Statistics and Computing (2022) 32:108
https://doi.org/10.1007/s11222-022-10173-4

Interpolating log-determinant and trace of the powers of matrix $\mathbf{A} + t\mathbf{B}$

Siavash Ameli[1] · **Shawn C. Shadden**[1]

Received: 1 April 2022 / Accepted: 21 October 2022 / Published online: 10 November 2022

Abstract
We develop heuristic interpolation methods for the functions $t \mapsto \log\det(\mathbf{A} + t\mathbf{B})$ and $t \mapsto \mathrm{trace}((\mathbf{A} + t\mathbf{B})^p)$ where the matrices $\mathbf{A}$ and $\mathbf{B}$ are Hermitian and positive (semi) definite and p and t are real variables. These functions are featured in many applications in statistics, machine learning, and computational physics. The presented interpolation functions are based on the modification of sharp bounds for these functions. We demonstrate the accuracy and performance of the proposed method with numerical examples, namely, the marginal maximum likelihood estimation for Gaussian process regression and the estimation of the regularization parameter of ridge regression with the generalized cross-validation method.

Keywords Parameter estimation · Gaussian process · Generalized cross-validation · Maximum likelihood method · Schatten norm · Anti-norm

1 Introduction

Estimation of the determinant and trace of matrices is a key component and often a computational challenge in many algorithms in data analysis, statistics, machine learning, computational physics, and computational biology. Some applications of trace estimation can be found in Ubaru and Saad (2018). A few examples of such applications are high-performance uncertainty quantification (Bekas et al. 2012; Kalantzis et al. 2013), optimal Bayesian experimental design (Chaloner and Verdinelli 1995), regression using Gaussian process (MacKay et al. 2003), rank estimation (Ubaru and Saad 2016), and computing observables in lattice quantum chromodynamics (Wu et al. 2016).

✉ Shawn C. Shadden
shadden@berkeley.edu

Siavash Ameli
sameli@berkeley.edu

[1] Mechanical Engineering, University of California, Berkeley, CA 94720, USA

1.1 Motivation

In this paper, we are interested in estimating the functions

$$t \mapsto \log\det(\mathbf{A} + t\mathbf{B}), \tag{1a}$$

and

$$t \mapsto \mathrm{trace}\left((\mathbf{A} + t\mathbf{B})^p\right), \tag{1b}$$

where $\mathbf{A}$ and $\mathbf{B}$ are Hermitian and positive semi-definite (positive-definite if $p < 0$), and p and t are real numbers.[1] These functions are featured in a vast number of applications in statistics and machine learning. Often, in these applications, the goal is to optimize a problem for the parameter t, and the above functions should be evaluated for a wide range of t during the optimization process.

A common example of such an application can be found in regularization techniques applied to inverse problems and supervised learning. For instance, in ridge regression by generalized cross-validation (Wahba 1977; Craven and Wahba 1978; Golub and von Matt 1997), the optimal regularization parameter t is sought by minimizing a function that involves (1b) at $p = -1$ (see Sect. 4.3). Another common usage of (1a) and (1b), for instance, is the mixed covariance functions of the form $\mathbf{A} + t\mathbf{I}$ that appear frequently in Gaussian

[1] We use boldface lowercase letters for vectors, boldface upper case letters for matrices, and normal face letters for scalars, including the components of vectors and matrices, such as x_i and H_{ij} respectively for the components of the vector $\boldsymbol{x}$ and the matrix $\mathbf{H}$.

processes with additive noise (Ameli and Shadden 2022c, d) (see also Sect. 4.2). In most of these applications, the log-determinant of the covariance matrix is common, particularly in likelihood functions or related variants. Namely, if one aims to maximize the likelihood by its derivative with respect to the parameter, the expression,

$$\frac{\partial}{\partial t} \log\det(\mathbf{A} + t\mathbf{I}) = \text{trace}\left((\mathbf{A} + t\mathbf{I})^{-1}\right),$$

frequently appears. More generally, the function (1b) for $p \in \mathbb{Z}_{<0}$ appears in the $|p|$th derivative of such likelihood functions. Other examples of (1a) and (1b) are in experimental design (Haber et al. 2008), probabilistic principal component analysis (Bishop 2006, Sect. 12.2), relevance vector machines (Tipping 2001) and (Bishop 2006, Sect. 7.2), kernel smoothing (Rasmussen and Williams 2006, Sect. 2.6), and Bayesian linear models (Bishop 2006, Sect. 3.3).

1.2 Overview of related works

The difficulty of estimating (1a) and (1b) in all the above applications is that the matrices are generally large. Also, often in these applications, cases of particular interest in (1b) are when $p < 0$, but the $|p|$th inverse of the matrix $\mathbf{A} + t\mathbf{B}$ is not available explicitly, rather it is implicitly known by matrix-vector multiplications through solving a linear system. Because of these, the evaluation of (1a) and (1b) are usually the main computational challenge in these problems, and several algorithms have been developed to address this problem.

The determinant and trace of the inverse of a Hermitian and positive-definite matrix can be calculated by the Cholesky factorization [cf. Eq. (24a) and (24b) in Sect. 4.2]. Using the Cholesky factorization, Takahashi et al. (1973) developed a method to find desired entries of a matrix inverse, such as its diagonals. The latter method was extended by Niessner and Reichert (1983). Also, Golub and Plemmons (1980) found entries of the inverse of the covariance matrix provided that the corresponding entries of its Cholesky factorization are non-zero. The complexity of this method is $\mathcal{O}(nw)$ where w is the bandwidth of the Cholesky matrix (see also Björck 1996, Sect. 6.7.4). Recently, probing and hierarchical probing methods were presented by Tang and Saad (2012) and Stathopoulos et al. (2013), respectively, to compute the diagonal entries of a matrix inverse.

In contrast to the above exact methods, many approximation methods have been developed. The stochastic trace estimator by Hutchinson (1990), which evolved from Girard (1989), uses Monte-Carlo sampling of random vectors with a Gaussian or Rademacher distribution. A similar concept was presented by Gibbs and MacKay (1997). Another randomized trace estimator was given by Avron and Toledo (2011) for symmetric and positive-definite implicit matrices. Based on the stochastic trace estimation, Wu et al. (2016) interpolated the diagonals of a matrix inverse. Also, Saibaba et al. (2017) improved the randomized estimation by a low-rank approximation of the matrix. Another tier of methods combines the idea of a stochastic trace estimator and Lanczos quadrature (Golub and Strakoš 1994; Bai et al. 1996; Bai and Golub 1997; Golub and Meurant 2009), which is known as stochastic Lanczos quadrature (SLQ). The numerical details of the SLQ method using either Lanczos tridiagonalization or Golub–Kahn bidiagonalization can be found for instance in Ubaru et al. (2017, Algorithms 1 and 2).

1.3 Objective and our contribution

Our objective is to develop a method to efficiently estimate (1a) or (1b) for a wide range of t. Note, if $\mathbf{B}$ is the identity matrix and the matrix $\mathbf{A}$ is small enough to pre-compute is eigenvalues, $\lambda_i(\mathbf{A})$, then, the evaluation of (1a) and (1b) for any t is immediate by

$$\log\det(\mathbf{A} + t\mathbf{I}) = \sum_{i=1}^{n} \log(\lambda_i(\mathbf{A}) + t), \tag{2a}$$

$$\text{trace}\left((\mathbf{A} + t\mathbf{I})^p\right) = \sum_{i=1}^{n} (\lambda_i(\mathbf{A}) + t)^p. \tag{2b}$$

However, for large matrices, estimating all eigenvalues is impractical. Thus, we herein develop an interpolation scheme for the functions (1a) and (1b) based on the following developments:

- We present a Schatten-type operator that unifies the representation of (1a) and (1b) by a single continuous function. This operator leads to definitions of an associated norm and anti-norm on matrices. Sharp inequalities for this norm and anti-norm on the sum of two Hermitian and positive (semi) definite matrices provide rough estimates for (1a) and (1b).
- We propose two interpolation methods based on the sharp norm and anti-norm inequalities mentioned above. Namely, we introduce interpolation functions based on a linear combination of orthogonal basis functions, or interpolation by rational polynomials.

We demonstrate the computational advantage of our method through two examples:

- *Gaussian process regression* We compute (1a) and (1b) in the context of marginal likelihood estimation of Gaussian process regression. We show that with very few interpolation points, an accuracy of 0.01% can be achieved.

- *Ridge regression* We estimate the regularization parameter of ridge regression with the generalized cross-validation method. We demonstrate that with only a few interpolation points, the ridge parameters can be estimated and the overall computational cost is reduced by 2 orders of magnitude.

The outline of the paper is as follows. In Sect. 2, we present matrix determinant and trace inequalities. In Sect. 3, we propose interpolation methods. In Sect. 4 we provide examples and a software package that implements the presented algorithms. In Sect. 5, we provide further applications of the method. Section 6 concludes the paper. Proofs are given in "Appendix A".

2 Determinant and trace inequalities

We will derive interpolations for (1a) and (1b) by modifying sharp bounds for these functions. In this section, we present these bounds. Without loss of generality, we temporarily omit the parameter t. However, in Sect. 3, we will retrieve the desired relations by replacing $\mathbf{B}$ with $|t|\mathbf{B}$.

