performance of Algorithm 2 with fixed gap width $\theta = 0.01$ and increasing matrix size n

n	True gap	Est. gap	est. $n_c(\mu)$	m	Time (s)	Time eig (s)
5000	[1000.58, 1177.15]	[1000.69, 1176.81]	2377	1067	0.259	0.077
10,000	[1001.11, 1175.88]	[1001.61, 1175.73]	4956	1101	0.405	0.291
20,000	[1000.23, 1177.01]	[1000.69, 1176.81]	9835	1136	0.572	1.071
40,000	[1002.67, 1175.70]	[1003.46, 1174.65]	19,930	1171	0.984	3.696
80,000	[1001.39, 1176.18]	[1001.61, 1175.73]	39,874	1205	1.842	13.161

Table 3 Performance of Algorithm 2 with fixed gap width $\theta = 0.01$ and increasing matrix size n . The number of Lanczos iterations is fixed to $m = 250$ and the failure probability is set to $\delta = 0.5$

n	True gap	Est. gap	Est. $\text{ne}(\mu)$	m	Time (s)	Time eig (s)
5000	[1000.58, 1177.15]	[998.85, 1178.98]	2377	250	0.072	0.077
10,000	[1001.11, 1175.88]	[1003.46, 1170.33]	4956	250	0.098	0.287
20,000	[1000.23, 1177.01]	[1001.61, 1167.10]	9835	250	0.145	1.037
40,000	[1002.67, 1175.70]	[1002.54, 1161.73]	19,930	250	0.230	3.748
80,000	[1001.39, 1176.18]	[1005.31, 1175.73]	39,874	250	0.399	13.085

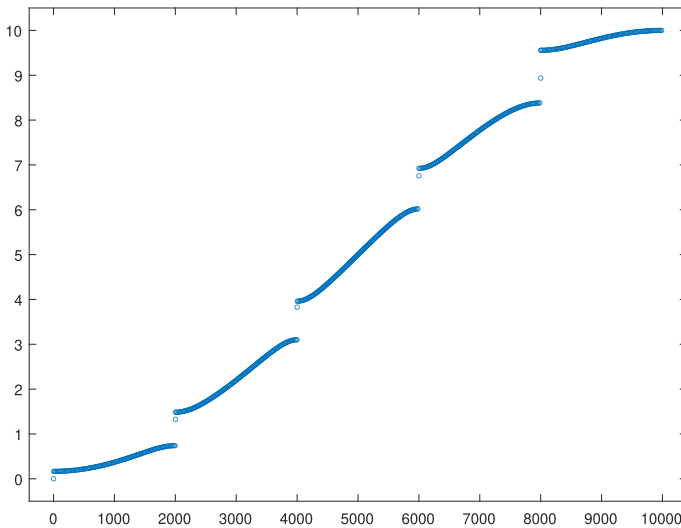


Fig. 4 Spectrum of the Dirac comb discrete Hamiltonian

In Table 4 we compare the exact gaps obtained by directly computing the eigenvalues of A with the approximate ones obtained with Algorithm 2, for the lower half of the four gaps, along with the time required to run each method. Note that the gaps found by using the a posteriori bounds from Proposition 3.8 are slightly smaller than the ones obtained by estimating the Lanczos error with consecutive differences; this is a consequence of the lower accuracy of the a posteriori bounds. Moreover, the execution time is quite higher with the a posteriori bounds, due to the $O(mN_f Ld)$ term in the computation of the error bounds from Proposition 3.8; indeed, it turns out that this is the most expensive part of the computation in this setting, where we use $N_f = 1000$ different values of μ and a discretization with $L = 1000$ points to evaluate the maximum of $|g_m(z)|$ over the spectral interval of A . On the other hand, when the error in the Lanczos algorithm is estimated using consecutive differences, the portion of the computational cost that depends on N_f scales as $O(mN_f d)$, which is significantly less expensive than the error bounds from Proposition 3.8 when $L = 1000$ (see Sect. 4.4).

5.3.2 Hydrogen molecule chain

As an example from electronic structure computations, we consider the discretized Hamiltonian associated with a linear chain of 1250H_2 molecules. In contrast to the atomic hydrogen chain, which is widely used as a theoretical benchmark but does not occur in nature, the molecular chain corresponds to a physically realizable system while still being straightforward to define; see, e.g., [42].

