Numerische Mathematik (2023) 155:377–414
https://doi.org/10.1007/s00211-023-01368-6

Numerische Mathematik

Computation of the von Neumann entropy of large matrices via trace estimators and rational Krylov methods

Michele Benzi[1] · **Michele Rinelli**[1] · **Igor Simunec**[1]

Received: 19 December 2022 / Accepted: 5 August 2023 / Published online: 28 September 2023

Abstract
We consider the problem of approximating the von Neumann entropy of a large, sparse, symmetric positive semidefinite matrix A, defined as $\mathrm{tr}(f(A))$ where $f(x) = -x \log x$. After establishing some useful properties of this matrix function, we consider the use of both polynomial and rational Krylov subspace algorithms within two types of approximations methods, namely, randomized trace estimators and probing techniques based on graph colorings. We develop error bounds and heuristics which are employed in the implementation of the algorithms. Numerical experiments on density matrices of different types of networks illustrate the performance of the methods.

Mathematics Subject Classification Primary 65F60 · 15A16

1 Introduction

The von Neumann entropy [56] of a symmetric positive semidefinite matrix A with $\mathrm{tr}(A) = 1$ is defined as $S(A) = \mathrm{tr}(-A \log A)$, where $\log A$ is the matrix logarithm. With the usual convention that $0 \cdot \log 0 = 0$, the von Neumann entropy is given by

$$S(A) = -\sum_{i=1}^{n} \lambda_i \log \lambda_i,$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of the $n \times n$ matrix A. The von Neumann entropy plays an important role in several fields including quantum statistical mechanics [42],

✉ Michele Rinelli
michele.rinelli@sns.it

Michele Benzi
michele.benzi@sns.it

Igor Simunec
igor.simunec@sns.it

[1] Scuola Normale Superiore, Piazza dei Cavalieri, 7, 56126 Pisa, Italy

quantum information theory [5], and network science [15]. For example, computing the von Neumann entropy is necessary in order to determine the ground state of many-electron systems at finite temperature [1]. The von Neumann entropy of graphs is also an important tool in the structural characterization and comparison of complex networks [22, 36].

If the size of A is large, the computation of $S(A)$ by means of explicit diagonalization can be too expensive, so it becomes necessary to resort to cheaper methods that compute approximations of the entropy. In recent years, a few papers have appeared devoted to this problem; see, e.g., [17, 31, 44, 59]. These papers investigate different approaches based on quadratic (Taylor) approximants, the global Lanczos algorithm, Gaussian quadrature and Chebyshev expansion. In general, the problem of computing the von Neumann entropy of a large matrix is difficult because the underlying matrix function is not analytic in a neighbourhood of the spectrum when the matrix is singular; difficulties can also be expected when A has eigenvalues close to zero, which is usually the case. In particular, polynomial approximation methods may converge slowly in such cases.

In this paper we propose to approximate the von Neumann entropy using either the probing approach developed in [29] or a stochastic trace estimator [39, 47, 49]. The main contributions of this work are outlined in the following.

In Sect. 2.1 we obtain an integral expression for the entropy function $f(x) = -x \log x$ that relates it with a Cauchy-Stieltjes function [58, Chapter VIII], and we use it to derive error bounds for the polynomial approximation of f, which in turn lead to a priori bounds for the approximation of $S(A)$ with probing methods. In order to also have a practical stopping criterion alongside the theoretical bounds, in Sect. 5.1 we propose some heuristics to estimate the error of probing methods, and in Sect. 6.2 we demonstrate their reliability with numerical experiments. In Sect. 3.1 we also use properties of symmetric M-matrices to show that, in the case of the graph entropy, the approximation obtained with a probing method is a lower bound for the exact entropy.

Both probing methods and stochastic trace estimators require the computation of a large number of quadratic forms with $f(A)$, which can be efficiently approximated using Krylov subspace methods [30, 35]. In Sect. 4.2 we propose to combine polynomial Krylov iterations with rational Krylov iterations that use asymptotically optimal poles for Cauchy-Stieltjes functions [45]. In Sect. 4.3 we obtain new a posteriori error bounds and estimates for this task, building on the ones presented in [34], and we discuss methods to compute them efficiently. For the case of the graph entropy, we make use of a desingularization technique introduced in [11] that exploits properties of the graph Laplacian to compute quadratic forms more efficiently.

The resulting algorithm can be seen as a black box method that only requires an input tolerance ϵ and computes an approximation of $S(A)$ with relative accuracy ϵ. While this accuracy is not guaranteed when the rigorous bounds are replaced by heuristics, we found the algorithm to be quite reliable in practice.

Our implementation of the probing algorithm is compared to a state-of-the-art randomized trace estimator developed in [49] with several numerical experiments in Sect. 6, in which we approximate the graph entropy of various complex networks.

The rest of the paper is organized as follows. In Sect. 2 we recall some properties of the von Neumann entropy and we obtain bounds for the polynomial approxima-

tion of $f(x) = -x \log x$, which we use in the subsequent sections. In Sect. 3 we give an overview of probing methods and stochastic trace estimators and we derive bounds for the convergence of probing methods. In Sect. 4 we describe the computation of quadratic forms using Krylov subpsace methods and we derive a posteriori error bounds and estimates. In Sect. 5 we summarize the overall algorithm and we discuss heuristics and stopping criteria, and in Sect. 6 we test the performance of the methods on density matrices of several complex networks. Finally, Sect. 7 contains some concluding remarks.

2 Properties of the von Neumann entropy

A *density matrix* ρ is a self-adjoint, positive semi-definite linear operator with unit trace acting on a complex Hilbert space. In quantum mechanics, density operators describe states of quantum mechanical systems. Here we consider only finite dimensional Hilbert spaces, so ρ is an $n \times n$ Hermitian matrix. For simplicity, here we focus on real symmetric matrices; all the results are easily extended to the complex case.

Recall that the *von Neumann entropy* of a system described by the density matrix ρ [56] is given by

$$S(\rho) = -\sum_{\lambda \in \sigma(\rho)} \lambda \log(\lambda) = -\operatorname{tr}(\rho \log \rho), \tag{2.1}$$

where $\sigma(\rho)$ denotes the spectrum of ρ, under the convention that $0 \cdot \log(0) = 0$. Here $\log(x)$ is the natural logarithm. Note that in the literature the entropy is sometimes defined using $\log_2(x)$ instead, but since $\log_2(x) = \log(x)/\log(2)$ the two definitions are equivalent up to a scaling factor. We also mention that in the original definition the entropy includes the factor κ_B (the Boltzmann constant), which we omit.

A straightforward way to compute $S(\rho)$ is by diagonalization. However, this approach is unfeasible when the dimension is very large. Here we propose some approaches to compute approximations to the von Neumann entropy based on the trace estimation of matrix functions.

Note that the formula (2.1) is well defined even without the hypothesis of unit trace. Moreover, if ρ is given in the form $\rho = \gamma A$ with $\gamma > 0$, we have the relation

$$S(\rho) = -\gamma \operatorname{tr}(A \log(A)) - \gamma \log(\gamma) \operatorname{tr}(A) = \gamma S(A) - \gamma \log(\gamma) \operatorname{tr}(A). \tag{2.2}$$

In quantum mechanics, the von Neumann entropy of a density matrix gives a measure of how much a density matrix is distant from being a *pure state*, that is a rank-1 matrix with only one nonzero eigenvalue equal to one [57]. For any density matrix of size n, we have $0 \le S(\rho) \le \log(n)$. Moreover, $S(\rho) = 0$ if and only if ρ is a pure state and $S(\rho) = \log(n)$ if and only if $\rho = \frac{1}{n} I$ [56, 57].

The von Neumann entropy also has applications in network theory. Recall that a graph $\mathcal{G}$ is given by a set of nodes $\mathcal{V} = \{v_1, \ldots, v_n\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The graph is called *undirected* if $(v_i, v_j) \in \mathcal{E}$ if and only if $(v_j, v_i) \in \mathcal{E}$ for all $i \neq j$. A *walk* on the graph of length ℓ is a sequence of $\ell + 1$ nodes where two consecutive

nodes are linked by an edge, for a total of ℓ edges. We say that $\mathcal{G}$ is *connected* if there exists a directed path from node i to node j for any pair of nodes (i, j). The *adjacency matrix* of a graph is a matrix $A \in \mathbb{R}^{n\times n}$ such that $[A]_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, and 0 otherwise. The *degree* of a node $\deg(v_i)$ is the number of nodes that are linked with v_i by an edge. It can be expressed in the form $\deg(v_i) = [A\mathbf{1}]_i$, where $\mathbf{1}$ is the vector of all ones. Finally, the *graph Laplacian* is given by

$$\mathcal{L} = D - A, \quad D = \operatorname{diag}((\deg v_i)_{i=1}^n) = \operatorname{diag}(A\mathbf{1}). \tag{2.3}$$

Note that $\mathcal{L}$ is a singular positive semidefinite matrix and the eigenvalue 0 is simple if and only if $\mathcal{G}$ is connected, in which case the associated one-dimensional eigenspace is spanned by $\mathbf{1}$. Given the density matrix $\rho = \mathcal{L}/\operatorname{tr}(\mathcal{L})$, the von Neumann entropy of $\mathcal{G}$ is defined as $S(\rho)$ [15, 17].

It should be mentioned that, strictly speaking, the graph entropy defined in this manner is not a "true" entropy, since it does not satisfy the sub-additivity requirement. In other words, the graph entropy could decrease when an edge is added to the graph, see [23]. For this reason, different notions of graph entropy have been proposed in the literature; see, e.g., [32, 33]. These entropies still have the form of a von Neumann entropy, $S = -\operatorname{tr}(\rho \log \rho)$, but with a different definition of the density matrix ρ which, however, is still expressed as a function of a matrix (Hamiltonian) associated to the graph. Therefore, the techniques developed in this paper are still applicable, at least in principle.

2.1 Polynomial approximation

Denote the set of all polynomials with degree at most k with Π_k. For a continuous function $f : [a, b] \to \mathbb{R}$, define the error of the best uniform polynomial approximation in Π_k as

$$E_k(f, [a, b]) = \min_{p\in\Pi_k} \max_{x\in[a,b]} |f(x) - p(x)|. \tag{2.4}$$

If A is a symmetric (or Hermitian) matrix with $\sigma(A) \subset [a, b]$, estimating or bounding this quantity is crucial in the study of polynomial approximations of the matrix function $f(A)$ [4] and for establishing decay bounds for the entries of $f(A)$ [8, 24, 27].

An important case is the inverse function $f(x) = 1/x$ for $x \in [a, b]$, $0 < a < b$, for which we have

$$E_k(1/x, [a, b]) = \frac{(\sqrt{\kappa}+1)^2}{2b}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{k+1}, \tag{2.5}$$

where $\kappa = b/a$; see, e.g., [46].

In general, an inequality of the form

$$E_k(f, [a, b]) \le Cq^k \tag{2.6}$$

for some fixed constants $C > 0$ and $0 < q < 1$, is equivalent to the fact that f is analytic over an ellipse containing $[a, b]$; see [46, Theorem 73] for more details.

The function $f(x) = x \log(x)$ is not analytic on any neighborhood of 0, so we cannot expect to find a geometric decay as in (2.6) if we consider $[0, b]$, $b > 0$, as the interval of definition. However, since f is continuous, by the Weierstrass Approximation Theorem $E_k(f, [0, b])$ must go to 0 as $k \to \infty$. A more precise estimate is the following [59], derived by computing the coefficients of the Chebyshev expansion of $f(x)$:

$$E_k(f, [0, b]) \leq \frac{b}{2k(k+1)} \quad \text{for all } k \geq 1. \tag{2.7}$$

This shows that the decay rate of the error is algebraic in k. Our approach is based on an integral representation and leads to sharper bounds.

Recall that a Cauchy-Stieltjes function [10, 45] has the form

$$f(z) = \int_0^\infty \frac{\mathrm{d}\mu(s)}{s+z}, \quad z \in \mathbb{C} \setminus [0, +\infty), \tag{2.8}$$

where μ is a (possibly signed) real measure supported on $[0, +\infty)$ and the integral is absolutely convergent. An example is given by the function $\log(1+z)/z$ that has the expression

$$\frac{\log(1+z)}{z} = \int_1^\infty \frac{1}{s(s+z)} \mathrm{d}s.$$

It is easy to check that the above identity also holds for $z \in (-1, 0)$. With the change of variable $x = 1 + z$ and some simple rearrangements, we get the following integral expression for the entropy:

$$-x \log(x) = \int_1^\infty \frac{x(1-x)}{s(s+x-1)} \mathrm{d}s, \qquad x \in [0, 1]. \tag{2.9}$$

With the additional change of variable $s = t + 1$, we can rewrite (2.9) in the form

$$-x \log(x) = x(1-x) \int_0^\infty \frac{1}{(t+x)(t+1)} \mathrm{d}t, \qquad x \in [0, 1]. \tag{2.10}$$

Note that the above identities also hold for $x = 0$ and $x = 1$, because of the factor $x(1-x)$ in front of the integral. This shows that although the entropy function $-x \log(x)$ is not itself a Cauchy-Stieltjes function, we can recognize a factor with the form (2.8) in its integral representation. This observation will be important later for the selection of poles in a rational Krylov method; see Sect. 4.2.

Integral representations have proved useful for many problems related to polynomial approximations and matrix functions; see e.g. [9, 10, 27]. The following result shows that we can derive explicit bounds for $E_k(f, [a, b])$.

Lemma 2.1 *Let $f(x)$ be defined for $x \in [a, b]$ by*

$$f(x) = \int_0^\infty g_t(x)\,\mathrm{d}t, \tag{2.11}$$

where $g_t(x)$ is continuous for $(t, x) \in (0, \infty) \times [a, b]$ and the integral (2.11) is absolutely convergent. Then

$$E_k(f, [a, b]) \le \int_0^\infty E_k(g_t(x), [a, b])\,\mathrm{d}t, \tag{2.12}$$

for all $k \ge 0$.