Let $\mathcal{M}_{n,m}(\mathbb{C})$ denote the space of all $n \times m$ matrices with entries over the field $\mathbb{C}$. We assume $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{n,n}(\mathbb{C})$ are Hermitian and positive semi-definite. Furthermore, for $p < 0$, we require matrices $\mathbf{A}$ and $\mathbf{B}$ to be positive-definite. The notations $\mathbf{A} \succ \mathbf{B}$ and $\mathbf{A} \succeq \mathbf{B}$ on matrices $\mathbf{A}$ and $\mathbf{B}$ denotes $\mathbf{A} - \mathbf{B}$ is positive-definite and positive semi-definite, respectively. Also, $\lambda(\mathbf{A}) := (\lambda_1(\mathbf{A}), \ldots, \lambda_n(\mathbf{A}))$ indicates the n-tuple of eigenvalues of matrix $\mathbf{A}$.

Define a Schatten-class operator $\|\cdot\|_p : \mathcal{M}_{n,n}(\mathbb{C}) \mapsto \mathbb{R}_{\geq 0}$ by

$$\|\mathbf{A}\|_p := \begin{cases} (\det(|\mathbf{A}|))^{\frac{1}{n}}, & p = 0, \\ \left(\frac{1}{n}\operatorname{trace}(|\mathbf{A}|^p)\right)^{\frac{1}{p}}, & p \in \mathbb{R} \setminus \{0\}, \end{cases} \tag{3}$$

where $|\mathbf{A}| := \sqrt{\mathbf{A}^*\mathbf{A}}$ and $\mathbf{A}^*$ denotes the Hermitian transpose of $\mathbf{A}$. Since we assume the matrices are Hermitian and at least positive semi-definite, we omit $|\cdot|$ in subsequent expressions. Also, we note that $p \mapsto \|\cdot\|_p$ is continuous at $p = 0$ since

$$\|\mathbf{A}\|_0 = \lim_{p \to 0} \|\mathbf{A}\|_p. \tag{4}$$

Namely, (4) is justified by observing that $\|\mathbf{A}\|_p = M_p(\lambda(\mathbf{A}))$, where $M_p(\lambda(\mathbf{A}))$ is the generalized mean of $\lambda(\mathbf{A})$ defined by

$$M_p(\lambda(\mathbf{A})) := \begin{cases} \left(\prod_{i=1}^n \lambda_i(\mathbf{A})\right)^{\frac{1}{n}}, & p = 0, \\ \left(\frac{1}{n}\sum_{i=1}^n \lambda_i^p(\mathbf{A})\right)^{\frac{1}{p}}, & p \in \mathbb{R} \setminus \{0\}. \end{cases} \tag{5}$$

It is known that the generalized mean converges to the geometric mean, M_0, as $p \to 0$ (Hardy et al. 1952, p. 15), which concludes (4).

For $p \in [1, \infty)$, the operator $\|\cdot\|_p$ is an equivalent norm to the Schatten p-norm of $\mathbf{A}$. Conventionally, the Schatten norm is defined without the normalizing factor $\frac{1}{n}$ in (3), but the inclusion of this factor is justified by the continuity granted by (4). The Schatten norm is a subadditive function, meaning that it satisfies the triangle inequality

$$\|\mathbf{A} + \mathbf{B}\|_p \leq \|\mathbf{A}\|_p + \|\mathbf{B}\|_p. \tag{6a}$$

The reverse triangle inequality follows from the above by

$$\|\mathbf{A} - \mathbf{B}\|_p \geq \|\mathbf{A}\|_p - \|\mathbf{B}\|_p, \tag{6b}$$

provided that $\mathbf{A} \succeq \mathbf{B}$ for (6b) to hold. For $p = 1$, the two above relations become equality by the additivity of the trace operator.

When $p < 1$, the operator (3) is no longer a norm, rather, is an anti-norm (Bourin and Hiai 2011) that satisfies super-additivity property

$$\|\mathbf{A} + \mathbf{B}\|_p \geq \|\mathbf{A}\|_p + \|\mathbf{B}\|_p. \tag{7a}$$

A reversed inequality can also be derived from the above as

$$\|\mathbf{A} - \mathbf{B}\|_p \leq \|\mathbf{A}\|_p - \|\mathbf{B}\|_p, \tag{7b}$$

provided that $\mathbf{A} \succeq \mathbf{B}$ (or $\mathbf{A} \succ \mathbf{B}$ if $p < 0$) for (7b) to hold. We also note that inequality (7a) at $p = 0$ reduces to Brunn–Minkowski determinant inequality (Horn and Johnson 1990, p. 482, Theorem 7.8.8)

$$(\det(\mathbf{A} + \mathbf{B}))^{\frac{1}{n}} \geq (\det(\mathbf{A}))^{\frac{1}{n}} + (\det(\mathbf{B}))^{\frac{1}{n}}. \tag{8}$$

For proofs and discussions of a general class of anti-norms, which includes (3), we refer the reader to Bourin and Hiai (2011, 2014). However, in "Appendix A", we provide a direct proof of (7a) and (7b) and the necessary and sufficient conditions for equality to hold in these relations for the operator (3) at $p < 1$.

Remark 1 (*Comparisons to other inequalities*) There are other known bounds to functions (1a) and (1b). For instance, for the common case of $p = -1$, we can obtain the upper bound (Zhang 2011, p. 210, Theorem 7.7)

$$\operatorname{trace}\left((\mathbf{A} + \mathbf{B})^{-1}\right) \leq \frac{1}{4}\left(\operatorname{trace}(\mathbf{A}^{-1}) + \operatorname{trace}(\mathbf{B}^{-1})\right). \tag{9}$$

Also, a lower bound can be obtained, for instance, by the arithmetic-harmonic mean inequality $M_{-1}(\lambda(\mathbf{A} \pm \mathbf{B})) \leq$

$M_1(\lambda(\mathbf{A} \pm \mathbf{B}))$, where M_{-1} and M_1 are the harmonic mean and arithmetic mean, respectively, (Mitrinović and Vasić 1970, Ch. 2, Theorem 1), which leads to

$$\operatorname{trace}((\mathbf{A} \pm \mathbf{B})^{-1}) \geq \frac{n^2}{\operatorname{trace}(\mathbf{A}) \pm \operatorname{trace}(\mathbf{B})}. \tag{10}$$

The inequalities (9) or (10), however, are not as useful as the inequality (7a) for $p = -1$, since if $\mathbf{B}$ is either too small or too large compared to $\mathbf{A}$, (9) and (10) do not asymptote to equality, whereas (7a) and (7b) become asymptotic equalities, which is a desired property for our purpose.

3 Interpolation of determinant and trace

We use the bounds provided by inequalities (6a), (6b), (7a), and (7b) to interpolate the functions (1b) and (1a). To this end, we replace the matrix $\mathbf{B}$ with $|t|\mathbf{B}$ in the bounds found in Sect. 2. Define

$$\tau_p(t) := \frac{\|\mathbf{A} + t\mathbf{B}\|_p}{\|\mathbf{B}\|_p}, \qquad \text{and} \qquad \tau_{p,0} := \tau_p(0).$$

We assume $\tau_{p,0}$ is known by directly computing $(\frac{1}{n}\operatorname{trace}(\mathbf{A}^p))^{\frac{1}{p}}$ and $(\frac{1}{n}\operatorname{trace}(\mathbf{B}^p))^{\frac{1}{p}}$ when $p \neq 0$, or $(\det(\mathbf{A}))^{\frac{1}{n}}$ and $(\det(\mathbf{B}))^{\frac{1}{n}}$ when $p = 0$.[2] Then, (6a) and (7a) imply

$$\tau_p(t) \leq \tau_{p,0} + t, \qquad p \geq 1, \quad t \in [0, \infty), \tag{11a}$$
$$\tau_p(t) \geq \tau_{p,0} + t, \qquad p < 1, \quad t \in [0, \infty), \tag{11b}$$

and (6b) and (7b) imply

$$\tau_p(t) \geq \tau_{p,0} + t, \qquad p \geq 1, \quad t \in (t_{\inf}, 0], \tag{11c}$$
$$\tau_p(t) \leq \tau_{p,0} + t, \qquad p < 1, \quad t \in (t_{\inf}, 0], \tag{11d}$$

where $t_{\inf} := \inf\{t \mid \mathbf{A} + t\mathbf{B} \succ \mathbf{0}\}$. The above sharp inequalities become equality at $t = 0$. Also, (11a) and (11b) become asymptotic equalities as $t \to \infty$. Based on the above, the bound function

$$\hat{\tau}_p(t) := \tau_{p,0} + t, \tag{12}$$

can be regarded as a reasonable approximation of $\tau_p(t)$ at $|t| \ll \tau_{p,0}$ where $\tau_p(t) \approx \tau_{p,0}$, and at $t \gg \tau_{p,0}$ where $\tau_p(t) \approx t$. We expect $\hat{\tau}_p(t)$ to deviate the most from $\tau_p(t)$ when $\mathcal{O}(t\tau_{p,0}^{-1}) \approx 1$.