The matrices are generated with the open-source package PySCF [43], a Python package for performing electronic structure computations using Density Functional Theory. The code returns the Hamiltonian \tilde{A} , obtained through a Galerkin discretization over a basis of Gaussian functions, and the overlap matrix S . Although the basis

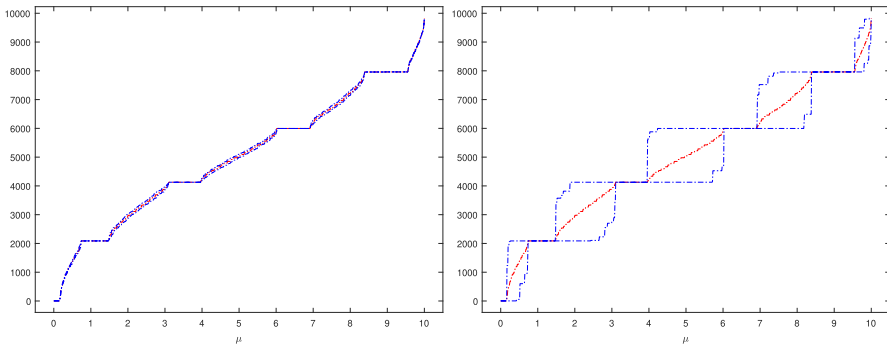


Fig. 5 Approximations to $\mathbf{x}^T P_\mu \mathbf{x}$, and upper and lower bounds obtained with Algorithm 2 for different μ , with $m = 150$ Lanczos iterations, for the problem in Sect. 5.3.1. Left: Algorithm 2 with robust consecutive difference error estimate (4.6). Right: Algorithm 2 with a posteriori error bound from Proposition 3.8

functions have global support, the Hamiltonian and overlap matrices are approximately sparse: most entries decay rapidly with the distance from the main diagonal and are set to zero once they fall below the threshold used for the integrals². The basis can be orthogonalized by taking the Cholesky factorization $S = R^T R$, where S is the overlap matrix. Thus, the new Hamiltonian is $A = R^{-T} \tilde{A} R^{-1}$, and the energy levels correspond to the eigenvalues of A ; see [2, Section 3]. When the intermolecular spacing in the chain is sufficiently large, the system behaves as a semiconductor [42], i.e., the spectrum exhibits a gap between occupied and unoccupied states. To ensure this, we impose an intermolecular distance of 2.5\AA , while the bond length within the H_2 molecule is set to 0.74\AA .

The test matrix A has size $n = 5000$. The spectrum, shown in Fig. 6, has two relatively large gaps and a smaller gap in the middle of the spectrum. There is a single isolated eigenvalue within the third gap, splitting it into two parts. In electronic structure computations, the gap of interest is the HOMO-LUMO gap, which in this setting corresponds to the first gap in Fig. 6. The original Hamiltonian \tilde{A} and the Cholesky factor R have 189,622 and 99,804 nonzero entries, respectively. Although A is also approximately sparse, in order to compute the matrix–vector products needed for the construction of the Krylov basis it is more computationally efficient to use the representation $A = R^{-T} \tilde{A} R^{-1}$ instead of forming A explicitly, so we rely on the solution of the linear systems with R and R^T , which are triangular and sparse, and on multiplications with \tilde{A} .

In this test we use a setup similar to Sect. 5.3.1. We find the gaps in the spectrum of A using Algorithm 2, and we compare the robust consecutive difference estimates defined in (4.6) with the a posteriori bound from Proposition 3.8, both obtained using $m = 100$ Lanczos iterations and $d = 3$; we use $N_f = 1000$ values of μ , and a discretization with $L = 1000$ points for evaluating the a posteriori error bounds. We plot in Fig. 7 the approximations to $\mathbf{x}^T P_\mu \mathbf{x}$ and the upper and lower bounds obtained with the two approaches. We again observe that the upper and lower bounds obtained

² In libcint, the constant `EXPCUTOFF` is set to 10^{-15} in `cint.h`; see the libcint GitHub repository <https://github.com/sunqm/libcint>.