Proof For all $t \in (0, \infty)$ and for any degree $k \ge 0$, since $g_t(x)$ is continuous over the compact interval $[a, b]$ there exists a unique polynomial $p_k^{(t)}(x) \in \Pi_k$ such that

$$E_k(g_t(x), [a, b]) = \max_{x\in[a,b]} |g_t(x) - p_k^{(t)}(x)|.$$

For all $x \in [a, b]$ we can define

$$p_k(x) := \int_0^\infty p_k^{(t)}(x)\,\mathrm{d}t.$$

This is well defined: for all $x \in [a, b]$, the mapping $t \mapsto p_k^{(t)}(x)$ is continuous for all $t \in (0, \infty)$ in view of [46, Theorem 24], since $g_t(x)$ is uniformly continuous for (t, x) in the compact sets contained in $(0, \infty) \times [a, b]$. Moreover, the integral is finite since

$$\begin{aligned}\int_0^\infty |p_k^{(t)}(x)|\mathrm{d}t &\le \int_0^\infty |g_t(x)|\mathrm{d}t + \int_0^\infty |g_t(x) - p_k^{(t)}(x)|\mathrm{d}t \\ &\le \int_0^\infty |g_t(x)|\mathrm{d}t + \int_0^\infty E_k(g_t(x), [a, b])\mathrm{d}t < +\infty.\end{aligned}$$

We want to show that $p_k(x)$ is also a polynomial. Consider the expression

$$p_k^{(t)}(x) = \sum_{i=0}^{k} a_i(t)\, x^i,$$

which defines the coefficients $a_i(t)$ as functions of t. Formally, we have

$$p_k(x) = \sum_{i=0}^{k} \left(\int_0^\infty a_i(t)\,\mathrm{d}t \right) \cdot x^i = \sum_{i=0}^{k} \bar{a}_i x^i,$$

where $\bar{a}_i := \int_0^\infty a_i(t)\,\mathrm{d}t$. To conclude, we need to show that this expression is well defined, that is, the functions $a_i(t)$ are integrable in t. Consider $k+1$ distinct points $x_0, \ldots, x_k$ in the interval $[a, b]$ and let $\boldsymbol{a}(t) = [a_0(t), \ldots, a_k(t)]^T$, $\boldsymbol{p}(t) = [p_k^{(t)}(x_0), \ldots, p_k^{(t)}(x_k)]^T$. We have that $V\boldsymbol{a}(t) = \boldsymbol{p}(t)$, where

$$V = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{bmatrix}$$

is a Vandermonde matrix. Since V is nonsingular, we have that $\boldsymbol{a}(t) = V^{-1}\boldsymbol{p}(t)$. Then, if we let $V^{-1} = (c_{ij})_{i,j=0}^k$, we obtain the expression

$$a_i(t) = \sum_{j=0}^{k} c_{ij}\, p_k^{(t)}(x_j), \quad i = 0, \ldots, k,$$

where c_{ij} is independent of t for all i, j. This shows that, for all i, $a_i(t)$ is continuous for $t \in (0, \infty)$, and we have

$$|\bar{a}_i| \le \int_0^\infty |a_i(t)|\,\mathrm{d}t \le \sum_{j=0}^{k} |c_{ij}| \int_0^\infty \left| p_k^{(t)}(x_j) \right| \mathrm{d}t < +\infty, \quad i = 0, \ldots, k,$$

hence $\bar{a}_i$ is well defined for $i = 0, \ldots, k$ and $p_k(x)$ is a polynomial of degree at most k. Finally, we have

$$\begin{aligned} E_k(f, [a, b]) &\le \max_{x\in[a,b]} |f(x) - p_k(x)| \\ &\le \max_{x\in[a,b]} \int_0^\infty |g_t(x) - p_k^{(t)}(x)|\,\mathrm{d}t \\ &\le \int_0^\infty E_k(g_t(x), [a, b])\,\mathrm{d}t. \end{aligned}$$

This concludes the proof. □

The following result gives a new bound for $E_k(x \log x, [a, b])$.

Theorem 2.2 *Let $0 \le a < b$ and $\gamma = a/b$. We have*

$$E_k(x\log(x), [a, b]) \le b\,(1 - \sqrt{\gamma})\frac{1 + \gamma + 2k\sqrt{\gamma}}{4(k^2 - 1)} \left(\frac{1 - \sqrt{\gamma}}{1 + \sqrt{\gamma}}\right)^k, \qquad (2.13)$$

for all $k \ge 2$.

Proof Let $f(x) = x\log(x)$. Notice that $f(x) = bf(b^{-1}x) + \log(b)x$, so

$$E_k(f(x),[a,b]) = E_k(bf(b^{-1}x),[a,b]) = b\,E_k(f(x),[\gamma,1])$$

since $k \geq 2$ and we can ignore terms of degree 1 for the polynomial approximation. For $x \in [\gamma, 1]$ we can use the representation (2.10). Let $g_t(x) := \frac{x(1-x)}{(1+t)(x+t)}$ be the integrand in (2.10) for all $t > 0$. Note that $g_t(x)$ is a continuous fuction in the variables $(t, x) \in (0, \infty) \times [a, b]$, so we can apply Lemma 2.1. Then we can write $g_t(x)$ as

$$g_t(x) = \frac{x}{x+t} - \frac{x}{1+t} = 1 - \frac{t}{x+t} - \frac{x}{1+t}, \tag{2.14}$$

and since $k \geq 2$ and $1 - \frac{x}{1+t}$ has degree 1, we get

$$E_k(g_t(x),[\gamma,1]) = E_k(t/(x+t),[\gamma,1]) = tE_k(1/x,[\gamma+t,1+t]).$$

Hence, by using (2.5) and Lemma 2.1 we get

$$E_k(x\log(x),[a,b]) \leq b\int_0^\infty \frac{t\left(\sqrt{\kappa(t)}+1\right)^2}{2(1+t)}\left(\frac{\sqrt{\kappa(t)}-1}{\sqrt{\kappa(t)}+1}\right)^{k+1} \mathrm{d}t, \tag{2.15}$$

where $\kappa(t) = (1+t)/(\gamma+t)$. In order to bound the integral, consider the identities

$$\frac{\sqrt{\kappa(t)}-1}{\sqrt{\kappa(t)}+1} = \frac{\left(\sqrt{1+t}-\sqrt{\gamma+t}\right)^2}{1-\gamma}, \quad \left(\sqrt{\kappa(t)}+1\right)^2 = \frac{\left(\sqrt{1+t}+\sqrt{\gamma+t}\right)^2}{\gamma+t}.$$

We have

$$\begin{aligned}
&\int_0^\infty \frac{t\left(1+\sqrt{\kappa(t)}\right)^2}{2(1+t)}\left(\frac{\sqrt{\kappa(t)}-1}{\sqrt{\kappa(t)}+1}\right)^{k+1} \mathrm{d}t \\
&= \frac{1}{2}\int_0^\infty \frac{t}{(\gamma+t)(1+t)} \cdot \frac{\left(\sqrt{1+t}+\sqrt{\gamma+t}\right)^2\left(\sqrt{1+t}-\sqrt{\gamma+t}\right)^{2k+2}}{(1-\gamma)^{k+1}} \mathrm{d}t \\
&\leq \frac{1}{2(1-\gamma)^{k-1}}\int_0^\infty \left(\sqrt{1+t}-\sqrt{\gamma+t}\right)^{2k} \mathrm{d}t.
\end{aligned} \tag{2.16}$$

By checking the derivative, it can be shown that

$$F(t) = \frac{\sqrt{1+t}\sqrt{\gamma+t}\left(\sqrt{1+t}-\sqrt{\gamma+t}\right)^{2k}\left(\frac{\left(\sqrt{1+t}-\sqrt{\gamma+t}\right)^4}{2k+2} - \frac{(1-\gamma)^2}{2k-2}\right)}{2\left(\sqrt{1+t}\sqrt{\gamma+t}-\gamma-t\right)\left(1+t-\sqrt{1+t}\sqrt{\gamma+t}\right)}$$

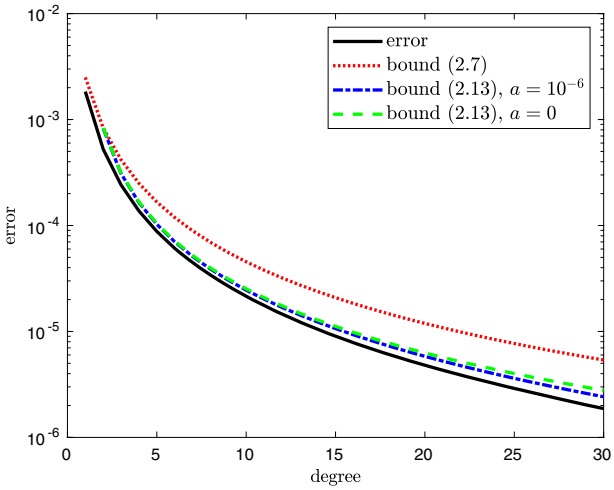

Fig. 1 Comparison of the bounds (2.7) and (2.13) with the error of the polynomial approximation of the entropy function $x \log x$ on the interval $[10^{-6}, 10^{-2}]$

is an antiderivative of $\left(\sqrt{1+t} - \sqrt{\gamma+t}\right)^{2k}$. Since $\lim_{t\to\infty} F(t) = 0$, we deduce that

$$\int_0^\infty \left(\sqrt{1+t} - \sqrt{\gamma+t}\right)^{2k} \mathrm{d}t = \frac{\sqrt{\gamma}\left(1-\sqrt{\gamma}\right)^{2k}\left(\frac{(1-\gamma)^2}{2k-2} - \frac{\left(1-\sqrt{\gamma}\right)^4}{2k+2}\right)}{2\left(1-\sqrt{\gamma}\right)\left(\sqrt{\gamma}-\gamma\right)}$$
$$= (1-\sqrt{\gamma})^{2k}\frac{1+\gamma+2\sqrt{\gamma}\,k}{2(k^2-1)}.$$

This, combined with (2.16), concludes the proof. □

Remark 2.3 The result of Theorem 2.2 is an improvement of (2.7). When $a > 0$, the right-hand side in (2.13) is the product of an algebraic and a geometric factor, so the decay is asymptotically faster. When $a = 0$, we have $\gamma = 0$ and (2.13) becomes

$$E_k(x\log(x), [0, b]) \le \frac{b}{4(k^2-1)},$$

which is better than (2.7) for all $k \ge 2$. The new bound (2.13) is compared with (2.7) in Fig. 1.

3 Trace estimation

Recall that a function $f(A)$ of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ can be defined via a spectral decomposition $A = U\Lambda U^T$, where U is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is

the diagonal matrix containing the eigenvalues. Then

$$f(A) := Uf(\Lambda)U^T, \quad f(\Lambda) := \mathrm{diag}(f(\lambda_1), \ldots, f(\lambda_n)),$$

provided that f is defined on the eigenvalues of A. A matrix function can be computed via diagonalization, using polynomial or rational approximants, or with algorithms tailored to specific matrix functions, such as scaling and squaring for the matrix exponential. We direct the reader to [38] for more details on these methods. In this section we present algorithms to estimate the trace of $f(A)$ that do not require the explicit computation of the diagonal entries of $f(A)$. It is necessary to resort to this class of algorithms whenever the size of A is too large to make the computation of $f(A)$ (or its diagonal) feasible. In the sections that follow, we use the notation A to denote general matrices and ρ to denote density matrices, since most of our results are applicable in general for symmetric positive semidefinite matrices and not only to density matrices.

3.1 Probing methods

Let us summarize the approach described in [29] to compute the trace of a matrix function $f(A)$, where $A \in \mathbb{R}^{n\times n}$ is a sparse matrix. Recall that the graph $\mathcal{G}(A)$ associated with A has nodes $\mathcal{V} = \{1, \ldots, n\}$ and edges $\mathcal{E} = \{(i, j) : [A]_{ij} \neq 0, i \neq j\}$, and the *geodesic distance* $\mathrm{d}(i, j)$ between i and j is the shortest length of a walk that starts with i and ends with j, and is ∞ if no such walk exists or 0 if $i = j$.

Let $V_1, \ldots, V_s$ be a coloring (i.e. a partitioning) of the set $\{1, \ldots, n\}$. We write $\mathrm{col}(i) = \mathrm{col}(j)$ if and only if i and j belong to the same set. We have a *distance-d coloring* if $\mathrm{col}(i) \neq \mathrm{col}(j)$ whenever $\mathrm{d}(i, j) \leq d$, where $\mathrm{d}(i, j)$ is the geodesic distance between i and j in the graph $\mathcal{G}(A)$ associated with A. The associated *probing vectors* are

$$\boldsymbol{v}_\ell = \sum_{i\in V_\ell} \boldsymbol{e}_i \in \mathbb{R}^n, \quad \ell = 1, \ldots, s, \tag{3.1}$$

where $\boldsymbol{e}_i$ is the i-th vector of the canonical basis. The induced approximation of the trace is

$$\mathcal{T}_d(f(A)) := \sum_{\ell=1}^{s} \boldsymbol{v}_\ell^T f(A)\boldsymbol{v}_\ell. \tag{3.2}$$

The problem of finding the minimum number s of colors, or sets, in the partition needed to get a distance-d coloring for a fixed d is NP-complete for general graphs [41]. It is important to keep s as small as possible since, in view of (3.2), it is the number of quadratic forms needed to compute $\mathcal{T}_d(f(A))$. A greedy and efficient algorithm to get a quasi optimal distance-d coloring is given by Algorithm 1 [53, Algorithm 4.2].

One can obtain different colorings depending on the order of the nodes; sorting the nodes by descending degree usually leads to a good performance [41, 53]. The number of colors obtained with this algorithm is at most $\Delta^d + 1$, where Δ is the

Algorithm 1 Greedy algorithm for a distance-d coloring

Input: Graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ and distance d
Output: Distance-d coloring col
1: $\text{col}(1) = 1$
2: **for** $i = 2 : n$ **do**
3: $\quad W_i = \{j \in \{1, \ldots, i-1\} : \text{d}(i, j) \leq d\}$
4: $\quad \text{col}(i) = \min\{k > 0 : k \neq \text{col}(j) \text{ for all } j \in W_i\}$
5: **end for**

maximum degree of a node in the graph induced by A, and the cost is at most $\mathcal{O}(n\,\Delta^d)$ if the d-neighbors W_i of each node i are computed with a traversal of the graph. Alternatively, the greedy coloring can be obtained by computing A^d, which can be done using at most $2\lfloor \log_2 d \rfloor$ matrix-matrix multiplications [38, Section 4.1]. In our Matlab implementation of the probing method, we construct the greedy distance-d coloring by computing A^d, since we found it to be faster than the graph-based approach; this is likely due to the more efficient Matlab implementation of matrix-matrix operations.