Furthermore, to improve the approximation in the intermediate interval $t\tau_{p,0}^{-1} \in (c, c^{-1})$ for some $c \ll 1$, we define interpolating functions based on the above bounds to honor the known function values at some intermediate points $t_i \in (c\tau_{p,0}, c^{-1}\tau_{p,0})$. In particular, we specify interpolation points over logarithmically spaced intervals, because t is usually varied in a wide logarithmic range in most applications. We compute the function values at the interpolation points, $\tau_p(t_i)$, with any of the trace estimation methods mentioned earlier.

Many types of interpolating functions can be employed to improve the above approximation. However, we seek interpolating functions whose parameters can be easily obtained by solving a linear system of equations. We define two such types of interpolations, namely, by a linear combination of basis functions and by rational polynomials, respectively in Sects. 3.1 and 3.2.

3.1 Interpolation with a linear combination of basis functions

Based on (12), we define an interpolating function $\tilde{\tau}_p(t)$ by

$$\tilde{\tau}_p(t) := \tau_{p,0} + \sum_{i=0}^{q} w_i \phi_i(t), \tag{13}$$

where ϕ_i are basis functions and w_i the weights. The basis functions

$$\phi_i(t) = t^{\frac{1}{i+1}}, \qquad i = 0, \ldots, q, \tag{14}$$

for the domain $t \in [0, \infty)$ can be used, which are inverse functions of the monomials and we refer to them as inverse-monomials. These basis functions satisfy the conditions $\phi_0(t) = t$, $\phi_i(0) = 0$, and $\phi_0(t) \gg \phi_i(t)$, $i > 0$ when $t \gg 1$. For consistency with (12), we set $w_0 = 1$. The coefficients $w_i, i = 1, \ldots, q$ are found by solving a linear system of q equations using a priori known values $\tau_{p,i} := \tau_p(t_i)$, $i = 1, \ldots, q$. When $q = 0$, no intermediate interpolation point is introduced and the approximation function is the same as the bound $\hat{\tau}_p(t)$ given by (12).

Remark 2 An alternative could be to use monomials t^i for interpolation functions, e.g.,

$$\tilde{\tau}_p(t)^{q+1} := \tau_{p,0}^{q+1} + \sum_{i=1}^{q+1} w_i t^i, \tag{15}$$

with $w_{q+1} = 1$, and the rest of the weights $w_i, i = 1, \ldots, q$ determined from the known values of the function. This is not particularly useful in practice, as the exponentiation terms, t^i, cause arithmetic underflows; also, Runge's phenomenon occurs even for low-order interpolations $q > 1$.

[2] Computing the determinant directly should be avoided as it can be a very large number. Rather, $(\det(\cdot))^{\frac{1}{n}}$ can be computed via $\exp(\frac{1}{n}\log\det(\cdot))$. See Sect. 4.2 for an example.

In practice, just a few interpolating points t_i are sufficient to obtain a reasonable interpolation of $\tau_p(t)$. However, when more interpolation points are used (such as when $p \geq 6$), the linear system of equations for the weights w_i becomes ill-conditioned. To overcome this issue, orthogonal basis functions can be used (see e.g., Seber and Lee 2012, Sect. 7.1 for a general discussion).

For our application, we seek basis functions $\phi_i^\perp(t)$ that are orthogonal on the unit interval $t \in [0, 1]$. Since we are interested in functions in the logarithmic scale of t, we define the inner product in the space of functions using the Haar measure $\mathrm{d}\log(t) = \mathrm{d}t/t$. Applicability of Haar measure can be justified by letting $t_i = e^{x_i}$, where x_i are normally spaced interpolant points. Following the discussion of Seber and Lee (2012, Sect. 7.1) for linear regression using orthogonal polynomials, we use the conventional integrals with the Lebesgue measure $\mathrm{d}x$ to define the inner product of functions. The measure $\mathrm{d}x$ is equivalent to the Haar measure $\mathrm{d}\log t$ for the variable t.

The desired orthogonality condition in the Hilbert space of functions on [0, 1] with respect to the Haar measure becomes

$$\langle \phi_i^\perp, \phi_j^\perp \rangle_{L^2([0,1],\mathrm{d}t/t)} = \int_0^1 \phi_i^\perp(t)\phi_j^\perp(t)\frac{\mathrm{d}t}{t} = \delta_{ij}, \tag{16}$$

where δ_{ij} is the Kronecker delta function. A set of orthogonal functions $\phi_i^\perp(t)$ can be constructed from the set of non-orthogonal basis functions $\{\phi_i\}_{i=1}^q$ in (14) by recursive application of Gram-Schmidt orthogonalization

$$\phi_i^\perp(t) = \alpha_i \sum_{j=1}^{q} a_{ij}\phi_j(t), \qquad i = 1, \ldots, q. \tag{17}$$

The first nine orthogonal basis functions are shown in Fig. 1 and the respective coefficients α_i and a_{ij} are given by Table 1.[3]

A set of orthogonal functions can also be defined on intervals other than [0, 1] by adjusting the bounds of integration in (16), which yields a different set of function coefficients. However, it is more convenient to fix the domain of orthogonal functions to the unit interval [0, 1], and later scale the domain as desired, e.g., to $[0, l]$ where $l := \max(t_i)$. Although this approach does not lead to orthogonal functions in $[0, l]$, it nonetheless produces a well-conditioned system of equations for the weights w_i.

Remark 3 The interpolation function defined in (13) asymptotes consistently to $\tilde{\tau}_p(t) \to t$ at $t \gg \tau_{p,0}$. On the other end, the convergence $\tilde{\tau}_p(t) \to \tau_{p,0}$ at $t \ll \tau_{p,0}$ is not uniform,

[3] We developed the python package `ortho` to generate an arbitrary number of orthogonal functions $\phi_j^\perp(t)$ using symbolic computations. See https://ameli.github.io/ortho for details.

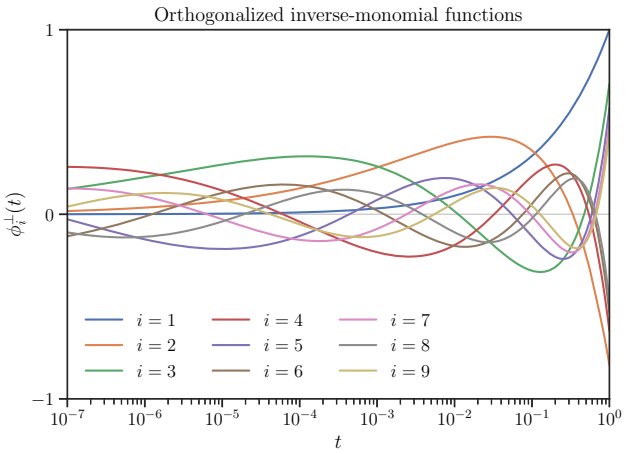

Fig. 1 Orthogonalized inverse-monomial functions $\phi_i^\perp(t)$ in the logarithmic scale of t

rather the interpolation function oscillates. This behavior is originated from the basis functions ϕ_i, $i > 0$, that are not independent at $t \ll \tau_{p,0}$, particularly, near the origin. This dependency of basis functions cannot be resolved by the orthogonalized functions $\phi_i^\perp$, as they are orthogonal with respect to the singular weight function t^{-1} at the origin. Thus, (13) should not be employed on very small logarithmic scales, rather, other interpolation functions should be employed for such purpose, such as presented in Sect. 3.2.

3.2 Interpolation with rational polynomials

We define another type of interpolating function that can perform well at small scales of t, by using rational polynomials. Define

$$\tilde{\tau}_p(t) := \frac{t^{q+1} + a_q t^q + \cdots + a_1 t + a_0}{t^q + b_{q-1}t^{q-1} + \cdots + b_1 t + b_0}, \tag{18}$$

which is the Padé approximation of τ_p of order $[q + 1, q]$. We set $a_0 = b_0\tau_{p,0}$ in order to satisfy $\tau_p(0) = \tau_{p,0}$. Also, we note that the above interpolation satisfies the asymptotic relation $\tau_p(t) \to t$ as $t \to \infty$. At $q = 0$, when no interpolant point is used, the above interpolation function falls back to (12) by setting $b_0 = 1$. For $q > 0$, $2q$ interpolant points t_i are needed to solve the linear system of equations for the coefficients $a_1, \ldots, a_q$ and $b_0, \ldots, b_{q-1}$.