Table 4 Performance of Algorithm 2 on the discrete Hamiltonian with a Dirac comb potential, using the robust consecutive difference estimate (4.6) and the a posteriori bound from Proposition 3.8. The number of Lanczos iterations is set to $m = 150$ and the failure probability to $\delta = 10^{-2}$. The gaps in the table correspond to the lower parts of the gaps in Fig. 4

Method	First gap	Second gap	Third gap	Fourth gap	Time (s)
e.i.g	[0.739, 1.326]	[3.101, 3.828]	[6.021, 6.757]	[8.383, 8.937]	9.498
Consec. diff	[0.741, 1.321]	[3.103, 3.824]	[6.106, 6.747]	[8.418, 8.929]	0.028
Proposition 3.8	[0.781, 1.321]	[3.223, 3.754]	[6.136, 6.717]	[8.418, 8.929]	0.365

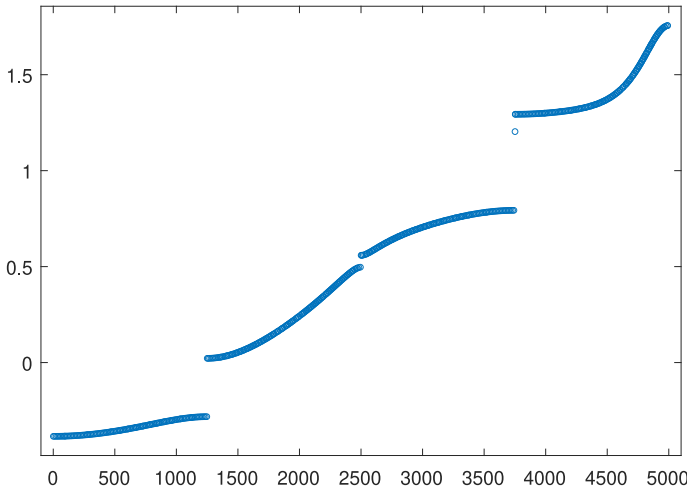


Fig. 6 Spectrum of the discrete Hamiltonian for the hydrogen molecule chain

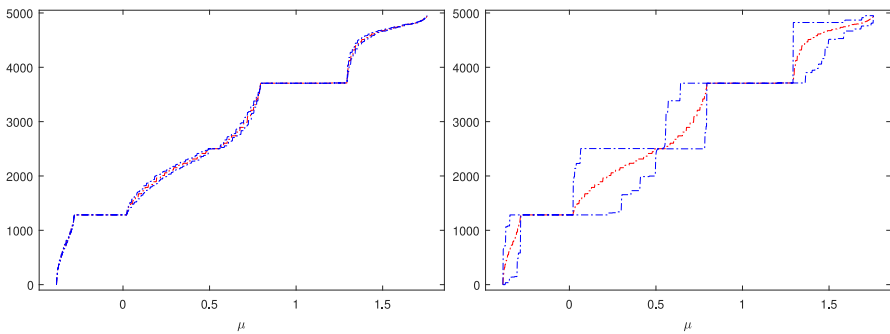


Fig. 7 Approximations to $x^T P_\mu x$, and upper and lower bounds obtained with Algorithm 2 for different μ , with $m = 100$ Lanczos iterations, for the problem in Sect. 5.3.2. Left: Algorithm 2 with robust consecutive difference error estimate (4.6). Right: Algorithm 2 with a posteriori error bound from Proposition 3.8

by using Proposition 3.8 are much more conservative compared to those based on consecutive difference estimates, especially far from the gaps in the spectrum.

The gaps found by the two variants of Algorithm 2 and the exact gaps obtained by diagonalizing A are shown in Table 5, where we also include the execution times for all the methods. With the chosen parameters, Algorithm 2 is not able to detect the small gap in the middle of the spectrum, but increasing the number of Lanczos iterations to $m = 300$ allows us to detect that gap as well. The results are consistent with the conclusions drawn from the experiment in Sect. 5.3.1.