If A is β-banded, i.e. if $[A]_{ij} = 0$ for $|i - j| > \beta$, a simple distance-d coloring is given by

$$\text{col}(i) = (i-1) \bmod (d\beta + 1) + 1, \quad i = 1, \ldots, n. \tag{3.3}$$

This coloring is optimal with $s = d\beta + 1$ if all the entries within the band are nonzero. Note that to diagonalize a symmetric banded matrix one must first reduce it to a tridiagonal form, and this costs $O(\beta n^2)$ [54]. On the other hand, if we assume that quadratic forms with $f(A)$ can be approximated with a fixed number of Krylov iterations, the computation of each quadratic form costs $O(\beta n)$, and the cost for $\mathcal{T}_d(f(A))$ is approximatively $O(d\beta^2 n)$ and this can be much less than the cost for the diagonalization provided that the requested accuracy is not too high. For a general sparse matrix A, as an alternative to Algorithm 1 one can use the reverse Cuthill-McKee algorithm [20] to reorder the nodes and reduce the bandwidth, and then use (3.3) to find a distance-d coloring [29]. Depending on the structure of the graph induced by A this can yield good results, but in many cases better colorings are obtained by using Algorithm 1; see Example 3.4.

Regarding the accuracy of the approximation of $\text{tr}(f(A))$ by $\mathcal{T}_d(f(A))$ we have the following result [29, Theorem 4.4].

Theorem 3.1 *([29]) Let $A \in \mathbb{R}^{n\times n}$ be symmetric with spectrum $\sigma(A) \subset [a, b]$. Let $f(x)$ be defined over $[a, b]$ and let $\mathcal{T}_d(f(A))$ be the approximation (3.2) of $\text{tr}(f(A))$ induced by a distance-d coloring. Then*

$$|\text{tr}(f(A)) - \mathcal{T}_d(f(A))| \leq 2n\, E_d(f, [a, b]). \tag{3.4}$$

If $f(x) = -x \log(x)$, so that $\text{tr}(f(A)) = S(A)$, we have the following result.

Corollary 3.2 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric with $\sigma(A) \subset [a, b]$, $0 \le a < b$, and put $\gamma = a/b$. Then*

$$|S(A) - \mathcal{T}_d(-A\log(A))| \le n\, b\, (1-\sqrt{\gamma})\frac{1+\gamma+2d\sqrt{\gamma}}{2(d^2-1)}\left(\frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}\right)^d, \quad (3.5)$$

for all $d \ge 2$. In particular, if $a = 0$, we have

$$|S(A) - \mathcal{T}_d(-A\log(A))| \le \frac{nb}{2(d^2-1)}, \quad (3.6)$$

for all $d \ge 2$.

Proof The inequality (3.5) follows from Theorem 2.2 and Theorem 3.1. The inequality (3.6) follows from (3.5) with $\gamma = a/b = 0$. □

Remark 3.3 The bound (3.5) can be pessimistic in practice, especially for large values of d. We will see this in Example 3.4 and in Sect. 6.2. A priori bounds based on polynomial approximations often fail to catch the exact convergence behavior in many problems related to matrix functions since a minimization problem over the spectrum of a matrix is relaxed to the whole spectral interval. This occurs in the convergence of polynomial Krylov methods [43, Section 5.6] and in the decay bounds on the entries of matrix functions [9, 28]. Also, the coloring we get via Algorithm 1 can return far more colors than needed for a distance-d coloring, and this can benefit the convergence in a way that is not predicted by the bound. We will see in Sect. 5.1 a more practical heuristic to predict the error with higher accuracy.

Example 3.4 Let us see how the bound and the convergence of the probing method perform in practice. We use the density matrix $\rho = \mathcal{L}/\operatorname{tr}(\mathcal{L})$, where $\mathcal{L}$ is the Laplacian of the graph `minnesota` from the SuiteSparse Matrix Collection [21]. More precisely, we consider the biggest connected component whose graph Laplacian has size $n = 2640$ and 9244 nonzero entries.

For several values of d, we compute the approximation $\mathcal{T}_d(-\rho\log\rho)$ associated with two different distance-d colorings: the first is obtained by the greedy coloring (Algorithm 1) after sorting the nodes by descending degree, while for the second we use the reverse Cuthill-McKee algorithm to get a 67-banded matrix and then (3.3).

In Fig. 2 we compare $\mathcal{T}_d(-\rho\log(\rho))$ with the value of $S(\rho)$ obtained by diagonalizing ρ, considered as exact. The left plot shows the error in terms of the number of colors (i.e. the number of probing vectors) associated with the distance-d colorings. In the right plot the errors are shown in terms of d together with bound (3.6), where $b = \lambda_{\max}(\rho)$.

For a fixed value of d, the coloring based on the bandwidth provides a smaller error than the greedy one, but it also uses a larger number of colors. Indeed, we can see from the left plot that with the same computational effort the greedy algorithm obtains a smaller error. On the right we can observe that bound (3.6) is close to the error given by the greedy coloring for small values of d, while it fails to catch the convergence behavior for large values of d.

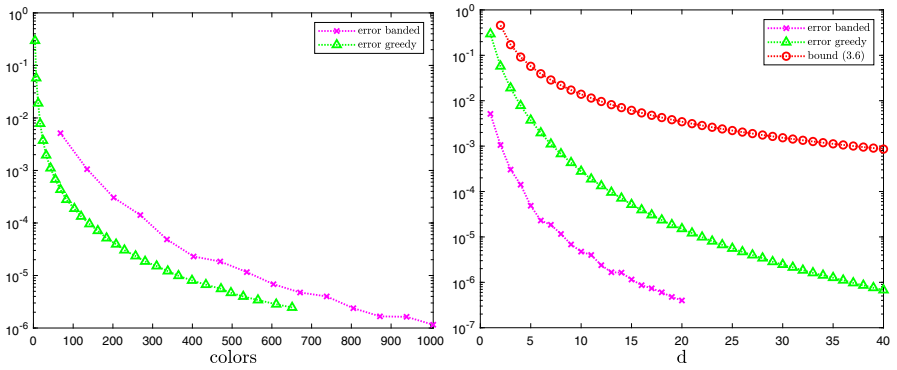

Fig. 2 Absolute errors of the probing approximation of $S(\rho)$ where the distance-d coloring is obtained either by the greedy procedure (Algorithm 1) or with the reverse Cuthill-McKee algorithm and the coloring (3.3) for banded matrices. On the left the abscissa represents the number of colors used for the coloring. On the right the errors are compared with bound (3.6) in terms of d

For the graph entropy, i.e. when $\rho = \mathcal{L}/\operatorname{tr}(\mathcal{L})$ and $\mathcal{L}$ is a graph Laplacian of a graph $\mathcal{G}$, we can show that $-\mathcal{T}_d(\rho \log \rho)$ is a lower bound for $S(\rho)$. In the proof we use the fact that ρ is a symmetric M-matrix [14, Chapter 6], i.e. it can be written in the form $\rho = \theta I - B$ where $\theta > 0$ and B is a nonnegative matrix such that $\lambda \leq \theta$ for all $\lambda \in \sigma(B)$.

Lemma 3.5 *Let $A = \theta I - B$ be a symmetric M-matrix. Let $\mathrm{d}(i, j)$ be the geodesic distance in the graph associated with A. Then $[-A \log(A)]_{ij} \leq 0$ whenever $\mathrm{d}(i, j) \geq 2$. If $\theta < \exp(-1)$, then $[-A \log(A)]_{ij} \leq 0$ for $i \neq j$ and $-A \log A$ is an M-matrix.*

Proof Suppose that $\theta > \|B\|_2 = \lambda_{\max}(B)$, so A is nonsingular. Then $f(x) = -x \log x$ is analytic for $|x - \theta| < \|B\|_2$ and it can be represented by the power series

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\theta)}{k!}(x - \theta)^k.$$

Since $\sigma(A) \subset (\theta - \|B\|_2, \theta + \|B\|_2)$, the series expansion holds for $f(A)$:

$$f(A) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\theta)}{k!}(A - \theta I)^k = -\theta \log \theta I + (1 + \log \theta)B - \sum_{k=2}^{\infty} \frac{\theta^{-(k-1)}}{k(k-1)} B^k.$$

If $\mathrm{d}(i, j) \geq 2$ we have $[B]_{ij} = [-A]_{ij} = 0$, and B^k is nonnegative for all $k \geq 0$, so $[f(A)]_{ij} \leq 0$. If $\theta < \exp(-1)$, then $(1 + \log \theta)B$ is nonpositive so $[f(A)]_{ij} \leq 0$ for $i \neq j$, and it is an M-matrix since it is positive semidefinite [14, Theorem 4.6].

If A is singular, consider the nonsingular M-matrix $A + \epsilon I$ for $\epsilon > 0$. Then the results hold for $f(A + \epsilon I)$ (we impose $\theta + \epsilon < \exp(-1)$ if $\theta < \exp(-1)$) and notice that $f(A) = \lim_{\epsilon \to 0} f(A + \epsilon I)$. This concludes the proof, since the limit of nonsingular M-matrices is an M-matrix. □

Proposition 3.6 *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric M-matrix, and let $\mathcal{T}_d(-A\log(A))$ be the approximation (3.2) of $S(A)$ induced by a distance-d coloring of $\mathcal{G}(A)$ with $d \geq 1$. Then $\mathcal{T}_d(-A\log(A)) \leq S(A)$.*

Proof If $V_1, \ldots, V_s$ is the graph partitioning associated with a distance-d coloring, the error of the approximation can be written as

$$S(A) - \mathcal{T}_d(-A\log(A)) = -\sum_{\ell=1}^{s} \sum_{\substack{i,j\in V_\ell \\ i\neq j}} [-A\log(A)]_{ij}; \tag{3.7}$$

see [29] for more details. By definition of a distance-d coloring, for all $i, j \in V_\ell$, $i \neq j$, we have $\mathrm{d}(i, j) \geq d + 1 \geq 2$. Then, in view of Lemma 3.5, the right-hand side of (3.7) is nonnegative. □

Remark 3.7 In this work we only consider the entropy of sparse density matrices. However, an important case is given by $\rho = g(H)$, where H is the Hamiltonian of a certain quantum system and g is a function defined on the spectrum of H. Notable examples are the Gibbs state [19, 57] and the Fermi-Dirac state [1]. Despite ρ being a dense matrix in general, a sparse structure of H implies that ρ exhibits decay properties and can be well approximated by a sparse matrix [6, 7]. Moreover, since $S(\rho) = -g(H)\log(g(H))$ one can apply the techniques described in this section to the composition $-g(x)\log(g(x))$. Although the investigation of this problem is outside the scope of this work, it represents an interesting direction for future research. These considerations also hold for the randomized techniques discussed below.

3.2 Stochastic trace estimation

Let us consider the problem of computing $\mathrm{tr}(B)$, where $B \in \mathbb{R}^{n\times n}$ is a matrix that is not explicitly available but can be accessed via matrix–vector products $B\boldsymbol{x}$ and quadratic forms $\boldsymbol{x}^T B\boldsymbol{x}$, for $\boldsymbol{x} \in \mathbb{R}^n$. The case of $B = f(A)$ can be seen as an instance of this problem, since $f(A)$ is expensive to compute, but $f(A)\boldsymbol{x}$ and $\boldsymbol{x}^T f(A)\boldsymbol{x}$ can be efficiently approximated using Krylov methods, as we recall in Sect. 4.

Stochastic trace estimators compute approximations of $\mathrm{tr}(B)$ by making use of the fact that, for any matrix B and any random vector $\boldsymbol{x}$ such that $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = I$, we have $\mathbb{E}[\boldsymbol{x}^T B\boldsymbol{x}] = \mathrm{tr}(B)$, where $\mathbb{E}$ denotes the expected value. Hutchinson's trace estimator [39] is a simple stochastic estimator that generates N vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ with i.i.d. random $\mathcal{N}(0, 1)$ entries and approximates $\mathrm{tr}(B)$ with

$$\mathrm{tr}_N^{\mathrm{Hutch}}(B) = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{x}_j^T B\boldsymbol{x}_j = \frac{1}{N}\mathrm{tr}(\boldsymbol{X}^T B\boldsymbol{X}), \qquad \boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]. \tag{3.8}$$

An algorithm that improves the convergence properties of Hutchinson's estimator has been proposed in [47] with the name of *Hutch++*. It samples $\Omega \in \mathbb{R}^{n\times N_r}$ with

random i.i.d. $\mathcal{N}(0, 1)$ entries, then computes $B\Omega$ and an orthonormal basis $Q \in \mathbb{R}^{n\times N_r}$ of range$(B\Omega)$. Then the estimator is given by

$$\mathrm{tr}^{\mathrm{Hutch}++}_{N_r,N_H}(B) := \mathrm{tr}(Q^T BQ) + \mathrm{tr}^{\mathrm{Hutch}}_{N_H}((I - QQ^T)B(I - QQ^T)). \tag{3.9}$$

This estimator computes the trace of a rank N_r approximation of B and estimates the trace of the remaining part using the Hutchinson estimator with N_H samples, so its total cost is N_r matrix–vector products and $N_r + N_H$ quadratic forms with B. It has been proven in [47, Theorem 4.1] that the complexity of Hutch++ is optimal up to logarithmic factors amongst algorithms that access a positive semidefinite matrix B via matrix–vector products.

The implementation of Hutch++ proposed in [18] allows to prescribe a target accuracy $\epsilon > 0$ and a failure probability $\delta \in (0, 1)$ in input and then choose adaptively the parameters N_r and N_H in order to get the tail bound

$$\mathbb{P}\left[|\mathrm{tr}^{\mathrm{Hutch}++}_{N_r,N_H}(B) - \mathrm{tr}(B)| \geq \epsilon\right] \leq \delta \tag{3.10}$$

with as little computational effort as possible. This implementation goes under the name of *adaptive Hutch++* and will be our stochastic estimator of choice for the von Neumann entropy in view of its flexibility and optimal convergence properties. Note that most of our theory for Krylov methods in Sect. 4 can be used in combination with any other stochastic estimator based on matrix–vector products and quadratic forms with $B = f(A)$. For instance, we mention [52] where a low rank approximation of A constructed using powers of A is used to get a good approximation in case of fast decaying eigenvalues or large spectral gaps, and the paper [16] in which a Krylov subspace projection is integrated with the low rank approximation. Furthermore, our theory in Sect. 4 also applies to the techniques presented in the recent preprint [25] which improves on standard Hutch++ but does not allow an adaptive choice of the parameters as in adaptive Hutch++.

4 Computation of quadratic forms with Krylov methods

As shown in Sect. 3, the approximate computation of $\mathrm{tr}(f(A))$ with probing methods or stochastic trace estimators can be reduced to the computation of several quadratic forms with $f(A)$, i.e. expressions of the form $\boldsymbol{b}^T f(A)\boldsymbol{b}$. In this section, we briefly describe how they can be efficiently computed using polynomial and rational Krylov methods.