An alternative rational polynomial is the Chebyshev rational function (Guo et al. 2002)

$$r_i(t) := T_i\left(\frac{t-1}{t+1}\right), \tag{19}$$

where T_i, $i \in \mathbb{N}$ are the Chebyshev polynomials of the first kind. The Chebyshev rational functions are orthogonal in $[0, \infty)$ with respect to the weight function $(t + 1)^{-1}\sqrt{t}$

Table 1 Coefficients of orthogonal functions in (17)

i	α_i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}	a_{i8}	a_{i9}
1	$+\sqrt{2/2}$	1								
2	$-\sqrt{2/3}$	6	−5							
3	$+\sqrt{2/4}$	20	−40	21						
4	$-\sqrt{2/5}$	50	−175	210	−84					
5	$+\sqrt{2/6}$	105	−560	1134	−1008	330				
6	$-\sqrt{2/7}$	196	−1470	4410	−6468	4620	−1287			
7	$+\sqrt{2/8}$	336	−3360	13860	−29568	34320	−20592	5005		
8	$-\sqrt{2/9}$	540	−6930	37422	−108108	180180	−173745	90090	−19448	
9	$+\sqrt{2/10}$	825	−13200	90090	−336336	750750	−1029600	850850	−388960	75582

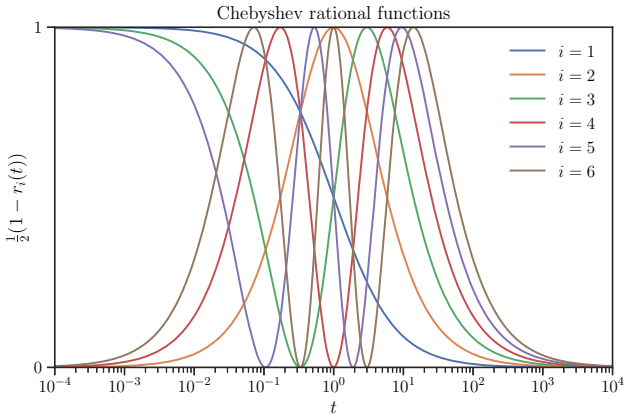

Fig. 2 Chebyshev rational functions (excluding r_0) used in (20) in the logarithmic scale of t

and satisfy the recursive relation $r_{i+1}(t) = 2((t-1)/(t+1))r_i(t) - r_{i-1}(t)$ with $r_0(t) = 1$ and $r_1(t) = (t-1)/(t+1)$. An interpolation of $\tau_p(t)$ using Chebyshev rational functions can be given by

$$\frac{\tilde{\tau}_p(t)}{\tau_{p,0} + t} - 1 = \sum_{i=1}^{q+1} \frac{w_i}{2} \left(1 - r_i\left(\frac{t}{\alpha}\right)\right), \tag{20}$$

where $\alpha > 0$ is a given scale parameter and will be explained shortly. Both sides of the above relation converge to zero at $t \to \infty$. To satisfy $\tau_p(0) = \tau_{p,0}$, we require $\sum_{i=1}^{q+1} \frac{w_i}{2}(1 - (-1)^i) = 0$ considering $r_i(0) = (-1)^i$. The latter condition together with q linear equations on the interpolating points $t_i > 0$, $i = 1, \ldots, q$ solve the weights w_i. An advantage of using the above interpolation scheme is that we can arrange the interpolant points t_i on the corresponding Chebyshev nodes to reduce the interpolation error.

Figure 2 shows the Chebyshev rational basis functions in the form that are used on the right-hand side of (20). These basis functions converge at $t \ll 1$ and $t \gg 1$, whereas the main variability of these functions is mostly observed near $\mathcal{O}(t) \approx 1$. Thus, it is desirable to shift the interval of interpolation to the vicinity of $\mathcal{O}(t) = 1$, which can be achieved by setting the scale parameter α. One approach to find an optimal interpolation parameter, α, is to minimize the curvature of the interpolating function, which is a common practice, for instance, in smoothing splines (Newbery and Garrett 1991). To this end, let $w_i(\alpha)$ denote the solved weights for a given α. For simplicity, we transform the graph $(t, \tilde{\tau}_p)$ to (x, y_α) where $x := (t-\alpha)/(t+\alpha)$ and $y_\alpha(x) := \sum_{i=1}^{q+1} \frac{w_i(\alpha)}{2}(1 - T_i(x))$. Then, an optimal α^* can be sought to minimize the arc integral of curvature squared of $y_\alpha(x)$ by

$$\alpha^* = \arg\min_\alpha \int_{-1}^{1} \frac{|y''_\alpha(x)|^2}{(1 + |y'_\alpha(x)|^2)^{\frac{5}{2}}} \mathrm{d}x. \tag{21}$$

We note that in the absence of enough interpolant points, minimizing the curvature of the interpolating curve does not necessarily reduce interpolation error. However, when an adequate number of interpolant points are employed, the above approach can practically lead to a scale parameter α that enhances the interpolation.

Finally, we note that unlike the interpolation scheme of Sect. 3.1 with the inverse monomial basis (14), both the Padé approximation of (18) and Chebyshev rational interpolation in (20) can interpolate τ_p at negative values of t, namely in the domain $t > t_{\text{inf}}$ when $t_{\text{inf}} < 0$ [see (11c) and (11d)].

4 Numerical examples

In Sect. 4.1, we briefly introduce a software package we developed for the presented numerical algorithm. This package was used to produce the results in Sects. 4.2 and 3.2. Indeed, the source code to reproduce the results and plots in the following sections can be found on the documentation of

Listing 1 A minimalistic usage of `imate.InterpolateSchatten` class

```
# Install imate with "pip install imate"
from numpy import logspace
import imate

# Generate a sample correlation matrix using
# the kernel e^{-r/0.1}.
A = imate.sample_matrices.correlation_matrix(
        50, dimension=2, kernel='exponential',
        scale=0.1)

# Create an interpolating object for
# f_p : t ↦ ||A + tI||_p,  p = -1.
f = imate.InterpolateSchatten(A, B=None, p=-1,
        ti=logspace(-4, 3, 8), kind='IMBF',
        options={'method': 'cholesky'})

# Interpolate 1000 points in [10^{-4}, 10^{3}].
t = logspace(-4, 3, 1000)
y = f(t)

# Plot the interpolated normalized curve τ_p(t) =
# ||A + tI||_p / ||I||_p, compare with exact values.
f.plot(t, compare=True, normalize=True)
```

the software package.[4] Section 4.2 considers the problem of marginal likelihood estimation, which considers a full rank correlation matrix, and for this we use the interpolation functions of Sect. 3.1. Section 4.3 considers the problem of ridge regression, which considers a singular matrix, and for this we use the rational polynomial interpolation method of Sect. 3.2. We note that the interpolation with Chebyshev rational functions provide similar results to the orthogonalized inverse-monomials in (13) and we omit in our numerical examples for brevity.

4.1 Software package

The methods developed in this manuscript have been implemented into the python package `imate`, an implicit matrix trace estimator (Ameli and Shadden 2022b). This library estimates the determinant and trace of various functions of implicit matrices using either direct or stochastic estimation techniques and can process both dense matrices and large-scale sparse matrices. The main library of this package is written in C++ and NVIDIA® CUDA and accelerated on both parallel CPU processors and CUDA-capable multi-GPU devices. The `imate` library is employed in the python package `glearn`, a machine learning library using Gaussian process regression (Ameli and Shadden 2022a).

In Lisiting 1, we demonstrate a minimalistic usage of `imate.InterpolateSchatten` class that interpolates $f_p : t \mapsto \|\mathbf{A} + t\mathbf{B}\|_p$. Briefly, Line 9 generates a sample correlation matrix $\mathbf{A} \in \mathcal{M}_{n,n}(\mathbb{R})$ on a randomly generated set of $n = 50^2$ points using an exponential decay kernel. In Line 15, we create an instance of the class `imate.InterpolateSchatten`. Setting `B=None` indicates $\mathbf{B}$ is the identity matrix using an efficient implementation that does not require storing identity matrix. The instantiation in Line 19 internally computes $\tau_{-1,i} = \tau_{-1}(t_i)$ on eight interpolant points $t_i = 10^{-4}, 10^{-3}, \ldots, 10^{3}$ and obtains the interpolation coefficients for the orthogonalized inverse-monomial basis functions (14) since `kind='IMBF'` was specified. Other possible methods can be the exact method with no interpolation (`EXT`), eigenvalue method (`EIG`) given in (2b), monomial basis functions (`MBF`) given in (15), Padé rational polynomial functions (`RPF`) given in (18), Chebyshev rational functions (`CRF`) given in (20), or radial basis functions (`RBF`) (which we do not cover herein for brevity). The evaluation of $\tau_{-1,i}$ can be configured by passing a dictionary of settings to the `options` argument, and we refer the interested reader to the package documentation for further details. In this minimalistic example, we compute $\tau_{-1,i}$ using Cholesky decomposition, as further detailed in Sect. 4.2. Other methods include stochastic Lanczos quadrature or Hutchinson estimation; we compare such methods in Sect. 4.3. Once the interpolation object is initialized, future calls to interpolate an arbitrary number of points t are returned almost instantly. In Line 19, the interpolation is performed on 1000 points in the interval $[10^{-4}, 10^{3}]$ spaced uniformly on the logarithmic scale. A comparison of the interpolated result versus the exact solution are shown in Fig. 3. It can be seen that with only eight interpolant points, the relative error of interpolation over a wide range of parameter t is around 0.1%.