Table 5 Performance of Algorithm 2 on the discrete Hamiltonian A associated with a H_2 chain, using the robust consecutive difference estimate (4.6) and the a posteriori bound from Proposition 3.8. The number of Lanczos iterations is set to $m = 100$ and the failure probability to $\delta = 10^{-2}$

Method	First gap	Second gap	Third gap	Time (s)
e.i.g	$[-0.282, 0.022]$	$[0.793, 1.204]$	$[1.204, 1.294]$	15.284
Algorithm 2 w/ consec. diff	$[-0.281, 0.021]$	$[0.794, 1.202]$	$[1.204, 1.294]$	0.064
Algorithm 2 w/ Proposition 3.8	$[-0.273, 0.006]$	$[0.814, 1.202]$	$[1.204, 1.272]$	0.386

6 Conclusions

We have described and analyzed an algorithm for identifying gaps in the spectrum of a real symmetric matrix by simultaneously approximating the traces of spectral projectors associated with different slices of its spectrum, combining Hutchinson's stochastic trace estimator and the Lanczos algorithm. We have investigated the behavior of Hutchinson's estimator, and provided a priori and a posteriori error bounds for the Lanczos algorithm. This analysis led to an algorithm capable of detecting all gaps larger than a specified threshold with high probability. We have also determined that for this problem the best strategy is to use a single random sample vector for Hutchinson's estimator, and utilize all the available matrix–vector products to approximate a single quadratic form via the Lanczos algorithm. Our numerical tests demonstrate that the developed algorithm can efficiently and reliably detect gaps, especially when their width is relatively large. Within our algorithm we have proposed to use either an a posteriori error bound for the Lanczos approximation error (Proposition 3.8), or a simple error estimate based on consecutive differences. Although the a posteriori error bound has better theoretical guarantees, the robust version of the error estimate from Example 4.1 has more accurate results in practice, and it is also computationally cheaper, notably when a large number of values of μ is used. Even when the algorithm fails to detect a gap, it still provides upper and lower bounds for $\mathbf{x}^T P_\mu \mathbf{x}$ for all considered values of μ , which can be valuable for selecting a starting point for other algorithms, such as those based on LDL^T factorizations.

Acknowledgements We are grateful to Tyler Chen and two anonymous reviewers for their valuable comments and suggestions. The first and third authors are members of the INdAM Research group GNCS (Gruppo Nazionale di Calcolo Scientifico), and their work was supported in part by MUR (Italian Ministry of University and Research) through the PRIN Project 20227PCCKZ (“Low-Rank Structures and Numerical Methods in Matrix and Tensor Computations and their Applications”). The first author also acknowledges partial support from MUR through the PNRR MUR Project PE0000023-NQSTI. The work of the second author is funded by the Research Foundation - Flanders (FWO) via the junior postdoctoral fellowship 12A1325N.

Funding Open access funding provided by Scuola Normale Superiore within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Lin, L., Lu, J., Ying, L.: Numerical methods for Kohn–Sham density functional theory. *Acta Numer.* **28**, 405–539 (2019). <https://doi.org/10.1017/s0962492919000047>
2. Benzi, M., Boito, P., Razouk, N.: Decay properties of spectral projectors with applications to electronic structure. *SIAM Rev.* **55**(1), 3–64 (2013). <https://doi.org/10.1137/100814019>
3. Kittel, C., McEuen, P.: *Introduction to Solid State Physics*. Wiley, Hoboken, NJ (2018)