A *polynomial Krylov subspace* associated to A and $\boldsymbol{b}$ is given by

$$\mathcal{P}_m(A, \boldsymbol{b}) = \mathrm{span}\left\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\right\} = \{p(A)\boldsymbol{b} : p \in \Pi_{m-1}\}.$$

More generally, given a sequence of poles $\{\xi_j\}_{j\geq 1} \subset (\mathbb{C} \cup \{\infty\})\backslash\sigma(A) \cup \{0\}$, we can define a *rational Krylov subspace* as follows,

$$\mathcal{Q}_m(A, \boldsymbol{b}) = q_{m-1}(A)^{-1}\mathcal{P}_m(A, \boldsymbol{b}) = \left\{ r(A)\boldsymbol{b} : r(z) = \frac{p_{m-1}(z)}{q_{m-1}(z)}, \text{ with } p_{m-1} \in \Pi_{m-1} \right\}, \tag{4.1}$$

where $q_{m-1}(z) = \prod_{j=1}^{m-1}(1 - z/\xi_j)$. If all poles are equal to ∞, we have $q_{m-1}(z) \equiv 1$ and $\mathcal{Q}_m(A, \boldsymbol{b})$ coincides with the polynomial Krylov subspace $\mathcal{P}_m(A, \boldsymbol{b})$, so $\mathcal{P}_m(A, \boldsymbol{b})$ can be considered as a special case of $\mathcal{Q}_m(A, \boldsymbol{b})$. Note that this definition of q_{m-1} does not allow us to have poles $\xi_j = 0$; this can be fixed by changing the definition of q_{m-1} but it is not required in our case, since we are only going to use real negative poles and poles at ∞.

Let us denote by $V_m = [\boldsymbol{v}_1 \ldots \boldsymbol{v}_m]$ a matrix whose orthonormal columns span the Krylov subspace $\mathcal{Q}_m(A, \boldsymbol{b})$, and by $A_m = V_m^T A V_m$ the projection of A onto the subspace. We can then project the problem on $\mathcal{Q}_m(A, \boldsymbol{b})$ and approximate $\psi = \boldsymbol{b}^T f(A)\boldsymbol{b}$ in the following way,

$$\psi \approx \psi_m = \boldsymbol{b}^T V_m f(A_m) V_m^T \boldsymbol{b}.$$

If the basis V_m is constructed incrementally using the rational Arnoldi algorithm [51], we have $\boldsymbol{v}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|_2$ and therefore

$$\psi_m = \|\boldsymbol{b}\|_2^2 \boldsymbol{e}_1^T f(A_m)\boldsymbol{e}_1. \tag{4.2}$$

Note that the approximation ψ_m is closely related to the rational Krylov approximation of $f(A)\boldsymbol{b}$, which is given by

$$f(A)\boldsymbol{b} \approx V_m f(A_m) V_m^T \boldsymbol{b} = \|\boldsymbol{b}\|_2 V_m f(A_m)\boldsymbol{e}_1, \tag{4.3}$$

and is known to converge with a rate determined by the quality of rational approximations of f [35, Corollary 3.4]. We refer to [34, 35] for an extensive discussion on rational Krylov methods for the computation of matrix functions. The approximation (4.2) can be also interpreted in terms of rational Gauss quadrature rules, see for instance [2, 50].

Remark 4.1 The standard Arnoldi algorithm is inherently sequential since the computation of the new vector of the Krylov basis $\boldsymbol{v}_{m+1}$ requires the previous computation of $\boldsymbol{v}_m$. It is possible to parallelize it by solving several linear systems simultaneously and expanding the Krylov basis with blocks of vectors, with one of the strategies presented in [13], at the cost of lower numerical stability. Since in this work we are expected to compute several quadratic forms $\boldsymbol{b}^T f(A)\boldsymbol{b}$, we can easily achieve parallelization by assigning the quadratic forms to different processors and thus we can neglect parallelism inside the computation of a single quadratic form.

Remark 4.2 We mention that for symmetric A it is possible to construct the Krylov basis V_m using a method based on short recurrences such as rational Lanczos [48]. This has the advantage of reducing the orthogonalization costs, which can become

significant if m is large, and also avoids the need to store the matrix V_m when approximating the quadratic form $\boldsymbol{b}^T f(A)\boldsymbol{b}$, see (4.2). However, the implementation in finite arithmetic of short recurrence methods can suffer from loss of orthogonality, which in turn can lead to a slower convergence. In order to avoid this potential problem, we use the rational Arnoldi method with full orthogonalization. Since we expect to attain convergence in a small number of iterations, the orthogonalization costs remain modest compared to the cost of operations with A.

4.1 Convergence

By [35, Corollary 3.4], the accuracy of the approximation (4.3) for $f(A)\boldsymbol{b}$ is related to the quality of rational approximations to the function f of the form $r(z) = q_{m-1}(z)^{-1}p_{m-1}(z)$, where $p_{m-1} \in \Pi_{m-1}$ and q_{m-1} is determined by the poles of the rational Krylov subspace (4.1).

In the case of quadratic forms we can prove a faster convergence rate using the fact that $\psi_m = \psi$ for rational functions of degree up to $(2m-1, 2m-2)$.

Lemma 4.3 *Assume that A is symmetric. Let $p_{2m-1} \in \Pi_{2m-1}$ and define the rational function $r(z) = q_{m-1}(z)^{-2}p_{2m-1}(z)$. Then we have*

$$\boldsymbol{b}^T r(A)\boldsymbol{b} = \boldsymbol{b}^T V_m r(A_m) V_m^T \boldsymbol{b}.$$

Proof It is sufficient to prove this fact for $p_{2m-1}(z) = z^k$, for $k = 0, \ldots, 2m-1$. Assuming for the moment that $k = 2j+1$ is odd, we have

$$\boldsymbol{b}^T r(A)\boldsymbol{b} = \boldsymbol{b}^T s(A) A s(A)\boldsymbol{b}, \qquad \text{with} \quad s(z) = q_{m-1}(z)^{-1} z^j, \quad j < m.$$

Now using [35, Lemma 3.1], we obtain

$$\boldsymbol{b}^T r(A)\boldsymbol{b} = \left(\boldsymbol{b}^T V_m s(A_m) V_m^T\right) A \left(V_m s(A_m) V_m^T \boldsymbol{b}\right) = \boldsymbol{b}^T V_m^T r(A_m) V_m^T \boldsymbol{b}.$$

The case of $p_{m-1}(z) = z^k$ with k even can be proved in the same way, by writing

$$\boldsymbol{b}^T r(A)\boldsymbol{b} = \boldsymbol{b}^T s(A)^2 \boldsymbol{b}, \quad \text{with} \quad s(z) = q_{m-1}(z)^{-1} z^j, \quad j < m.$$

□

Lemma 4.3 leads to the following convergence result for the approximation of quadratic forms, with the same proof as [35, Corollary 3.4].

Proposition 4.4 *Let A be symmetric with spectrum contained in $[\lambda_{min}, \lambda_{max}]$, $\psi = \boldsymbol{b}^T f(A)\boldsymbol{b}$ and denote by ψ_m the approximation (4.2). We have*

$$|\psi - \psi_m| \le 2\|\boldsymbol{b}\|_2^2 \min_{p \in \Pi_{2m-1}} \|f - q_{m-1}^{-2} p\|_{[\lambda_{\min}, \lambda_{\max}]},$$

By comparing Proposition 4.4 with [35, Corollary 3.4], we can expect the convergence for quadratic forms to be roughly twice as fast as the one for matrix–vector products with $f(A)$.

4.2 Poles for the rational Krylov subspace

Recall that the function $f(z) = x \log x$ has the integral expression (2.10), which corresponds to a Cauchy-Stieltjes function multiplied by the polynomial $x(1-x)$. This implies that we can expect that a pole sequence that yields fast convergence for Cauchy-Stieltjes functions will be also effective in our case, especially if we add two poles at ∞ to account for the degree-two polynomial.

The authors of [45] consider the case of a positive definite matrix A with spectrum in $[a, b]$ and a Cauchy-Stieltjes function f, and relate the error for the computation of $f(A)\boldsymbol{b}$ with a rational Krylov method to the *third Zolotarev problem* in approximation theory. The solution to this problem is known explicitly and it can be used to find poles on $(-\infty, 0)$ that provide in some sense an optimal convergence rate for the rational Krylov method [45, Corollary 4]. However, the optimal Zolotarev poles are not nested, so they cannot be used to expand the Krylov subspace incrementally, and they are practical only if one knows in advance how many iterations to perform, for instance by relying on an a priori error bound. This drawback can be overcome by constructing a nested sequence of poles that is equidistributed according to the limit measure identified by the optimal poles, which can be done with the method of equidistributed sequences (EDS) described in [45, Section 3.5]. These poles have the same asymptotic convergence rate as the optimal Zolotarev poles and are usually better for practical purposes. To be computed, they require the knowledge of $[a, b]$ or a positive interval Σ such that $[a, b] \subset \Sigma$.

As an alternative, one can also use poles obtained from Leja-Bagby points [3, 35]. These points can be computed with an easily implemented greedy algorithm and they have an asymptotic convergence rate that is close to the optimal one. See [35, Section 4] and the references therein for additional information.

Remark 4.5 For the function $f(x) = x \log x$, the first few iterations of a polynomial Krylov method have a fast convergence rate that is close to the convergence rate of rational Krylov methods, even if it becomes asymptotically much slower for ill conditioned matrices. The faster initial convergence can be explained by the algebraic factor in the bound (2.13) for polynomial approximations of $x \log x$. Since polynomial Krylov iterations are cheaper than rational Krylov iterations, this suggests the use of a mixed polynomial-rational method, that starts with a few polynomial Krylov steps and then switches to a rational Krylov method with, e.g., EDS poles to achieve a higher accuracy. These methods are compared numerically in Example 4.10, where we also test the performance of the a posteriori error bound that we prove in Sect. 4.3.

Remark 4.6 Note that in the context of the graph entropy the matrix A is a graph Laplacian, which is a singular matrix. Therefore in principle it is not possible to use the poles described in this section, since here we assume that A is positive definite. However, we can use one of the desingularization strategies described in [11] to remove the 0 eigenvalue of the graph Laplacian, obtaining a matrix with spectrum contained in $[\lambda_2, \lambda_n]$, where λ_2 is the second smallest eigenvalue of A and λ_n is the largest one. In our implementation we use the approach that is called implicit desingularization in [11], which consists in replacing the initial vector $\boldsymbol{b}$ for the Krylov subspace with $\boldsymbol{c} = \boldsymbol{b} - \frac{\mathbb{1}^T \boldsymbol{b}}{n} \mathbb{1}$, where $\mathbb{1}$ is the vector of all ones. Since $\boldsymbol{c}$ is orthogonal to the eigenvector

$\mathbb{1}$ associated to the eigenvalue 0, it can be shown that the convergence of a Krylov subspace method with starting vector $\boldsymbol{c}$ is the same as for a matrix with spectrum in $[\lambda_2, \lambda_n]$. An approximation of $f(A)\boldsymbol{b}$ can be then cheaply recovered from $f(A)\boldsymbol{c}$ using the fact that $f(A)\mathbb{1} = f(0)\mathbb{1}$, and similarly for $\boldsymbol{b}^T f(A)\boldsymbol{b}$. See [11] for more details.

4.3 A posteriori error bound

In this section we prove an a posteriori bound for the error in the computation of the quadratic form $\boldsymbol{b}^T f(A)\boldsymbol{b}$ with a rational Krylov method. This bound is a variant of the one described in [34, Section 6.6.2] for $f(A)\boldsymbol{b}$, modified in order to account for the faster convergence rate in the case of quadratic forms.

We recall that after m iterations the rational Arnoldi algorithm yields the rational Arnoldi decomposition [12, Definition 2.3]

$$AV_{m+1}\underline{K_m} = V_{m+1}\underline{H_m}, \tag{4.4}$$

where span $V_{m+1} = \mathcal{Q}_{m+1}(A, \boldsymbol{b})$ and $\underline{K_m}$, $\underline{H_m}$ are $(m+1) \times m$ upper Hessenberg matrices with full rank. Let us consider the situation when $\xi_m = \infty$: in this case the last row of $\underline{K_m}$ is zero, and the decomposition simplifies to

$$AV_m K_m = V_m H_m + \boldsymbol{v}_{m+1}\boldsymbol{h}_{m+1}^T,$$

where H_m and K_m denote the $m \times m$ leading principal blocks of $\underline{H_m}$ and $\underline{K_m}$, respectively, and $\boldsymbol{h}_{m+1}^T = h_{m+1,m}\boldsymbol{e}_m^T$ denotes the last row of $\underline{H_m}$. Note that K_m is nonsingular since $\underline{K_m}$ has full rank, so we can rewrite the decomposition as

$$AV_m = V_m A_m + \boldsymbol{v}_{m+1}\boldsymbol{h}_{m+1}^T K_m^{-1}, \qquad \text{where} \quad A_m = V_m^T A V_m = H_m K_m^{-1}. \tag{4.5}$$

Remark 4.7 To derive the bound, we assume that $\xi_m = \infty$ because it simplifies the rational Arnoldi decomposition and hence the expression of the bound. Such an assumption is not restrictive, since the value of ξ_m does not have any impact on V_m and A_m, but only on $\boldsymbol{v}_{m+1}$ and the last column of $\underline{H_m}$ and $\underline{K_m}$. As we shall see later, we can use a technique described in [34, Section 6.1] to compute the bound for all m, without having to set the corresponding poles $\xi_m = \infty$. Note that if $\xi_m \neq \infty$, then (4.5) does not hold, and in particular $V_m^T A V_m \neq H_m K_m^{-1}$.

By using the Cauchy integral formula for f, we can obtain the following expression for the error [34, Section 6.2.2]:

$$f(A)\boldsymbol{b} - V_m f(A_m) V_m^T \boldsymbol{b} = \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A)^{-1}\boldsymbol{r}_m(z)\mathrm{d}z, \tag{4.6}$$

where Γ is a contour contained in the region of analyticity of f that encloses the spectrum of A, and

$$\boldsymbol{r}_m(z) = \boldsymbol{b} - (zI - A)\boldsymbol{x}_m(z), \qquad \text{with} \quad \boldsymbol{x}_m(z) = V_m(zI - A_m)^{-1}V_m^T\boldsymbol{b},$$

which can be seen as a residual vector of the shifted linear system $(zI - A)\boldsymbol{x} = \boldsymbol{b}$. It turns out that [34, Section 6.2.2]

$$\boldsymbol{r}_m(z) = \|\boldsymbol{b}\|_2\varphi_m(z)\boldsymbol{v}_{m+1}, \qquad \text{with} \quad \varphi_m(z) = \boldsymbol{h}_{m+1}^T K_m^{-1}(zI - A_m)^{-1}\boldsymbol{e}_1.$$

Observe that we have

$$\begin{aligned}\boldsymbol{b}^T(zI - A)^{-1}\boldsymbol{r}_m(z) &= \boldsymbol{r}_m(z)^T(zI - A)^{-1}\boldsymbol{r}_m(z) + \boldsymbol{x}_m(z)^T\boldsymbol{r}_m(z)\\ &= \boldsymbol{r}_m(z)^T(zI - A)^{-1}\boldsymbol{r}_m(z),\end{aligned}$$

where we exploited the fact that $\boldsymbol{x}_m(z) \in \mathcal{Q}_m(A, \boldsymbol{b}) \perp \boldsymbol{r}_m(z)$.