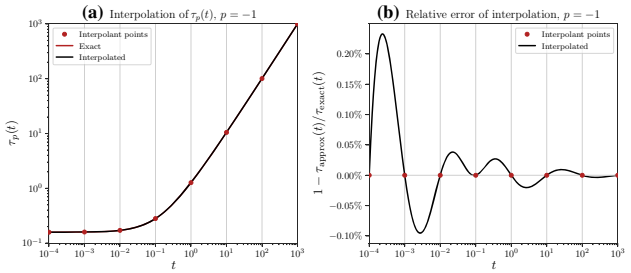

Fig. 3 Result of the code in Lisiting 1. **a** Comparison of interpolated versus exact value of the function $\tau_{-1}(t)$. The exact function (red curve) is overlaid by the interpolated curve (black curve). **b** Relative error of the comparison

4.2 Marginal likelihood estimation for Gaussian process regression

Here we generate a full rank correlation matrix from a spatially correlated set of points $\boldsymbol{x} \in \mathcal{D} = [0, 1]^2$. To define a

[4] See https://ameli.github.io/imate.

spatial correlation function, we use the isotropic exponential decay kernel given by

$$\kappa(\boldsymbol{x}, \boldsymbol{x}'|\rho) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\rho}\right), \tag{22}$$

where ρ is the correlation scale, set to $\rho = 0.1$. The above exponential decay kernel represents an Ornstein-Uhlenbeck random process, which is a Gaussian and zeroth-order Markov process (Rasmussen and Williams 2006, p. 85). To produce discrete data, we sample $n = 50^2$ points from $\mathcal{D}$, which yields the symmetric and positive-definite correlation matrix $\mathbf{A}$ with the components $A_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j|\rho)$. We aim to interpolate functions

$$\log\det(\mathbf{A}+t\mathbf{I}) = n\log\tau_0(t), \tag{23a}$$
$$\text{trace}\left((\mathbf{A}+t\mathbf{I})^p\right) = n(\tau_p(t))^p, \tag{23b}$$

for $p = -1, -2$, which appear in many statistical applications, such as the estimation of noise in Gaussian process regression (Ameli and Shadden 2022c). Specifically, the above functions for $p = 0, -1$, and -2 appear in the corresponding likelihood function, and its Jacobian and Hessian, respectively.

We compute the exact value of $\tau_p(t)$ for $p \in \mathbb{Z}_{\le 0}$ (either at interpolant points t_i or at all points t for the purpose of benchmark comparison) as follows. We compute the Cholesky factorization of $(\mathbf{A}+t\mathbf{I})^{|p|} = \mathbf{L}_{|p|}\mathbf{L}_{|p|}^\intercal$, where $\mathbf{L}_{|p|}$ is lower triangular. Then

$$\log\det(\mathbf{A}+t\mathbf{I}) = 2\sum_{i=1}^{n}\log((\mathbf{L}_1)_{ii}), \tag{24a}$$

$$\begin{aligned}\text{trace}\left((\mathbf{A}+t\mathbf{I})^p\right) &= \text{trace}(\mathbf{L}_{|p|}^{-\intercal}\mathbf{L}_{|p|}^{-1})\\ &= \text{trace}(\mathbf{L}_{|p|}^{-1}\mathbf{L}_{|p|}^{-\intercal})\\ &= \|\mathbf{L}_{|p|}^{-1}\|_F^2, \quad p \in \mathbb{Z}_{<0},\end{aligned} \tag{24b}$$

where $(\mathbf{L}_1)_{ii}$ is the ith diagonal element of $\mathbf{L}_1$ and $\|\cdot\|_F$ is the Frobenius norm. The second equality in (24b) employs the cyclic property of the trace operator. A simple method to compute $\|\mathbf{L}_{|p|}^{-1}\|_F^2$ without storing $\mathbf{L}_{|p|}^{-1}$ is to solve the lower triangular system $\mathbf{L}_{|p|}\boldsymbol{x}_i = \boldsymbol{e}_i$ for $\boldsymbol{x}_i$, $i = 1, \dots, n$, where $\boldsymbol{e}_i = (0, \dots, 0, 1, 0, \dots, 0)^\intercal$ is a column vector of zeros, except, its ith entry is one. The solution vector $\boldsymbol{x}_i$ is the ith column of $\mathbf{L}_{|p|}^{-1}$. Thus, $\|\mathbf{L}_{|p|}^{-1}\|_F^2 = \sum_{i=1}^n \|\boldsymbol{x}_i\|^2$. This method is memory efficient since the vectors $\boldsymbol{x}_i$ do not need to be stored.

We note that the complexity of the interpolation method is the number of evaluations of τ_p at interpolant points t_i and at $t = 0$ (which is proportional to q) times the complexity of computing τ_p at a single point t. For instance, by using the Cholesky method in (24a) or (24b) which costs $\mathcal{O}(\frac{1}{3}n^3)$ for a matrix of size n, the complexity of the interpolation method is $\mathcal{O}(\frac{1}{3}n^3 q)$.

Remark 4 (Case of Sparse Matrices) There exist efficient methods to compute the Cholesky factorization of sparse matrices (see e.g., Davis 2006, Ch. 4). Also, the inverse of the sparse triangular matrix $\mathbf{L}_{|p|}$ can be computed at $\mathcal{O}(n^2)$ complexity (Stewart 1998, pp. 93-95), and a linear system with both sparse kernel $\mathbf{L}_{|p|}$ and sparse right-hand side $\boldsymbol{e}_i$ can be solved efficiently (see Davis 2006, Sect. 3.2).

The exact value of $\tau_p(t)$, for $p = 0, -1, -2$, computed directly using the Cholesky factorization method described above are respectively shown in Fig. 4a, c, e by the solid black curve (overlaid by the red curve) in the range $t \in [10^{-4}, 10^3]$. The dashed black curves in Fig. 4a, c, e are the lower bounds $\hat{\tau}_p(t)$ given by (12), which can be thought of as the estimation with zero interpolant points, i.e., $q = 0$. For completeness, we have also shown the upper bound of $\tau_{-1}(t)$ by the black dash-dot line in Fig. 4c, given by

$$\check{\tau}_{-1}(t) := 1 + t \ge \tau_{-1}(t). \tag{25}$$

The above upper bound can be obtained from (10) and the fact that $\text{trace}(\mathbf{A}) = n$, since the diagonals of the correlation matrix are 1. However, unlike the lower bound in (7a), the upper bound (25) is not useful for approximation as it does not asymptote to $\tau_{-1}(t)$ at small t. Nonetheless, both the lower and upper bounds asymptote to t at large t.

To estimate τ_p, we used the interpolation function in (13) with the set of orthonormal basis functions in Table 1. The colored solid lines in Fig. 4a, c, e are the interpolations $\tilde{\tau}_p(t)$ with $q = 1, 3, 5, 7$, and 9 interpolant points, t_i, spanning from 10^{-4} to 10^3. It can be seen from the embedded diagrams in Fig. 4a, c, e that $\tilde{\tau}_p(t)$ is remarkably close to the true function value. In practice, fewer interpolant points in a small range, e.g., $[10^{-2}, 10^2]$, are sufficient to effectively interpolate τ_p.

To better compare the exact and interpolated functions, the relative error of the interpolations is shown in Fig. 4b, d, f. The relative error of the lower bound (dashed curve) rapidly vanishes at both ends, namely, at $t \ll \tau_{p,0}$ and $t \gg \tau_{p,0}$, where $\tau_{0,0} = 0.22$, $\tau_{-1,0} = 0.16$, and $\tau_{-2,0} = 0.14$. The absolute error of the upper bound is highest at $\mathcal{O}(t\tau_{p,0}^{-1}) = 1$, or $t \approx \tau_{p,0}$, which is slightly to the left of the relative error peak on each diagram.

Based on the lower bound, we distribute the interpolant points, t_i, almost evenly around $t \approx \tau_{p,0}$ where the lower bound has the highest error. The blue curve in Fig. 4b, d, f corresponds to the case with only one interpolation point at $t_1 = 10^{-1}$, which already leads to a relative error less than 3% almost everywhere. On the other hand, with only nine interpolation points $t_i \in \{10^{-4}, 4\times10^{-4}, 10^{-3}, 10^{-2}, \dots, 10^3\}$ the relative error becomes less than 0.01%. Beyond the strong