4. Reed, M., Simon, B.: *Methods of Modern Mathematical Physics. IV. Analysis of Operators*. Academic Press, New York (1978)
5. Andreotti, E., Edelmann, D., Guglielmi, N., Lubich, C.: Measuring the stability of spectral clustering. *Linear Algebra Appl.* **610**, 673–697 (2021). <https://doi.org/10.1016/j.laa.2020.10.015>
6. Vömel, C., Parlett, B.N.: Detecting localization in an invariant subspace. *SIAM J. Sci. Comput.* **33**(6), 3447–3467 (2011). <https://doi.org/10.1137/09077624X>
7. Benzi, M., Rinelli, M.: Refined decay bounds on the entries of spectral projectors associated with sparse Hermitian matrices. *Linear Algebra Appl.* **647**, 1–30 (2022). <https://doi.org/10.1016/j.laa.2022.04.005>
8. Björck, Å.: *Numerical Methods in Matrix Computations*. Texts in Applied Mathematics, vol. 59, p. 800. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-05089-8>
9. Scott, J., Tůma, M.: *Algorithms for Sparse Linear Systems*. Nečas Center Series, p. 242. Birkhäuser/Springer, Cham (2023). <https://doi.org/10.1007/978-3-031-25820-6>
10. Niklasson, A.M.N., Tymczak, C.J., Challacombe, M.: Trace resetting density matrix purification in $O(N)$ self-consistent-field theory. *J. Chem. Phys.* **118**(19), 8611–8620 (2003). <https://doi.org/10.1063/1.1559913>
11. Niklasson, A.M.N.: Density matrix methods in linear scaling electronic structure theory. In: Zalesny, R., Papadopoulos, M. G., Mezey, P.G., Leszczynski, J. (eds.) *Linear Scaling Techniques in Computational Chemistry and Physics*. Springer, Dordrecht, pp. 439–473 (2011)
12. Frommer, A., Maass, P.: Fast CG-based methods for Tikhonov–Phillips regularization. *SIAM J. Sci. Comput.* **20**(5), 1831–1850 (1999). <https://doi.org/10.1137/S1064827596313310>
13. Di Napoli, E., Polizzi, E., Saad, Y.: Efficient estimation of eigenvalue counts in an interval. *Numer. Linear Algebra Appl.* **23**(4), 674–692 (2016). <https://doi.org/10.1002/nla.2048>
14. Lin, L., Saad, Y., Yang, C.: Approximating spectral densities of large matrices. *SIAM Rev.* **58**(1), 34–65 (2016). <https://doi.org/10.1137/130934283>
15. Braverman, V., Krishnan, A., Musco, C.: Sublinear time spectral density estimation. In: *STOC '22—Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1144–1157 (2022). <https://doi.org/10.1145/3519935.3520009>
16. Chen, T., Trogdon, T., Ubaru, S.: Analysis of stochastic Lanczos quadrature for spectrum approximation. In: *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 1728–1739 (2021). <https://proceedings.mlr.press/v139/chen21s.html>
17. Silver, R., Röder, H.: Densities of states of mega-dimensional Hamiltonian matrices. *Int. J. Mod. Phys. C* **5**(04), 735–753 (1994). <https://doi.org/10.1142/S0129183194000842>
18. Wang, L.-W.: Calculating the density of states and optical-absorption spectra of large quantum systems by the plane-wave moments method. *Phys. Rev. B* **49**(15), 10154 (1994). <https://doi.org/10.1103/PhysRevB.49.10154>
19. Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun. Stat. Simul. Comput.* **18**(3), 1059–1076 (1989). <https://doi.org/10.1080/03610918908812806>
20. Cortinovis, A., Kressner, D.: On randomized trace estimates for indefinite matrices with an application to determinants. *Found. Comput. Math.* **22**(3), 875–903 (2022). <https://doi.org/10.1007/s10208-021-09525-9>
21. Roosta-Khorasani, F., Ascher, U.: Improved bounds on sample size for implicit matrix trace estimators. *Found. Comput. Math.* **15**(5), 1187–1212 (2015). <https://doi.org/10.1007/s10208-014-9220-1>
22. Meyer, R.A., Musco, C., Musco, C., Woodruff, D.P.: Hutch++: optimal stochastic trace estimation. In: *Symposium on Simplicity in Algorithms (SOSA)*, pp. 142–155. SIAM, Philadelphia (2021). <https://doi.org/10.1137/1.9781611976496.16>
23. Persson, D., Cortinovis, A., Kressner, D.: Improved variants of the Hutch++ algorithm for trace estimation. *SIAM J. Matrix Anal. Appl.* **43**(3), 1162–1185 (2022). <https://doi.org/10.1137/21M1447623>
24. Epperly, E.N., Tropp, J.A., Webber, R.J.: XTrace: making the most of every sample in stochastic trace estimation. *SIAM J. Matrix Anal. Appl.* **45**(1), 1–23 (2024). <https://doi.org/10.1137/23M1548323>
25. Frommer, A., Schimmel, C., Schweitzer, M.: Analysis of probing techniques for sparse approximation and trace estimation of decaying matrix functions. *SIAM J. Matrix Anal. Appl.* **42**(3), 1290–1318 (2021). <https://doi.org/10.1137/20M1364461>
26. Benzi, M., Rinelli, M., Simuncic, I.: Computation of the von Neumann entropy of large matrices via trace estimators and rational Krylov methods. *Numer. Math.* **155**(3–4), 377–414 (2023). <https://doi.org/10.1007/s00211-023-01368-6>