By using this property in conjunction with (4.6), we can write the error for the quadratic form $\boldsymbol{b}^T f(A)\boldsymbol{b}$ as

$$\begin{aligned}\psi - \psi_m &= \frac{1}{2\pi i}\int_\Gamma f(z)\boldsymbol{r}_m(z)^T(zI - A)^{-1}\boldsymbol{r}_m(z)\mathrm{d}z\\ &= \frac{1}{2\pi i}\|\boldsymbol{b}\|_2^2\int_\Gamma f(z)\varphi_m(z)^2\boldsymbol{v}_{m+1}^T(zI - A)^{-1}\boldsymbol{v}_{m+1}\mathrm{d}z. \qquad (4.7)\end{aligned}$$

We can now follow the same steps used in [34, Section 6.2.2] to bound the integral in (4.7). Assume that A_m has the spectral decomposition $A_m = U_m D_m U_m^T$, with U_m orthogonal and $D_m = \mathrm{diag}(\theta_1, \ldots, \theta_m)$, and define the vectors

$$[\alpha_1, \ldots, \alpha_m] = \boldsymbol{h}_{m+1}^T K_m^{-1}U_m \qquad \text{and} \qquad [\beta_1, \ldots, \beta_m]^T = U_m^T\boldsymbol{e}_1,$$

so that we have

$$\varphi_m(z) = \boldsymbol{h}_{m+1}^T K_m^{-1}U_m(zI - D_m)^{-1}U_m^T\boldsymbol{e}_1 = \sum_{j=1}^m \alpha_j\beta_j\frac{1}{z - \theta_j},$$

and

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

where we defined $\gamma_j = \sum_{\ell\,:\,\ell\neq j} \alpha_\ell\beta_\ell \dfrac{1}{\theta_j-\theta_\ell}$. By plugging this expression into (4.7) we get

$$\begin{aligned}
&\frac{1}{\|\boldsymbol{b}\|_2^2}(\psi-\psi_m)\\
&\quad=\frac{1}{2\pi i}\int_\Gamma f(z)\varphi_m(z)^2\boldsymbol{v}_{m+1}^T(zI-A)^{-1}\boldsymbol{v}_{m+1}\mathrm{d}z\\
&\quad=\sum_{j=1}^m\alpha_j^2\beta_j^2\frac{1}{2\pi i}\int_\Gamma\frac{f(z)}{(z-\theta_j)^2}\boldsymbol{v}_{m+1}^T(zI-A)^{-1}\boldsymbol{v}_{m+1}\mathrm{d}z\\
&\qquad+2\sum_{j=1}^m\alpha_j\beta_j\gamma_j\frac{1}{2\pi i}\int_\Gamma\frac{f(z)}{z-\theta_j}\boldsymbol{v}_{m+1}^T(zI-A)^{-1}\boldsymbol{v}_{m+1}\mathrm{d}z\\
&\quad=\sum_{j=1}^m\alpha_j^2\beta_j^2\boldsymbol{v}_{m+1}^T\Big((f(A)-f(\theta_j)I)(A-\theta_jI)^{-2}-f'(\theta_j)(A-\theta_jI)^{-1}\Big)\boldsymbol{v}_{m+1}\\
&\qquad+2\sum_{j=1}^m\alpha_j\beta_j\gamma_j\boldsymbol{v}_{m+1}^T(f(A)-f(\theta_j)I)(A-\theta_jI)^{-1}\boldsymbol{v}_{m+1},
\end{aligned}$$

where for the last equality we used the residue theorem [37, Theorem 4.7a].

Let us define

$$g_m(z)=\sum_{j=1}^m\begin{cases}\alpha_j^2\beta_j^2\left(\dfrac{f(z)-f(\theta_j)}{(z-\theta_j)^2}-\dfrac{f'(\theta_j)}{z-\theta_j}\right)+2\alpha_j\beta_j\gamma_j\dfrac{f(z)-f(\theta_j)}{z-\theta_j} & \text{if } z\neq\theta_j,\\ \dfrac{1}{2}\alpha_j^2\beta_j^2f''(\theta_j)+2\alpha_j\beta_j\gamma_jf'(\theta_j) & \text{if } z=\theta_j,\end{cases}\tag{4.8}$$

where the expression for $z=\theta_j$ is obtained by taking the limit for $z\to\theta_j$ in the definition for $z\neq\theta_j$. The above computations immediately lead to the following a posteriori bound for the quadratic form error.

Theorem 4.8 *Let A be a symmetric matrix with spectrum $\sigma(A)\subset[\lambda_{min},\lambda_{max}]$. Using the same notation as above, we have*

$$\|\boldsymbol{b}\|_2^2\min_{z\in[\lambda_{min},\lambda_{max}]}|g_m(z)|\le|\psi-\psi_m|\le\|\boldsymbol{b}\|_2^2\max_{z\in[\lambda_{min},\lambda_{max}]}|g_m(z)|.\tag{4.9}$$

Remark 4.9 We are mainly interested in the upper bound in (4.9) to have a reliable stopping criterion for the rational Krylov method, although the lower bound can also be of interest. We also mention that other bounds and error estimates can be obtained, such as the other ones described in [34, Section 6.6], but we found that the one derived in this section worked well enough for our purposes. Under certain assumptions, it is also possible to obtain upper and lower bounds for the quadratic form $\boldsymbol{b}^Tf(A)\boldsymbol{b}$ using pairs of rational Gauss quadrature rules, such as Gauss and Gauss-Radau quadrature rules. We refer to [2] for more details.

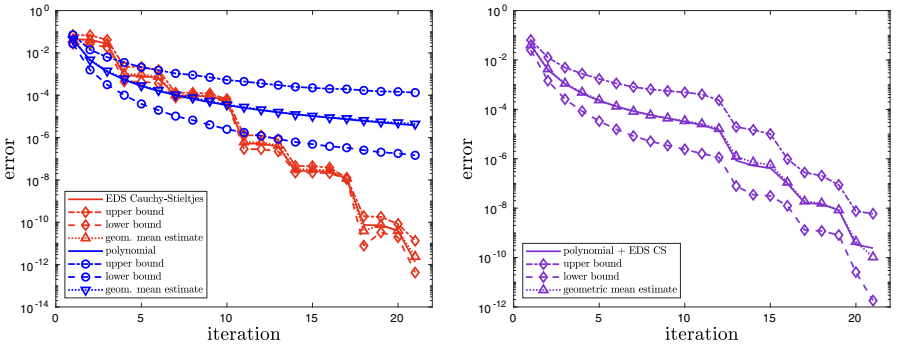

Fig. 3 Accuracy of error bounds and estimates for the relative error in the computation of $\boldsymbol{b}^T f(A)\boldsymbol{b}$ with Krylov methods, where $\boldsymbol{b}$ is a random vector, $f(x) = x\log(x)$ and A is a 2000×2000 matrix with eigenvalues that are Chebyshev points in the interval $[10^{-3}, 10^3]$. Left: polynomial Krylov and rational Krylov with EDS poles for Cauchy-Stieltjes function. Right: 10 poles at ∞ and 10 EDS poles for Cauchy-Stieltjes functions

Example 4.10 In this example we test the accuracy of the lower and upper bounds given in (4.9) for polynomial and rational Krylov methods. We consider the computation of $\boldsymbol{b}^T f(A)\boldsymbol{b}$, for a 2000×2000 matrix A with eigenvalues given by the Chebyshev points for the interval $\Sigma = [10^{-3}, 10^3]$, a random vector $\boldsymbol{b}$ and $f(x) = x\log(x)$. The upper and lower bounds are computed numerically by evaluating g_m on a discretization of the interval $[\lambda_{\min}, \lambda_{\max}]$. In addition to the lower and upper bounds, we also consider a heuristic estimate of the error given by the geometric mean of the upper and lower bound in (4.9), i.e.

$$\texttt{est}_m = \|\boldsymbol{b}\|_2^2 \sqrt{\min_{z\in\Sigma}|g_m(z)| \max_{z\in\Sigma}|g_m(z)|} \tag{4.10}$$

The results are shown in Fig. 3. On the left plot we show the convergence for the polynomial Krylov method and for a rational Krylov method with poles from an EDS for Cauchy-Stieltjes functions, and on the right plot a mixed polynomial-rational method that uses 10 poles at ∞ (which correspond to polynomial Krylov steps) followed by 10 EDS poles (see Remark 4.5). The upper and lower bounds match the shape of the convergence curve quite well, although they are less accurate when polynomial iterations are used. Rather surprisingly, the geometric mean of the bounds gives a very accurate estimate for the error, even in the case when the bounds themselves are less accurate.

Remark 4.11 We do not have a rigorous explanation for the accuracy of the estimate based on the geometric mean of the bounds in (4.9), but from further experiments it seems to be very accurate also for other functions. Unfortunately, if the spectral interval Σ is known only approximately, the bounds become less tight and the geometric mean estimate usually ends up underestimating the actual error by one or two orders of magnitude.

4.3.1 Computation of the bound

Recall that the a posteriori bounds in (4.9) hold after the m-th iteration only if $\xi_m = \infty$. In a practical scenario, i.e. when using the bounds as a stopping criterion for a rational Krylov method, it is desirable to evaluate the bounds after each iteration, without being forced to set the corresponding pole to ∞. As anticipated in Remark 4.7, we provide here two approaches to evaluate the bounds in (4.9) even when $\xi_m \neq \infty$.

One way to avoid setting poles to ∞, proposed in [34, Section 6.1], is to use an auxiliary basis vector $\boldsymbol{v}_\infty$, which is initialized as $\boldsymbol{v}_\infty^{(1)} = A\boldsymbol{v}_1$ at the beginning of the rational Arnoldi algorithm, and maintained orthonormal to the basis vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_j\}$ at each iteration j, at the cost of only one additional orthogonalization per iteration. The basis $[V_j \ \boldsymbol{v}_\infty^{(j)}]$ is an orthonormal basis of the rational Krylov subspace $\mathcal{Q}_{j+1}(A, \boldsymbol{b})$ with poles $\{\xi_1, \ldots, \xi_{j-1}, \infty\}$, and it is associated to the auxiliary Arnoldi decomposition

$$AV_j\widetilde{K}_j = [V_j \ \boldsymbol{v}_\infty^{(j)}]\underline{\widetilde{H}_j},$$

where $\widetilde{K}_j\boldsymbol{e}_j = \boldsymbol{e}_1$ and the last column of $\underline{\widetilde{H}_j}$ contains the orthogonalization coefficients for $\boldsymbol{v}_\infty^{(j)}$. This decomposition can be used to compute the bound (4.9) since the last row of $\underline{\widetilde{K}_j}$ is zero by construction.

Remark 4.12 We point out that if $\xi_j = \infty$ for some j, then the approach described above will not work from iteration $j+1$ onward, since $A\boldsymbol{v}_1 \in \mathcal{Q}_{j+1}(A, \boldsymbol{b})$ and therefore $\boldsymbol{v}_\infty^{(j+1)} = \boldsymbol{0}$ after orthogonalization. This is easily fixed by switching to a different auxiliary vector at iteration $j+1$, such as $\boldsymbol{v}_\infty = A\boldsymbol{v}_{j+1}$, or by setting directly at the start $\boldsymbol{v}_\infty = A^{\ell+1}\boldsymbol{v}_1$, where ℓ is the number of poles at ∞ used to construct the rational Krylov subspace.

In our experience, the technique described above can be sometimes subject to instability due to a large condition number of the matrix $\widetilde{K}_j$. We therefore also propose another approach, which is inspired by the methods for moving the poles of a rational Krylov subspace presented in [12, Section 4]. The idea is to add a pole at ∞ at the beginning of the pole sequence, and reorder the poles at each iteration in order to always have the last pole equal to ∞.

First of all, we recall how to swap poles in a rational Arnoldi decomposition. This procedure is a special case of the algorithm described in [12], but we still describe it in some detail for completeness. Recall that the poles of a rational Krylov subspace are the ratios of the entries below the main diagonals of H_j and K_j [12, Definition 2.3], i.e. $\xi_j = h_{j+1,j}/k_{j+1,j}$. In other words, the poles $\{\xi_1, \ldots, \xi_j\}$ are the eigenvalues of the upper triangular pencil $(\widehat{H}_j, \widehat{K}_j)$, where we denote by $\widehat{H}_j$ the bottom $j \times j$ block of H_j, and similarly for $\widehat{K}_j$. So we can obtain a transformation that swaps two adjacent poles in the same way as orthogonal transformations that reorder eigenvalues in a generalized Schur form [40]. Let U_j and W_j be $j \times j$ orthogonal matrices such that the pencil $U_j^T(\widehat{H}_j, \widehat{K}_j)W_j$ is still in upper triangular form and has the last two eigenvalues in reversed order; the matrices U_j and W_j only involve 2×2 rotations,

and they can be computed and applied cheaply as described in [40]. Defining $\widehat{U}_j = \text{blkdiag}(1, U_j) \in \mathbb{R}^{(j+1)\times(j+1)}$, we have the new rational Arnoldi decomposition

$$A\widetilde{V}_{j+1}\underline{\widetilde{K}_j} = \widetilde{V}_{j+1}\underline{\widetilde{H}_j},$$

where

$$\widetilde{V}_{j+1} = V_{j+1}\widehat{U}_j, \qquad \underline{\widetilde{K}_j} = \widehat{U}_j^T\underline{K_j}W_j \qquad \text{and} \qquad \underline{\widetilde{H}_j} = \widehat{U}_j^T\underline{H_j}W_j.$$

This decomposition has the same poles as $AV_{j+1}\underline{K_j} = V_{j+1}\underline{H_j}$, with the difference that the last two poles ξ_{j-1} and ξ_j are now swapped. In particular, if $\xi_{j-1} = \infty$, the last pole of the new decomposition is now ∞, and hence the last row of $\underline{\widetilde{K}_j}$ is equal to zero.