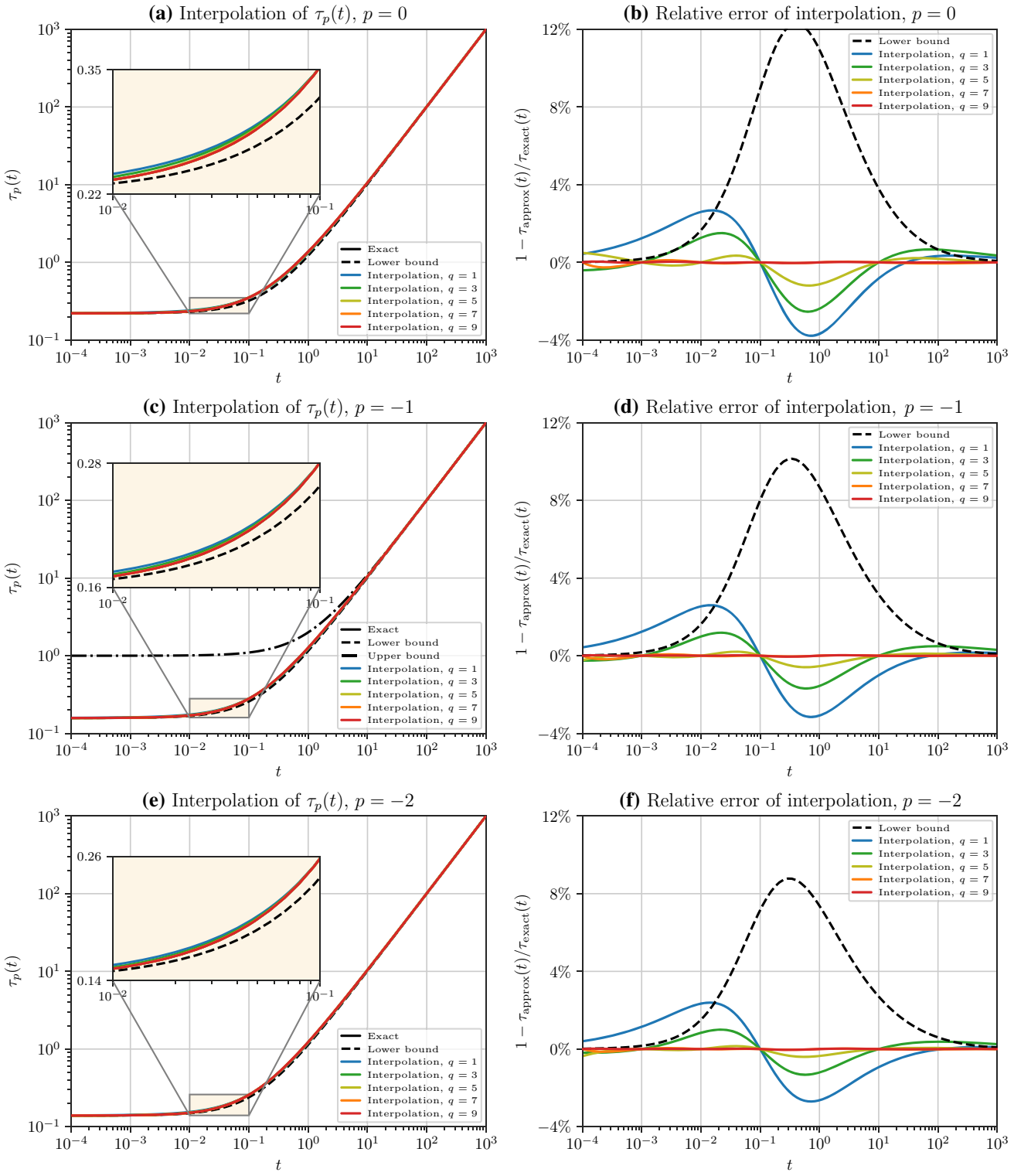

Fig. 4 Left columns: Comparison of the exact function $\tau_p(t)$, bounds $\hat{\tau}_p$, and the interpolations $\tilde{\tau}_p(t)$ for various numbers of interpolant points. The interpolation becomes almost indistinguishable from the exact solution once 5 or more interpolation points are used. Right columns: Relative error of the interpolations and the bounds. The interpolations using 7 and 9 points lead to relative errors of less than 0.02% and 0.01%, respectively. Rows correspond to $p = 0$, -1, and -2, respectively

accuracy shown by the relative errors, the absolute errors are more compelling since $\tau_p(t)$ decays by orders of magnitude at large t, making the absolute error negligible at $t \gg \tau_{p,0}$.

4.3 Ridge regression with generalized cross-validation

Here we calculate the optimal regularization parameter for a linear ridge regression model using generalized cross-validation (GCV). Consider the linear model $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{y} \in \mathbb{R}^n$ is a column vector of given data, $\mathbf{X} \in \mathcal{M}_{n,m}(\mathbb{R})$ is the known design matrix representing m basis functions where $m < n$, $\boldsymbol{\beta} \in \mathbb{R}^m$ is the unknown coefficients of the linear model, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{K})$ is the correlated residual error of the model, which is a zero-mean Gaussian random vector with the symmetric and positive-definite correlation matrix $\mathbf{K}$ and unknown variance σ^2. A generalized least-squares solution to this problem minimizes the square Mahalanobis distance $\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2_{\mathbf{K}^{-1}} := (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\intercal \mathbf{K}^{-1}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})$ yielding an estimation of $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}\boldsymbol{y}$ (Seber and Lee 2012, p. 67).

When $\mathbf{X}$ is not full rank, the least-squares problem is not well-conditioned. A resolution of the ill-conditioned problems is the ridge (Tikhonov) regularization, where the function $\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2_{\mathbf{K}^{-1}} + n\theta\|\boldsymbol{\beta}\|^2_{\boldsymbol{\Omega}}$ is minimized instead (Seber and Lee 2012, Sect. 12.5.2). Here, the penalty term is $\|\boldsymbol{\beta}\|^2_{\boldsymbol{\Omega}} = \boldsymbol{\beta}^\intercal\boldsymbol{\Omega}\boldsymbol{\beta}$ where $\boldsymbol{\Omega}$ is the symmetric and positive-definite penalty matrix. The estimate of $\boldsymbol{\beta}$ using the penalty term becomes

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X} + n\theta\boldsymbol{\Omega})^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}\boldsymbol{y}. \tag{26}$$

Also, the fitted values on the training points are $\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$, which can be written as $\hat{\boldsymbol{y}} = \mathbf{S}_\theta\boldsymbol{y}$, where the smoother matrix $\mathbf{S}_\theta$ is defined by

$$\mathbf{S}_\theta := \mathbf{X}(\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X} + n\theta\boldsymbol{\Omega})^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}. \tag{27}$$

The regularization parameter, θ, plays a crucial role to balance the residual error versus the added penalty term. The generalized cross-validation method (Wahba 1977; Craven and Wahba 1978; Golub et al. 1979) is a popular way to seek an optimal regularization parameter without needing to estimate the error variance σ^2. Namely, the regularization parameter is sought as the minimizer of

$$V(\theta) := \frac{\frac{1}{n}\|(\mathbf{I} - \mathbf{S}_\theta)\boldsymbol{y}\|^2_{\mathbf{K}^{-1}}}{\left(\frac{1}{n}\operatorname{trace}(\mathbf{I} - \mathbf{S}_\theta)\right)^2}, \tag{28}$$

(Hastie et al. 2001, p. 244).[5] For large matrices, it is difficult is to compute $\operatorname{trace}(\mathbf{S}_\theta)$ (also known as the effective degrees of freedom) in the denominator of (28), and several methods have been developed to address this problem (Golub and von Matt 1997; Lukas et al. 2010).

4.3.1 Estimating the trace

Using the presented interpolation method, we aim to estimate $\operatorname{trace}(\mathbf{S}_\theta)$. Let $\boldsymbol{\Omega} = \mathbf{L}\mathbf{L}^\intercal$ be the Cholesky decomposition of $\boldsymbol{\Omega}$. Using the cyclic property of trace operator, we have

$$\begin{aligned}
&\operatorname{trace}(\mathbf{S}_\theta) \\
&\quad = \operatorname{trace}((\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X} + n\theta\boldsymbol{\Omega})^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X}) \\
&\quad = \operatorname{trace}(\mathbf{I}_{m\times m} - n\theta(\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X} + n\theta\boldsymbol{\Omega})^{-1}\boldsymbol{\Omega}) \\
&\quad = m - n\theta\operatorname{trace}(\mathbf{L}^\intercal(\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X} + n\theta\boldsymbol{\Omega})^{-1}\mathbf{L}) \\
&\quad = m - n\theta\operatorname{trace}((\mathbf{L}^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X}\mathbf{L}^{-\intercal} + n\theta\mathbf{I})^{-1}).
\end{aligned} \tag{29}$$

In the above, $\mathbf{I}_{m\times m}$ is identity matrix of size m. To compute the above term, we interpolate

$$\operatorname{trace}\left((\mathbf{A} + t\mathbf{I})^{-1}\right) = m(\tau_{-1}(t))^{-1}, \tag{30}$$

where $t := n\theta - s$ and

$$\mathbf{A} := \mathbf{L}^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X}\mathbf{L}^{-\intercal} + s\mathbf{I}.$$

We note that the size of $\mathbf{A}$ and $\mathbf{I}$ is m. Also, $\mathbf{A}$ is symmetric and positive-definite since it can be written as a Gramian matrix. The purpose of the fixed parameter $s \ll 1$ is to slightly shift the singular matrix $\mathbf{L}^{-1}\mathbf{X}^\intercal\mathbf{K}^{-1}\mathbf{X}\mathbf{L}^{-\intercal}$ to make $\mathbf{A}$ non-singular. The shift is necessary since without it, (30) is undefined at $t = 0$, and we cannot compute $\tau_{-1,0} = m/\operatorname{trace}(\mathbf{A}^{-1})$. Also, the shift can improve interpolation by relocating the origin of t to the vicinity of the interval where we are interested to compute $V(\theta)$.