27. Frommer, A., Rinelli, M., Schweitzer, M.: Analysis of stochastic probing methods for estimating the trace of functions of sparse symmetric matrices. *Math. Comput.* **94**(352), 801–823 (2025). <https://doi.org/10.1090/mcom/3984>
28. Saad, Y.: *Iterative Methods for Sparse Linear Systems*, 2nd edn., p. 528. SIAM, Philadelphia, PA (2003). <https://doi.org/10.1137/1.9780898718003>
29. Golub, G.H., Meurant, G.: *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics, p. 363. Princeton University Press, Princeton, NJ (2010)
30. Dhillon, I.S., Parlett, B.N.: Orthogonal eigenvectors and relative gaps. *SIAM J. Matrix Anal. Appl.* **25**(3), 858–899 (2003). <https://doi.org/10.1137/S0895479800370111>
31. Dhillon, I.S., Parlett, B.N.: Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices. *Linear Algebra Appl.* **387**, 1–28 (2004). <https://doi.org/10.1016/j.laa.2003.12.028>
32. Dhillon, I.S., Parlett, B.N., Vömel, C.: Glued matrices and the MRRR algorithm. *SIAM J. Sci. Comput.* **27**(2), 496–510 (2005). <https://doi.org/10.1137/040620746>
33. Güttel, S.: Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. *GAMM-Mitt.* **36**(1), 8–31 (2013). <https://doi.org/10.1002/gamm.201310002>
34. Chen, T., Greenbaum, A., Musco, C., Musco, C.: Error bounds for Lanczos-based matrix function approximation. *SIAM J. Matrix Anal. Appl.* **43**(2), 787–811 (2022). <https://doi.org/10.1137/21M1427784>
35. Eremenko, A., Yuditskii, P.: Uniform approximation of $\operatorname{sgn} x$ by polynomials and entire functions. *J. Anal. Math.* **101**, 313–324 (2007). <https://doi.org/10.1007/s11854-007-0011-3>
36. Eremenko, A., Yuditskii, P.: Polynomials of the best uniform approximation to $\operatorname{sgn}(x)$ on two intervals. *J. Anal. Math.* **114**, 285–315 (2011). <https://doi.org/10.1007/s11854-011-0018-7>
37. Kuijlaars, A.B.J.: Convergence analysis of Krylov subspace iterations with methods from potential theory. *SIAM Rev.* **48**(1), 3–40 (2006). <https://doi.org/10.1137/S0036144504445376>
38. Gautschi, W.: *Orthogonal Polynomials: Computation and Approximation*. Numerical Mathematics and Scientific Computation, p. 301. Oxford University Press, New York (2004)
39. van den Eshof, J., Frommer, A., Lippert, T., Schilling, K., van der Vorst, H.A.: Numerical methods for the QCDd overlap operator. I. Sign-function and error bounds. *Comput. Phys. Commun.* **146**(2), 203–224 (2002). [https://doi.org/10.1016/S0010-4655\(02\)00455-1](https://doi.org/10.1016/S0010-4655(02)00455-1)
40. Paige, C.C., Parlett, B.N., van der Vorst, H.A.: Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numer. Linear Algebra Appl.* **2**(2), 115–133 (1995). <https://doi.org/10.1002/nla.1680020205>
41. Li, L., Hu, J., Zhang, Y.: One-dimension periodic potentials in Schrödinger equation solved by the finite difference method. *Eur. J. Phys.* **46**(1), 015402 (2024). <https://doi.org/10.1088/1361-6404/ad8d23>
42. Biborski, A., Kadzielawa, A.P., Spatek, J.: Atomization of correlated molecular-hydrogen chain: a fully microscopic variational Monte Carlo solution. *Phys. Rev. B* **98**(8), 085112 (2018). <https://doi.org/10.1103/PhysRevB.98.085112>
43. Sun, Q., Berkelbach, T.C., Blunt, N.S., Booth, G.H., Guo, S., Li, Z., Liu, J., McClain, J.D., Sayfutyarova, E.R., Sharma, S., Wouters, S., Chan, G.K.-L.: PySCF: the python-based simulations of chemistry framework. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **8**(1) (2018). <https://doi.org/10.1002/wcms.1340>

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