Given the pole sequence $\{\xi_1, \xi_2, \dots\}$, let us consider the rational Krylov subspace associated to the modified pole sequence $\{\infty, \xi_1, \xi_2, \dots\}$. Clearly, after the first iteration both pole sequences identify the same subspace $\mathcal{Q}_1(A, \boldsymbol{b})$, but the last (and first) pole of the modified sequence is ∞, so the last row of $\underline{K_1}$ is zero and we can use the decomposition $AV_1K_1 = V_2\underline{H_1}$ to compute the bound (4.9). After the second iteration, if $\xi_1 \neq \infty$, we can swap the poles ξ_1 and ∞ with the procedure outlined above to obtain the decomposition $AV_2K_2 = V_3\underline{H_2}$ associated to the poles $\{\xi_1, \infty\}$, where again the last row of $\underline{K_2}$ is equal to zero (for simplicity we still use the notation K_j instead of $\widetilde{K}_j$, and similarly for V_j and H_j). If $\xi_1 = \infty$, there is no need to swap poles and we can proceed to the next iteration.

By repeating the same steps at each iteration, we can ensure that after j iterations we have a decomposition $AV_jK_j = V_{j+1}\underline{H_j}$, associated to the poles $\{\xi_1, \dots, \xi_{j-1}, \infty\}$ in this order, so that the last row of $\underline{K_j}$ is equal to zero and it can be used to compute the bound (4.9).

Remark 4.13 Note that V_j is a basis of $\mathcal{Q}_j(A, \boldsymbol{b})$, which is the same subspace that we would have obtained if we had run the rational Arnoldi algorithm with poles $\{\xi_1, \dots, \xi_{j-1}\}$; so the method described in this section actually computes the approximation ψ_j and the bound associated to the poles $\{\xi_1, \dots, \xi_{j-1}\}$, and not to the modified pole sequence $\{\infty, \xi_1, \dots, \xi_{j-2}\}$. The initial pole at ∞ is only added to enable the computation of the bound and it is never used in the actual approximation.

5 Implementation aspects

In this section we outline the algorithm used to compute the entropy obtained by connecting the different components presented in the previous sections, and we briefly comment on some of the decisions that have to be taken in an implementation, especially concerning stopping criteria. Given a symmetric positive semidefinite matrix A with $\text{tr}(A) = 1$ and a target relative accuracy ϵ, the algorithm should output an estimate `trest` of $\text{tr}(f(A))$, where $f(x) = -x \log x$, such that

$$|\text{tr}(f(A)) - \texttt{trest}| \leq \epsilon\, \text{tr}(f(A)),$$

using either the probing approach of Sect. 3.1 or a stochastic trace estimator from Sect. 3.2. Observe that the entropy of an $n \times n$ density matrix is always bounded form above by $\log n$, but it may be in principle very small, so we prefer to aim for a certain relative accuracy rather than an absolute accuracy. Quadratic forms and matrix–vector products with $f(A)$ are computed using Krylov methods, specifically using a certain number of poles at ∞ followed by the EDS poles described in Sect. 4.2.

Remark 5.1 In the following, we are going to use $\hat{\epsilon}$ to denote an absolute error, to distinguish it from the target relative accuracy ϵ. Note that we can easily transform absolute inequalities for the error into relative inequalities if we know in advance an estimate or a lower bound for $\mathrm{tr}(f(A))$. Recall that if A is an M-matrix, $\mathcal{T}_d(f(A))$ is actually a lower bound for $\mathrm{tr}(f(A))$ (Proposition 3.6). In the general case, any rough approximation of the entropy can be used for this purpose, since the important point is determining the order of magnitude of $\mathrm{tr}(f(A))$.

Remark 5.2 The error in the approximation of $S(A)$ can be divided into the error in the approximation of the trace using a probing method or a stochastic estimator, and the error in the approximation of the quadratic forms with $f(A)$ using a Krylov subspace method. For simplicity, in the following we impose that the relative error associated to each of these two components is smaller than $\epsilon/2$.

5.1 Probing method implementation

We begin by observing that it is not possible to cheaply estimate the error of a probing method a posteriori, since error estimates are usually based on approximations with different values of the distance d, which in general lead to completely different colorings that would require computing all quadratic forms from scratch.

For this reason, it is best to find a value of d that ensures a relative accuracy ϵ a priori when using a distance-d coloring. This can be done using one of the bounds in Corollary 3.2, but it can often lead to unnecessary additional work, since the bounds usually overestimate the error by a couple of orders of magnitude; see Fig. 2. Therefore we also provide a heuristic criterion for choosing d that does not have the same theoretical guarantee as the bounds, but appears to work quite well in practice. In view of Corollary 3.2, we can expect the absolute error to behave as

$$|\mathrm{tr}(f(A)) - \mathcal{T}_d(f(A))| \sim \frac{C}{d^k} q^d, \tag{5.1}$$

for $k = 2$ and some parameters $C > 0$ and $q \in (0, 1)$. However, we found that sometimes the actual error behavior is better described with a different value of k, such as $k = 3$, so we do not impose that $k = 2$. To estimate the values of the parameters, we compute $\mathcal{T}_d(f(A))$ for $d = 1, 2, 3$ and use the estimate

$$|\mathrm{tr}(f(A) - \mathcal{T}_d(f(A))| \approx |\mathcal{T}_{d+1}(f(A)) - \mathcal{T}_d(f(A))|, \qquad d = 1, 2.$$

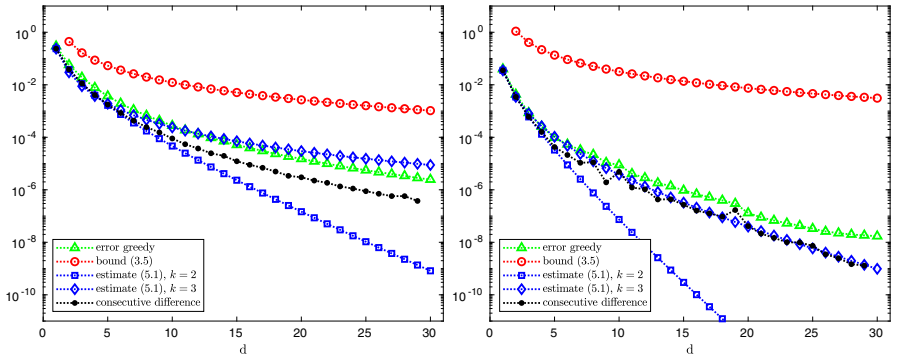

Fig. 4 Absolute error of the probing method with greedy coloring (Algorithm 1) compared with the theoretical bound (3.5) using $a = \lambda_2$, the heuristic error estimate (5.1) and the simple error estimate $|\mathrm{tr}(f(A) - \mathcal{T}_d(f(A))| \approx |\mathcal{T}_{d+1}(f(A)) - \mathcal{T}_d(f(A))|$. Left: Laplacian of the largest connected component of the graph `minnesota`, with 2640 nodes. Right: Laplacian of the largest connected component of the graph `eris1176`, with 1174 nodes

Assuming that (5.1) holds exactly and fixing the value of k, we can determine C and q by solving the equations

$$|\mathcal{T}_{d+1}(f(A)) - \mathcal{T}_d(f(A))| = \frac{C}{d^k} q^d, \qquad d = 1, 2.$$

We can check when the resulting estimate is below $\hat{\epsilon}$ to heuristically determine d, i.e., we select d as

$$d_\star = \min \left\{ d : \frac{C}{d^k} q^d \le \hat{\epsilon} \right\},$$

in order to have the approximate absolute error inequality

$$\left| \mathrm{tr}(f(A)) - \mathcal{T}_{d_\star}(f(A)) \right| \lesssim \hat{\epsilon}$$

We found that the best results are obtained for $k = 2$ and $k = 3$, so in our implementation we use the maximum of the two corresponding estimates. Variants of this estimate include using other values of d to estimate the parameters instead of $d = 1, 2, 3$, and using four different values in order to also estimate the parameter k. However, they usually give results that are similar or sometimes worse than the estimate presented above, so they are often not worth the additional effort required to compute them. In particular, using four values of d raises the risk of misjudging the value of q, causing the estimate to be inaccurate for large values of d. The error estimate (5.1) is compared to the actual error and the theoretical bound (3.5) for two different graphs in Fig. 4. The figure also includes a simple error estimate based on consecutive differences, which requires the computation of $\mathcal{T}_{d+1}(f(A))$ to estimate the error for $\mathcal{T}_d(f(A))$.

Remark 5.3 The heuristic criterion for selecting d requires the computation of $\mathcal{T}_d(f(A))$ for $d = 1, 2, 3$, so it is more expensive to use than the theoretical

bound (3.5). However, this cost is usually small compared to the cost of computing $\mathcal{T}_d(f(A))$ for the selected value of d, especially if the requested accuracy is small. Note also that the heuristic criterion always computes $\mathcal{T}_3(f(A))$, so it does more work than necessary when $d \leq 2$ would be sufficient. Nevertheless, in such a situation the theoretical bound (3.5) may suggest to use an even higher value of d (see Fig. 4).

After choosing d such that

$$|\mathrm{tr}(f(A)) - \mathcal{T}_d(f(A))| \leq \hat{\epsilon}, \tag{5.2}$$

using either the a priori bound (3.5) or the estimate (5.1), a distance-d coloring can be computed with one of the coloring algorithms described in Sect. 3.1, depending on the properties of the graph. The greedy coloring [53, Algorithm 4.2] is usually a good choice for general graphs.

Let us now determine the accuracy required in the computation of the quadratic forms. Assume that we have

$$\mathcal{T}_d(f(A)) = \sum_{\ell=1}^{s} \boldsymbol{v}_\ell^T f(A) \boldsymbol{v}_\ell,$$

where $\{\boldsymbol{v}_\ell\}_{\ell=1}^s$ are the probing vectors used in the distance-d coloring. Let $\hat{\psi}_\ell$ denote the approximation of $\boldsymbol{v}_\ell^T f(A)\boldsymbol{v}_\ell$ obtained with a Krylov method. Recall that $\|\boldsymbol{v}_\ell\|_2 = |V_\ell|^{1/2}$, where V_ℓ denotes the set of the partition associated to the ℓ-th color. If we impose the conditions

$$\left| \boldsymbol{v}_\ell^T f(A)\boldsymbol{v}_\ell - \hat{\psi}_\ell \right| \leq \hat{\epsilon} \cdot \frac{|V_\ell|}{n} \qquad \ell = 1, \ldots, s, \tag{5.3}$$

where we normalized the accuracy requested for each quadratic form depending on $\|\boldsymbol{v}_\ell\|_2$, we obtain the desired absolute accuracy on the probing approximation

$$\left| \mathcal{T}_d(f(A)) - \sum_{\ell=1}^{s} \hat{\psi}_\ell \right| \leq \hat{\epsilon}. \tag{5.4}$$

If we are aiming for a relative accuracy ϵ, we should select $\hat{\epsilon} = \frac{1}{2}\epsilon\,\mathrm{tr}(f(A))$ in (5.2) and (5.4). In practice, $\mathrm{tr}(f(A))$ will be replaced by a rough approximation (see Remark 5.1).

The overall probing algorithm is summarized in Algorithm 2.

5.2 Adaptive Hutch++ implementation

We use the Matlab code of [49, Algorithm 3] provided by the authors, modified to use Krylov methods for the computations with $f(A)$. This algorithm requires an absolute

Algorithm 2 Probing method for $S(A)$

Input: $A \in \mathbb{R}^{n\times n}$ density matrix, ϵ relative error tolerance
Output: trest $\approx S(A)$ such that $|\text{tr} - S(A)|/S(A) \lesssim \epsilon$

1: Select d such that $|\mathcal{T}_d(f(A)) - S(A)|/S(A) \lesssim \epsilon/2$, using either the bound (3.5) or the heuristic (5.1). The heuristic (5.1) requires the computation of $\mathcal{T}_d(f(A))$ for $d = 1, 2, 3$, which can be done by running steps 2 – 4.
2: Compute a distance-d coloring of $G(A)$ with, e.g., Algorithm 1 and the associated probing vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s\}$.
3: For $\ell = 1, \ldots, s$, compute $\hat{\psi}_\ell \approx \boldsymbol{v}_\ell^T f(A)\boldsymbol{v}_\ell$ such that (5.3) holds, using a rational Krylov method with either the upper bound (4.9) or the estimate (4.10) as stopping criterion.
4: **return** trest $= \sum_{\ell=1}^{s} \hat{\psi}_\ell$, satisfying $\left|\sum_{\ell=1}^{s} \hat{\psi}_\ell - S(A)\right|/S(A) \lesssim \epsilon$.

tolerance $\hat{\epsilon}$ and a failure probability δ, and outputs an approximation $\text{tr}_{\text{adap}}(f(A))$ such that

$$\mathbb{P}\big[|\text{tr}(f(A)) - \text{tr}_{\text{adap}}(f(A))| \geq \hat{\epsilon}\big] \leq \delta.$$

To obtain an approximation within a relative accuracy ϵ, we can use $\hat{\epsilon} \approx \epsilon\,\text{tr}(f(A))$, using a rough approximation of $\text{tr}(f(A))$. Similarly to the probing method, in order to have a final relative error bounded by ϵ, in our implementation we use a tolerance $\hat{\epsilon} \approx \frac{1}{2}\epsilon\,\text{tr}(f(A))$ for adaptive Hutch++, and we set the accuracy for the computation of matrix–vector products and quadratic forms in order to ensure that the total error due to the Krylov approximations remains below $\frac{1}{2}\epsilon\,\text{tr}(f(A))$. We omit the technical details to simplify the presentation.

5.3 Krylov method implementation

Quadratic forms with $f(A)$ are approximated using a Krylov method with some poles at ∞ followed by the EDS poles of Sect. 4.2, using as a stopping criterion either the a posteriori upper bound (4.9) or the estimate shown in Example 4.10.

The number of poles at ∞ is chosen in an adaptive way, switching to finite poles when the error reduction in the last few iterations of the polynomial Krylov method is "small". Specifically, we decide to switch to EDS poles after the k-th iteration if on average the last $\ell \geq 1$ iterations did not reduce the error bound or estimate `err_est` by at least a factor $c \in (0, 1)$, i.e. if

$$\frac{\texttt{err\_est}_k}{\texttt{err\_est}_{k-\ell}} \geq c^\ell.$$

In our implementation we use $\ell = 3$ and $c = 0.75$, usually leading to at most 10 polynomial Krylov iterations.