For simplicity in our numerical experiment, we set $\mathbf{K}$ and $\boldsymbol{\Omega}$ to identity matrices of sizes n and m, respectively. We also set $s = 10^{-3}$. We create an ill-conditioned design matrix $\mathbf{X}$ for our numerical example using singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\intercal$. The orthogonal matrices $\mathbf{U} \in \mathcal{M}_{n,n}(\mathbb{R})$ and $\mathbf{V} \in \mathcal{M}_{m,m}(\mathbb{R})$ were produced by the Householder matrices

$$\mathbf{U} := \mathbf{I} - 2\frac{\boldsymbol{u}\boldsymbol{u}^\intercal}{\|\boldsymbol{u}\|_2^2}, \quad \text{and} \quad \mathbf{V} := \mathbf{I} - 2\frac{\boldsymbol{v}\boldsymbol{v}^\intercal}{\|\boldsymbol{v}\|_2^2},$$

[5] The function (28) is modified to incorporate the correlation of error using $\mathbf{K}$, and can be derived from the conventional definition of generalized cross-validation function for the decorrelated error $\boldsymbol{\epsilon}' := \mathbf{L}^{-1}\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ where $\mathbf{K} = \mathbf{L}\mathbf{L}^\intercal$ is the Cholesky decomposition of $\mathbf{K}$.

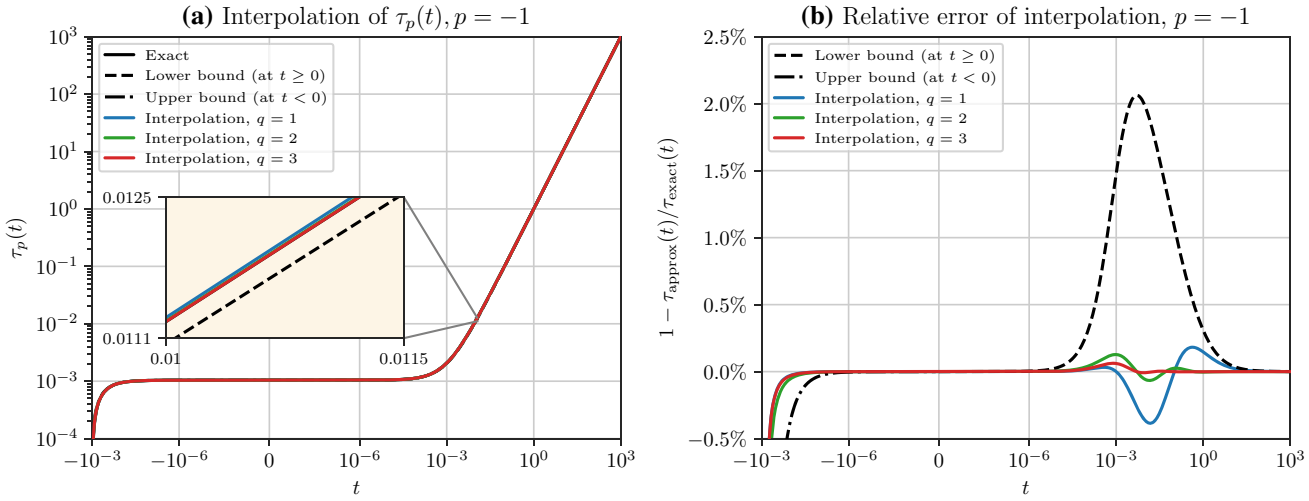

Fig. 5 **a** The exact solution $\tau_{-1}(t)$, bounds $\hat{\tau}_{-1}(t)$, and Padé rational polynomial interpolations $\tilde{\tau}_{-1}(t)$ for $q = 1, 2, 3$ are shown. The green curve and the exact solution in the solid black curve are overlaid by the red curve. The embedded diagram (with linear axes) magnifies a portion of the curves with the highest interpolation error. **b** The relative error of the curves in **a** with respect to the exact solution is shown. In both diagrams (a) and **b**, the horizontal axis in the interval $[-10^{-6}, 10^{-6}]$ is linear, but outside this interval, the axis is logarithmic

where $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^m$ are random vectors (see also Golub and von Matt 1997, Sect. 10). The diagonal matrix $\boldsymbol{\Sigma} \in \mathcal{M}_{n,m}(\mathbb{R})$ was defined by

$$\Sigma_{ii} := \exp\left(-40\left(\frac{i-1}{m}\right)^{3/4}\right), \quad i = 1, \ldots, m. \tag{31}$$

We set $n = 10^3$ and $m = 500$. We generated data by letting $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}$, and $\boldsymbol{\epsilon}$, were randomly generated with a unit variance, and $\sigma = 0.4$, respectively.

We computed the exact solution of $\tau_{-1}(t)$ in (30), and interpolation points $\tau_{-1}(t_i)$, using the Cholesky factorization method described by (24b). The exact solution is shown by the solid black curve in Fig. 5 (overlaid by the red curve) with $\tau_{-1,0} = 960.5^{-1}$. The lower bound $\hat{\tau}_{-1}(t)$ from (7a) is shown by the dashed black curve for $t > 0$. In contrast, at $t \in (t_{\text{inf}}, 0]$, the upper bound from (7b) is shown, where $t_{\text{inf}} = -\min(\lambda(\mathbf{A}))$ and $\min(\lambda(\mathbf{A})) \approx s = 10^{-3}$ is the smallest eigenvalue of $\mathbf{A}$. The relative error of the bounds with respect to the exact solution are shown in Fig. 5b. The peak of the absolute error of the lower bound is located approximately at $t \approx \tau_{-1,0} \approx 10^{-3}$, and the peak of its relative error of the lower bound is slightly to the right of this value.

We sought to interpolate (30) in the interval $\theta \in [10^{-7}, 10]$. Accordingly, since we set $s = 10^{-3}$, we shifted the origin of $t = n\theta - s$ inside the interval $n\theta \in [10^{-4}, 10^4]$. Thus, we approximately had $-10^{-3} < t < 10^4$. Because this interval contains the origin, we employed the Padé rational polynomial interpolation method in Sect. 3.2. (Recall that at small t, particularly at $|t| \ll \tau_{-1,0}$, the rational polynomial interpolation performs better than the basis functions interpolation.) We distributed the interpolant points at $t_i \geq \tau_{-1,0} \approx 10^{-3}$ where the rational polynomial interpolation has to adhere to the exact solution.

The interpolation function $\tilde{\tau}_{-1}(t)$ with $q = 1, 2, 3$ is shown in Fig. 5a using $2q$ interpolation points t_i in the interval $t_i \in [5 \times 10^{-3}, 5]$ that are equally distanced in the logarithmic scale. The red curve corresponding to $q = 3$ and the black curve (exact solution) are indistinguishable even in the embedded diagram that magnifies the location with the highest error. The relative error of the interpolations is shown by Fig. 5b. On the far left of the range of t, the error spikes due to the singularity of the matrix $\mathbf{X}$, which makes $\tau_{-1}(t)$ undefined at $t = -10^{-3}$, corresponding to $\theta = 0$. On the rest of the range, the green and red curves respectively show less than 0.1% and 0.05% relative errors, which are compelling accuracy for a broad range of t, and achieved with only four and six interpolation points, respectively.

4.3.2 Optimization of generalized cross-validation

Here we apply the result of our trace interpolations above to solve the generalized cross-validation problem. The function $V(\theta)$ from (28) is plotted in Fig. 6, with the black curve, corresponding to the exact solution with $\tau_{-1}(t)$ applied in the denominator of $V(\theta)$, serving as a benchmark for comparison. The blue, green, and red curves correspond to the proceeding trace interpolations applied in the denominator of $V(\theta)$. The interpolated curves exhibit both local minima of $V(\theta)$ similar to the benchmark curve, but with slight differences in the positions of the minima. Due to the singularity at $\theta = 0$, the interpolations of $\tau_{-1}(t)$ become less accurate

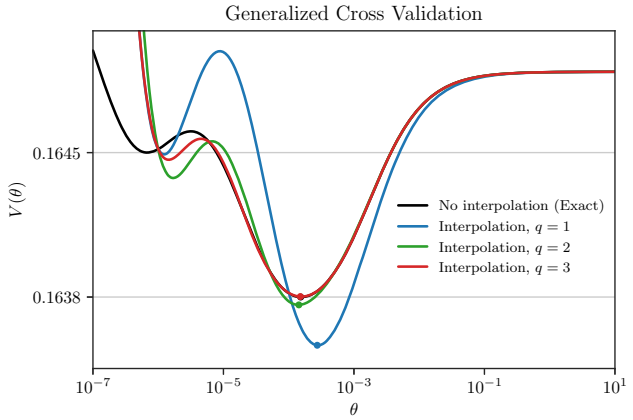

Fig. 6 The generalized cross-validation function is shown, where the black and colored curves correspond respectively to the exact and interpolated computation of $\tau_{-1}(t)$ in the denominator of $V(\theta)$. The global minimum of each curve is shown by a dot

at low values of θ. At higher values of θ, all curves steadily asymptote to a constant. We note that the results in Fig. 6 are compelling since the estimation of $V(\theta)$ is sensitive to the interpolation of its denominator. Namely, a consistent interpolation accuracy over all the parameter range is essential to capture the qualitative shape of $V(\theta)$ correctly.