Since EDS poles are contained in $(-\infty, 0)$, each rational Krylov iteration involves the solution of a symmetric positive definite linear system, which can be computed either with a direct method using a sparse Cholesky factorization, or iteratively with the conjugate gradient method using a suitable preconditioner. Note that the same EDS

poles can be used for all quadratic forms, so the number of different matrices that appear in the linear systems is usually small and independent of the total number of quadratic forms. Although this depends on the accuracy requested for the entropy, the number of EDS poles used is almost always bounded by 10, and often much smaller than that: see the numerical experiments in Sect. 6 for some examples. This is a great advantage for direct methods, especially when the Cholesky factor remains sparse, since we can compute and store a Cholesky factorization for each pole and then reuse it for all quadratic forms. If the fill-in in the Cholesky factor is moderate, the cost of a rational iteration can become comparable to the cost of a polynomial one, leading to large savings when computing many quadratic forms. Of course, for large matrices with a general sparsity structure the computation of even a single Cholesky factor may be unfeasible, so the only option is to use a preconditioned iterative method. In such a situation, it is still possible to benefit from the small number of different matrices that appear in linear systems by storing and reusing preconditioners, but the gain is less evident compared to direct methods.

The matrix–vector products with $f(A)$ in the Hutch++ algorithm are approximated with the same Krylov subspace method, with the difference that we use the a posteriori upper and lower bounds from [34, eq. (6.15)]. A geometric mean estimate similar to the one used in Example 4.10 can be also used in this context. For the computation of the a posteriori bounds we use the pole swapping technique with an auxiliary pole at ∞ described in Sect. 4.3.1.

6 Numerical experiments

The experiments were done in Matlab R2021b on a laptop with operating system Ubuntu 20.04, using a single core of an Intel i5-10300H CPU running at 2.5 GHz, with 32 GB of RAM. Since we are using Matlab, the execution times may not reflect the performance of a high performance implementation, but they are still a useful indicator when comparing different methods.

6.1 Test matrices

We consider a number of symmetric test matrices from the SuiteSparse Matrix Collection [21]. All matrices are treated as binary matrices, i.e. all edge weights are set to one. For each matrix, we extract the graph Laplacian associated to the largest connected component and we normalize it so that it has unit trace. We report some information on the resulting matrices in Table 1. For the four smallest matrices, the eigenvalues were computed via diagonalization, while for the larger matrices the eigenvalues λ_2 and λ_n were approximated using `eigs`. The cost of solving a linear system with a direct method is highly dependent on the fill-in in the Cholesky factorization; the column labelled `fill-in` in Table 1 contains the ratios $\texttt{nnz}(R)/\texttt{nnz}(\rho)$, where ρ is the test matrix and R is the Cholesky factor of any shifted matrix $\rho + \alpha I$, for $\alpha > 0$ ($\texttt{nnz}(M)$ denotes the number of nonzeros of a matrix M). All matrices have been

Table 1 Information on the matrices used in the experiments

Test matrix	n	nnz(ρ)	Fill-in	λ_2	λ_n	Entropy
Yeast	2224	15442	3.6	4.54e−06	4.96e−03	7.055
Minnesota	2640	9244	1.3	1.28e−07	1.04e−03	7.607
ca-HepTh	8638	58250	7.5	4.92e−07	1.33e−03	8.540
bcsstk29	13830	618678	2.9	7.22e−08	1.25e−04	9.440
Cond-mat-2005	36458	379926	21.7	5.63e−08	8.13e−04	9.958
Loc-Brightkite	56739	482629	32.6	7.10e−08	2.67e−03	9.896
ut2010	115406	687472	1.2	2.72e−10	3.44e−04	11.361
usroads	126146	450046	1.4	2.39e−11	2.54e−05	11.478
Com-Amazon	334863	2186607	105.9	6.69e−10	2.97e−04	12.400
ny2010	350167	2059711	1.8	4.67e−12	3.63e−05	12.541
RoadNet-PA	1087562	4170590	1.6	5.54e−13	3.37e−06	13.628

ordered using the approximate minimum degree reordering option available in Matlab before factorization.

6.2 Probing bound vs. estimate

In this experiment, we fix a relative error tolerance $\epsilon = 10^{-3}$ and we compare the choice of d given by the theoretical bound (3.5) with the one provided by the heuristic estimate (5.1). We report in Table 2 the error, the execution time, the value of d and the number of colors used in the two cases. When the theoretical bound is used, the selected value of d is significantly higher compared to the one chosen by the heuristic estimate, but in both cases the overall error remains below the tolerance ϵ. Moreover, for certain graphs using the theoretical bound leads to greedy colorings with a number of colors equal to the number of nodes in the graph, completely negating the advantage of using a probing method. Observe that the errors obtained with the larger value of d are not much smaller than the ones for the smaller value of d, because in both cases the quadratic forms are computed with target relative accuracy ϵ, so the probing error for the larger value of d is dominated by the error in the quadratic forms. In the case of the heuristic estimate, the execution time includes the time required to run the probing method for $d = 1, 2, 3$ in order to evaluate (5.1). The stopping criterion for the Krylov subspace method uses the upper bound (4.9). The execution time can be further reduced by using the estimate (4.10) for the Krylov subspace method, as the following experiment shows. The diagonalization time for the matrices used in this experiment can be found in the last column of Table 4. Note that for smaller matrices, diagonalization is often the fastest method, but the advantage of approximating the entropy with a probing method is already evident for matrices of size $n \approx 10000$.

Table 2 Comparison of the theoretical bound (3.5) against the heuristic estimate (5.1) for choosing d, using relative tolerance $\epsilon = 10^{-3}$ in the probing method. Top row: heuristic estimate. Bottom row: theoretical bound

Test matrix	n	Error	d	Colors	Time (s)
`Yeast`	2224	3.062e−04	3	222	0.952
		3.733e−05	25	2224	4.464
`Minnesota`	2640	4.456e−04	5	24	0.196
		3.173e−05	18	255	0.779
`ca-HepTh`	8638	2.974e−04	3	252	2.273
		3.161e−05	27	8638	42.078
`bcsstk29`	13830	4.912e−05	3	176	2.292
		8.497e−05	12	2095	17.849

Table 3 Comparison of the upper bound (4.9) against the geometric mean estimate (4.10) for Krylov methods used in the probing method, using relative tolerance $\epsilon = 10^{-5}$ for the probing method. Top row: geometric mean estimate. Bottom row: upper bound

Test matrix	n	Error	Poly iter	Rat iter	Time (s)
`Yeast`	2224	4.405e−06	16140	748	7.095
		6.023e−06	16419	2237	8.231
`Minnesota`	2640	5.728e−07	2983	289	1.535
		2.490e−06	2987	598	1.734
`ca-HepTh`	8638	8.195e−07	39481	2442	40.240
		2.395e−06	39747	6389	50.133
`bcsstk29`	13830	6.133e−06	4704	0	12.093
		7.545e−06	6363	44	16.047

6.3 Krylov bound vs. estimate

We fix an error tolerance $\epsilon = 10^{-5}$ and compare the performance of the geometric mean error estimate (4.10) with the theoretical upper bound (4.9) for the Krylov subspace method. The value of d for the probing method is selected using the heuristic estimate (5.1). The entropy error, execution time, and total number of polynomial and rational Krylov iterations are reported in Table 3. We can see that using the estimate instead of the theoretical bound moderately reduces the computational effort, while still attaining the requested accuracy ϵ on the entropy. In particular, observe that the number of rational Krylov iterations is significantly higher when using the upper bound (4.9). In this experiment, all linear systems are solved with direct methods and Cholesky factorizations are stored and reused.

Table 4 Results for Hutch++ applied to some test matrices. For each matrix, the first and second row show the results for $\epsilon = 10^{-2}$ and $\epsilon = 10^{-3}$, respectively. The failure probability is $\delta = 10^{-2}$ in both cases. The last column contains the diagonalization times

Test matrix	Avg error	Worst error	N_r	$N_r + N_H$	Time (s)	`eig` (s)
`Yeast`	2.55e−03	9.94e−03	3	282	0.421	0.508
	3.56e−04	1.16e−03	1228	2160	19.735	
`Minnesota`	3.46e−03	1.07e−02	3	154	0.111	0.819
	4.53e−04	1.26e−03	1854	2684	35.721	
`ca-HepTh`	2.83e−03	9.92e−03	3	81	0.205	22.170
	3.55e−04	9.14e−04	635	3968	66.350	
`bcsstk29`	1.84e−03	6.97e−03	3	38	0.171	86.696
	2.39e−04	8.24e−04	3	1883	13.276	

6.4 Adaptive Hutch++

Here we test the performance and the accuracy of the adaptive implementation of Hutch++ [49, Algorithm 3]. A relative accuracy ϵ is achieved by setting the absolute tolerance to $\epsilon S(\rho)$, where $S(\rho)$ is computed via diagonalization and considered as exact. The computational effort of Hutch++ is determined by the parameters N_r and N_H described in Sect. 3.2. In particular, the number of matrix–vector products is equal to N_r and the number of quadratic forms is equal to $N_r + N_H$. We used $\epsilon = 10^{-2}, 10^{-3}$ as target tolerances and $\delta = 10^{-2}$ as failure probability. Matrix–vector products and quadratic forms are computed using the Krylov subspace method with the geometric mean estimate as a stopping criterion (4.10). In Table 4 we compare the results for the two tolerances, obtained as an average of 100 runs of the algorithm, including both the average and worst relative error. In the majority of cases, the worst error is below the input tolerance ϵ. We see that for $\epsilon = 10^{-2}$ the computation with Hutch++ is very fast for all test matrices; on the other hand, the cost becomes significantly higher for $\epsilon = 10^{-3}$, showing that the stochastic estimator quickly becomes inefficient as the required accuracy increases. Observe that for $\epsilon = 10^{-2}$ adaptive Hutch++ uses only 3 matvecs for all tests problems, which is the minimum amount that can be used by the implementation in [49]. This means that the internal criteria of the algorithm have determined that spending more matvecs in the low rank approximation is not beneficial, and hence the convergence of the method is roughly the same as for Hutchinson's estimator. A similar behavior can be observed in Table 7, and can be linked to the fact that for these test matrices ρ, the matrix function $-\rho \log \rho$ does not exhibit eigenvalue decay and hence cannot be well-approximated by low rank matrices. On the other hand, for problems where low rank approximation is more effective, stochastic estimators that exploit it such as Hutch++ can have much faster convergence.

Table 5 Results for the probing method applied to test matrices with large-world sparsity structure, using relative tolerance $\epsilon = 10^{-4}$

Test matrix	n	d	Colors	Poly iter	Rat iter	Time (s)
`ut2010`	115406	4	504	7070	919	79.60
`usroads`	126146	8	77	626	0	6.30
`ny2010`	350167	5	329	3914	15	111.39
`RoadNet-PA`	1087562	8	106	827	0	84.46

6.5 Larger matrices

In this section we test the probing method and the adaptive Hutch++ algorithm on larger matrices, for which it would be extremely expensive to compute the exact entropy. In light of the results shown in Tables 2 and 3, we select the value of d for the probing method using the heuristic estimate (5.1) and we use the geometric mean estimate (4.10) for the Krylov subspace method. The results are reported in Tables 5 and 6 for the probing method, and in Table 7 for Hutch++. Figure 5 contains a more detailed breakdown of the execution time for the probing method used on the matrices of Table 5. We separate the time in preprocessing, where we evaluate the heuristic (5.1) to select d, and the main run of the algorithm with the chosen value of d. The time for the main run is further divided in coloring, and polynomial and rational Krylov iterations. The time for the Cholesky factorizations refers to the whole process, since the factors are computed and stored when a certain pole for a rational Krylov iteration is encountered for the first time.

In Table 5, we consider matrices with a "large-world" sparsity structure, such as road networks, and we use a relative tolerance of $\epsilon = 10^{-4}$. For these matrices, for which the diameter and the average path length are relatively large, it is possible to compute distance-d colorings with a relatively small number of colors, and therefore probing methods converge quickly. Moreover, Cholesky factorizations can be computed cheaply and have a small fill-in, so it is possible to rapidly solve linear systems using a direct method. On the other hand, in Table 6 we consider matrices with a "small-world" sparsity structure, more typical of social networks and scientific collaboration networks. These matrices require a much larger number of colors to construct distance-d colorings, even for small values of d. The cost of probing methods is thus significantly higher on this kind of problem. In Table 6, only polynomial Krylov iterations are used due to the low relative tolerance $\epsilon = 10^{-2}$, so it is never necessary to solve linear systems. Recall that these matrices also have a high fill-in in the Cholesky factorizations (see Table 1), so the conjugate gradient method with a suitable preconditioner is likely to be much more efficient than a direct method for solving a linear system.

In Table 7 we show the results for the adaptive Hutch++ algorithm, using relative tolerance $\epsilon = 10^{-2}$. The results are obtained as an average of 100 runs of the algorithm. We can observe that the stochastic trace estimator works well for both large-world and small-world graphs, in contrast to the probing method.

Table 6 Results for the probing method applied to test matrices with small-world sparsity structure, using relative tolerance $\epsilon = 10^{-2}$

Test matrix	n	d	Colors	Poly iter	Time (s)
cond-mat-2005	36458	3	1221	3883	13.809
loc-Brightkite	56739	3	3946	18765	90.564
Com-Amazon	334863	3	625	1285	47.458

Table 7 Results for Hutch++ applied to large test matrices, with relative tolerance $\epsilon = 10^{-2}$ and failure probability $\delta = 10^{-2}$. The parameters N_r and N_H are defined in Sect. 3.2

Test matrix	n	N_r	$N_r + N_H$	Time (s)
ny2010	350167	3	10	0.490
usroads	126146	3	14	0.190
ny2010	350167	3	10	0.487
roadNet-PA	1087562	3	8	1.126
cond-mat-2005	36458	3	34	0.448
loc-Brightkite	56739	3	42	4.576
Com-Amazon	334863	3	11	1.171

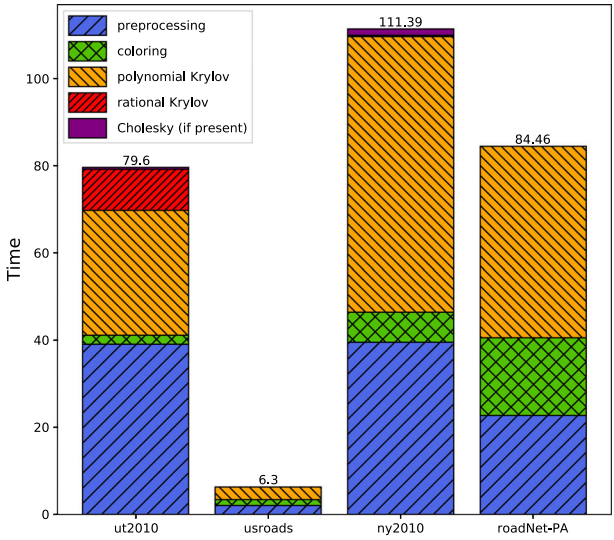

Fig. 5 Breakdown of the execution time of the probing method for the test matrices in Table 5

6.6 Algorithm scaling

To investigate how the complexity of the algorithms scales with the matrix size, we compare the scaling of the probing method and the stochastic trace estimator on two different test problems with increasing dimension. The first one is the graph Laplacian

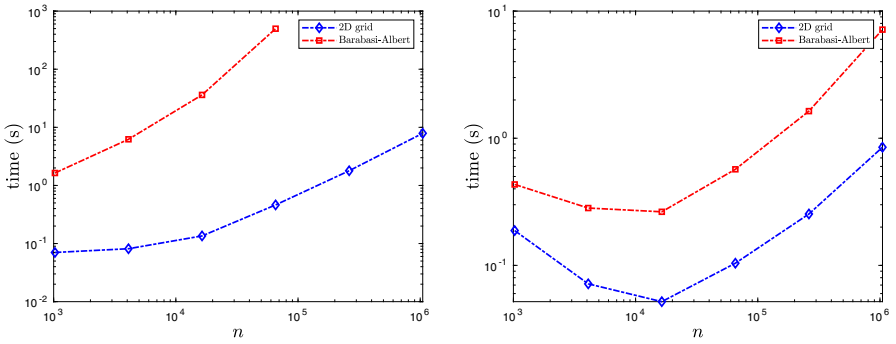

Fig. 6 Execution times for the probing method (left, $\epsilon = 10^{-4}$) and the adaptive Hutch++ algorithm (right, $\epsilon = 10^{-2}$) on the graph Laplacian of a 2D regular grid and a Barabasi-Albert random graph, as a function of the number of nodes n

of a 2D regular square grid, and the second one is the graph Laplacian of a Barabasi-Albert random graph, generated using the `pref` function of the CONTEST Matlab package [55]. For the probing method on the 2D grid, we use the optimal distance-d coloring with $\lceil \frac{1}{2}(d+1)^2 \rceil$ colors described in [26]. For both test problems, we use a relative tolerance $\epsilon = 10^{-4}$ for the probing method, and a relative tolerance $\epsilon = 10^{-2}$ and failure probability $\delta = 10^{-2}$ for adaptive Hutch++, averaging over 100 runs. The results are summarized in Fig. 6, for graphs with a number of nodes from $n = 2^{10}$ to $n = 2^{20}$. As expected, the probing method is much more efficient in the case of the 2D grid, since the number of colors used in the distance-d colorings remains constant as n increases. On the other hand, for the Barabasi-Albert random graph, which has a small-world structure, the number of colors used in a distance-d coloring increases with the number of nodes, and hence the scaling for the probing method is significantly worse. The adaptive Hutch++ algorithm also has a better performance for the 2D grid, but the scaling in the problem size is good for both graph categories, since the number of vectors used in the trace approximation does not increase with the matrix dimension. However, the stochastic approach is only viable with a loose tolerance ϵ for this kind of problem since low rank approximation is not effective, as discussed in Sect. 6.4. The initial decrease in the execution time for Hutch++ as n increases is caused by the fact that the adaptive algorithm uses a larger number of vectors for the graphs with fewer nodes.

7 Conclusions

In this paper we have investigated two approaches for approximating the von Neumann entropy of a large, sparse, symmetric positive semidefinite matrix. The first method is a state-of-the-art randomized approach, while the second one is based on the idea of probing. Both methods require the computation of many quadratic forms involving the matrix function $f(A)$ with $f(x) = -x \log x$, an expensive task given the lack of smoothness of $f(x)$ at $x = 0$. We have examined the use of both polynomial and rational Krylov subspace methods, and combinations of the two. Pole selection and

several implementation aspects, such as heuristics and stopping criteria, have been investigated. Numerical experiments in which the entropy is computed for a variety of networks have been used to test the various approximation methods. Not surprisingly, the performance of the methods is affected by the structure of the underlying network, especially for the method based on the probing idea. Our main conclusion is that the probing approach is better suited than the randomized one for graphs with a large-world structure, since they admit distance-d colorings with a relatively small number of colors. Conversely, for complex networks with a small-world structure, the number of colors required for distance-d colorings is larger, so the probing approach becomes more expensive. For this type of graphs, the randomized method is more competitive than the one based on probing, since it is less affected by the structure of the graph; however, for matrices in which low rank approximation cannot be exploited such as the graph Laplacians that we consider, randomized trace estimators are best suited for computing approximations with a relatively low accuracy, since their cost quickly grows as the requested accuracy is increased.

Acknowledgements We would like to thank the editor Ilse Ipsen and two anonymous reviewers for their insightful comments.

Funding Open access funding provided by Scuola Normale Superiore within the CRUI-CARE Agreement. We acknowledge financial support by MUR (Italian Ministry for University and Research) through the PNRR MUR project PE0000023-NQSTI and by INDAM (Italian Institute of High Mathematics) through the INDAM-GNCS project "Metodi basati su matrici e tensori strutturati per problemi di algebra lineare di grandi dimensioni". Funding for the second and third author's PhD scholarship is provided by the "Departments of Excellence" program of the Italian Ministry for University and Research.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Aarons, J., Skylaris, C.K.: Electronic annealing Fermi operator expansion for DFT calculations on metallic systems. J. Chem. Phys **148**(7), 074107 (2018)
2. Alahmadi, J., Pranić, M., Reichel, L.: Rational Gauss quadrature rules for the approximation of matrix functionals involving Stieltjes functions. Numer. Math. **151**(2), 443–473 (2022)
3. Bagby, T.: On interpolation by rational functions. Duke Math. J. **36**, 95–104 (1969)
4. Beckermann, B., Reichel, L.: Error estimates and evaluation of matrix functions via the Faber transform. SIAM J. Numer. Anal. **47**(5), 3849–3883 (2009)
5. Bengtsson, I., Zyczkowski, K.: Geometry of Quantum States: An Introduction to Quantum Entanglement. Cambridge University Press, Cambridge (2006)

6. Benzi, M.: Localization in matrix computations: theory and applications. In: Exploiting Hidden Sstructure in Matrix Computations: Algorithms and Applications, volume 2173 of Lecture Notes in Math., 2173, Fond. CIME/CIME Found. Subser., pages 211–317. Springer, Cham (2016)
7. Benzi, M., Boito, P., Razouk, N.: Decay properties of spectral projectors with applications to electronic structure. SIAM Rev. **55**(1), 3–64 (2013)
8. Benzi, M., Golub, G.H.: Bounds for the entries of matrix functions with applications to preconditioning. BIT **39**(3), 417–438 (1999)
9. 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)
10. Benzi, M., Simoncini, V.: Decay bounds for functions of Hermitian matrices with banded or Kronecker structure. SIAM J. Matrix Anal. Appl. **36**(3), 1263–1282 (2015)
11. Benzi, M., Simunec, I.: Rational Krylov methods for fractional diffusion problems on graphs. BIT **62**(2), 357–385 (2022)
12. Berljafa, M., Güttel, S.: Generalized rational Krylov decompositions with an application to rational approximation. SIAM J. Matrix Anal. Appl. **36**(2), 894–916 (2015)
13. Berljafa, M., Güttel, S.: Parallelization of the rational Arnoldi algorithm. SIAM J. Sci. Comput. **39**(5), S197–S221 (2017)
14. Berman, A., Plemmons. R.J.: Nonnegative Matrices in the Mathematical Sciences, volume 9 of Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994. Revised reprint of the 1979 original
15. Braunstein, S.L., Ghosh, S., Severini, S.: The Laplacian of a graph as a density matrix: a basic combinatorial approach to separability of mixed states. Ann. Comb. **10**(3), 291–317 (2006)
16. Chen, T., Hallman, E.: Krylov-aware stochastic trace estimation. SIAM J. Matrix Anal. Appl. **44**(3), 1218–1244 (2023)
17. Choi, H., He, J., Hu, H., Shi, Y.: Fast computation of von Neumann entropy for large-scale graphs via quadratic approximations. Linear Algebra Appl. **585**, 127–146 (2020)
18. Cortinovis, A., Kressner, D.: On randomized trace estimates for indefinite matrices with an application to determinants. Found. Comput. Math. **22**(3), 875–903 (2022)
19. Cramer, M., Eisert, J.: Correlations, spectral gap and entanglement in harmonic quantum systems on generic lattices. New J. Phys. **8**(5), 71–71 (2006)
20. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: ACM '69: Proceedings of the 1969 24th National Conference, pp 157–172, New York, NY, USA, (1969)
21. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), Art. 1, 25 (2011)
22. De Domenico, M., Biamonte, J.: Spectral entropies as information-theoretic tools for complex network comparison. Phys. Rev. X **6**, 041062 (2016)
23. De Domenico, M., Nicosia, V., Arenas, A., Latora, V.: Structural reducibility of multilayer networks. Nat. Commun. **6**(1), 6864 (2015)
24. Demko, S., Moss, W.F., Smith, P.W.: Decay rates for inverses of band matrices. Math. Comp. **43**(168), 491–499 (1984)
25. Epperly, E.N., Tropp, J.A., Webber, R.J.: Xtrace: Making the most of every sample in stochastic trace estimation, arXiv:2301.07825 [math.NA], (2023)
26. Fertin, G., Godard, E., Raspaud, A.: Acyclic and k-distance coloring of the grid. Inform. Process. Lett. **87**(1), 51–58 (2003)
27. Frommer, A., Schimmel, C., Schweitzer, M.: Bounds for the decay of the entries in inverses and Cauchy-Stieltjes functions of certain sparse, normal matrices. Numer. Linear Algebra Appl. **25**(4), e2131, 17, (2018)
28. Frommer, A., Schimmel, C., Schweitzer, M.: Non-Toeplitz decay bounds for inverses of Hermitian positive definite tridiagonal matrices. Electron. Trans. Numer. Anal. **48**, 362–372 (2018)
29. 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)
30. Frommer, A., Simoncini, V.: Matrix functions. In: Model Order Reduction: Theory, Research Aspects and Applications, volume 13 of Math. Ind., pp. 275–303. Springer, Berlin, (2008)
31. Fuentes, R.D., Donatelli, M., Fenu, C., Mantica, G.: Estimating the trace of matrix functions with application to complex networks. Numer. Algorithms **92**(1), 503–522 (2023)

32. Ghavasieh, A., De Domenico, M.: Statistical physics of network structure and information dynamics. J. Phys. Complex. **3**(1), 011001 (2022)
33. Ghavasieh, A., Domenico, M.D.: Generalized network density matrices for analysis of multiscale functional diversity. Phys. Rev. E **107**(4), 044304 (2023)
34. Güttel, S.: Rational Krylov Methods for Operator Functions. PhD thesis, Technische Universität Bergakademie Freiberg, Germany, Dissertation available as MIMS Eprint 2017.39 (2010)
35. Güttel, S.: Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. GAMM-Mitt. **36**(1), 8–31 (2013)
36. Han, L., Escolano, F., Hancock, E.R., Wilson, R.C.: Graph characterizations from von Neumann entropy. Pattern Recognit. Lett. **33**(15), 1958–1967 (2012)
37. Henrici, P.: Applied and Computational Complex Analysis. Vol. 1. Wiley Classics Library. John Wiley & Sons, Inc., New York, (1988). Reprint of the 1974 original, A Wiley-Interscience Publication
38. Higham, N.J.: Functions of Matrices. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, Theory and Computation (2008)
39. Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Comm. Statist. Simulation Comput. **18**(3), 1059–1076 (1989)
40. Kressner, D.: Block algorithms for reordering standard and generalized Schur forms. ACM Trans. Math. Softw. **32**(4), 521–532 (2006)
41. Kubale, M.: Graph Colorings. Contemp. Math. American Mathematical Society, Providence, RI (2004)
42. Landau, L.D., Lifshitz, E.M.: Statistical Physics. Pergamon Press, London (1958)
43. Liesen, J., Strakoš, Z.: Krylov Subspace Methods. Principles and Analysis. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2013)
44. Mantica, G.: Quantum dynamical entropy and an algorithm by Gene Golub. Electr. Trans. Numer. Anal. **28**, 190–205 (2008)
45. Massei, S., Robol, L.: Rational Krylov for Stieltjes matrix functions: convergence and pole selection. BIT **61**(1), 237–273 (2021)
46. Meinardus, G.: Approximation of Functions: Theory and Numerical Methods. Expanded translation of the German edition. Translated by Larry L. Schumaker. Springer Tracts in Natural Philosophy, Vol. 13. Springer-Verlag New York, Inc., New York, (1967)
47. 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, (2021)
48. Palitta, D., Pozza, S., Simoncini, V.: The short-term rational Lanczos method and applications. SIAM J. Sci. Comput. **44**(4), A2843–A2870 (2022)
49. 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)
50. Pranić, M.S., Reichel, L.: Rational Gauss quadrature. SIAM J. Numer. Anal. **52**(2), 832–851 (2014)
51. Ruhe, A.: Rational Krylov algorithms for nonsymmetric eigenvalue problems. In: Recent Advances in Iterative Methods, volume 60 of IMA Vol. Math. Appl., pp. 149–164. Springer, New York, (1994)
52. Saibaba, A.K., Alexanderian, A., Ipsen, I.C.F.: Randomized matrix-free trace and log-determinant estimators. Numer. Math. **137**(2), 353–395 (2017)
53. Schimmel, C.: Bounds for the Decay in Matrix Functions and its Exploitation in Matrix Computations. PhD thesis, Bergische Universität Wuppertal, Wuppertal, Germany (2019)
54. Schwarz, H.R.: Tridiagonalization of a symmetric band matrix. Numer. Math. **12**(4), 231–241 (1968)
55. Taylor, A., Higham, D.J.: CONTEST: a controllable test matrix toolbox for MATLAB. ACM Trans. Math. Softw. **35**(4), 1–17 (2009)
56. von Neumann, J.: Mathematical Foundations of Quantum Mechanics. Princeton University Press, Princeton (1955)
57. Wehrl, A.: General properties of entropy. Rev. Mod. Phys. **50**, 221–260 (1978)
58. Widder, D.V.: The Laplace Transform. Princeton Mathematical Series, Princeton University Press, Princeton (1941)
59. Wihler, T.P., Bessire, B., Stefanov, A.: Computing the entropy of a large matrix. J. Phys. A **47**(24), 245201, 15 (2014)