The global minimum of $V(\theta)$ at $\theta = \theta^*$ is the optimal compromise between an ill-conditioned regression problem (small θ) and a highly regularized regression problem (large θ). We aimed to test the practicality of our interpolation method in searching the global minimum, $V(\theta^*)$, by numerical optimization. We note that we constructed $\mathbf{X}$ in (31) so that $V(\theta)$ would have two local minima thus making optimization less trivial. In general, the generalized cross-validation function may have more than one local minimum necessitating global search algorithms (Kent and Mohammadzadeh 2000). The optimization was performed using a differential evolution optimization method (Storn and Price 1997) with a best/1/exp strategy and 40 initial guess points. The results are shown in the first four rows of Table 2, where the trace of a matrix inverse is computed by the Cholesky factorization described in (24b). In the first row, τ_{-1} is computed exactly, i.e., without interpolation, at all requested locations t during the optimization procedure. On the second to fourth rows, τ_{-1} is first pre-computed at the interpolation points, t_i, by the Cholesky factorization, and then the interpolation is subsequently used during the optimization procedure.

In the table, N_{tr} counts the number of exact evaluations of τ_{-1}. For the Padé rational polynomial interpolation method of degree q, we had $N_{\text{tr}} = 2q + 1$, accounting for $2q$ interpolant points in addition to the evaluation of $\tau_{-1,0}$ at $t = 0$. N_{tot} is the total number of estimations of τ_{-1} during the optimization process. In the first row, $N_{\text{tr}} = N_{\text{tot}}$ as all points are evaluated exactly, i.e., without interpolation. However, for the interpolation methods, N_{tot} consists of N_{tr} plus the evaluations of τ_{-1} via interpolation.

The exact computations of τ_{-1} (at N_{tr} points) are the most computationally expensive part of the overall process. Our numerical experiments were performed on the Intel Xeon E5-2640 v4 processor using shared memory parallelism. We measured computational costs by the total CPU processing time of all computing cores. T_{tr} denotes the processing time of computing τ_{-1} exactly at the N_{tr} points. T_{tot} measures the processing time of the overall optimization, which includes T_{tr}. As shown, the interpolation methods took significantly less processing time compared to no interpolation, namely, by two orders of magnitude for T_{tr}, and an order of magnitude for T_{tot}. We also observe that without interpolation, T_{tr} is the

Table 2 Comparison of methods to optimize the regularization parameter θ, with and without interpolation of $\tau_{-1}(t)$, and by various algorithms of computing trace of a matrix inverse

Computing $\tau_{-1}(t)$		Iterations		Time (s)		Results		Relative error		
Algorithm	Interpolation method	N_{tr}	N_{tot}	T_{tr}	T_{tot}	$V(\theta^*)$	$\log_{10}\theta^*$	$\frac{\lvert\Delta\log_{10}\theta^*\rvert}{\lvert\log_{10}\theta^*\rvert}$ (%)	$\frac{\lVert\Delta\hat{\boldsymbol{\beta}}\rVert_2}{\lVert\hat{\boldsymbol{\beta}}\rVert_2}$ (%)	$\frac{\lVert\Delta\hat{\mathbf{y}}\rVert_2}{\lVert\hat{\mathbf{y}}\rVert_2}$ (%)
Cholesky	No interpolation (exact)	282	282	27.49	30.90	0.16376	−3.8164	0.00	0.00	0.00
	Rational polynomial, $q = 1$	3	364	0.29	4.70	0.16352	−3.5628	6.65	29.71	17.59
	Rational polynomial, $q = 2$	5	282	0.49	3.93	0.16372	−3.8446	0.74	3.69	1.95
	Rational polynomial, $q = 3$	7	284	0.69	4.12	0.16376	−3.8218	0.14	0.71	0.37
Hutchinson	No interpolation	312	312	61.33	65.14	0.16372	−3.7939	0.59	2.98	1.58
	Rational polynomial, $q = 1$	3	364	0.57	4.96	0.16348	−3.5649	6.59	29.49	17.45
	Rational polynomial, $q = 2$	5	282	0.88	4.29	0.16371	−3.8274	0.29	1.45	0.76
	Rational polynomial, $q = 3$	7	282	1.25	4.66	0.16374	−3.8119	0.12	0.61	0.32
SLQ	No interpolation	322	322	66.67	88.75	0.16373	−3.7939	0.59	2.98	1.58
	Rational polynomial, $q = 1$	3	364	0.58	5.04	0.16352	−3.5597	6.73	30.03	17.81
	Rational polynomial, $q = 2$	5	282	1.03	4.52	0.16376	−3.8778	1.61	7.88	4.18
	Rational polynomial, $q = 3$	7	282	0.70	4.13	0.16378	−3.7770	1.03	5.17	2.76

dominant part of the total processing time, T_{tot}. In contrast, with interpolation, T_{tr} becomes so small that T_{tot} is dominated by the cost of evaluating the numerator of $V(\theta)$ in (28), which is proportional to N_{tot}.

The results of computing the optimized parameter, θ^*, and the corresponding minimum, $V(\theta^*)$, are shown in the seventh and eighth columns of Table 2, respectively. The ninth column is the relative error of estimating θ^*, and obtained by comparing $\log_{10}\theta^*$ between the interpolated and benchmark solution (i.e., first row). The last two columns are the ℓ^2 norm of the error of $\hat{\boldsymbol{\beta}}$ [using (26)] and $\hat{\boldsymbol{y}}$ compared to their exact solution, and their relative error are obtained by normalizing with the ℓ^2 norm of their exact solution. We observed, for example for $q = 2$, that with one-tenth of the processing time, an accuracy of 2% error for $\hat{\boldsymbol{y}}$ is achieved, which is generally sufficient in practical applications. Also, for $q = 3$, the error reduces to <1% with similar processing time. In general, the error can be improved simply by using more interpolant points. We have found that simple heuristics for setting defaults for q are broadly effective. Namely, if θ^* is expected to be found in an known interval, one can use a small number of interpolating points ($q = 1 \sim 2$) on the boundary or center of the interval. If there is no prior knowledge of the range, one can let an optimization scheme search for θ^* in a wide logarithmic range, e.g., $[10^{-7}, 10^{+1}]$ with $q = 3 \sim 4$.

4.3.3 Testing alternative trace estimators

Besides the Cholesky factorization algorithm, we also repeated the numerical experiments above with stochastic trace estimators, namely, the Hutchinson's algorithm (Hutchinson 1990) and stochastic Lanczos quadrature algorithm (Golub and Meurant 2009, Sect. 11.6.1) to compute the trace of a matrix inverse. These class of randomized methods are attractive due to their scalability to very large matrices, where employing the exact methods could be inefficient, if not infeasible. However, these methods do not compute the exact value of determinant or trace, rather they converge to a solution by Monte-Carlo sampling through iterations. The complexity of Hutchinson method using conjugate gradient is $\mathcal{O}(\rho n^2 s)$ where ρ is the density of matrix ($\rho = 1$ for dense matrices) and s is the number of random vectors for Monte-Carlo sampling. We recall that in our application, the cost of the interpolation scheme is q times the above-mentioned complexity, i.e.,

$$\mathcal{O}\left(q\rho n^2 s\right).$$

Alternatively, the computational cost of the SLQ method is $\mathcal{O}((\rho n^2 + nl)sl)$ where l is the Lanczos degree, which is the number of Lanczos tri-diagonalization iterations (see details e.g., in Ubaru et al. 2017, Sect. 3). Thus, the complexity of the interpolation method becomes

$$\mathcal{O}\left(q(\rho n^2 + nl)sl\right).$$

In both these algorithms, we employed $s = 30$ random vectors with Rademacher distribution for Monte-Carlo sampling. Also, in SLQ algorithm, we set the Lanczos degree to 30.

The results for Hutchinson's algorithm are shown in the fifth to eighth rows, and results for the SLQ algorithm are shown in the ninth to twelfth rows, of Table 2. Similar to the Cholesky factorization, in both stochastic estimators, the interpolation technique reduces the processing times compared to no interpolation, namely, T_{tr} is reduced by two orders of magnitude, and T_{tot} by an order of magnitude, while maintaining a reasonable accuracy.

We note that the interpolation with the stochastic methods introduces error due to the uncertainly in the randomized estimation of τ_{-1} at the interpolant points t_i. However, the additional error caused by interpolation itself can be less than the error due to aforementioned stochastic estimation. For instance, without interpolation, the SLQ method estimates $\hat{\boldsymbol{y}}$ with a 1.58% error, whereas, interpolation with $q = 3$ results in a similar error of 2.76% but at a greater than 20-fold reduction in computational cost.

5 Further applications

We recall that the presented interpolation scheme can be applied to any formulation that consists of the trace or determinant of a power of the one-parameter affine matrix function $\mathbf{A} + t\mathbf{B}$ where $\mathbf{A}$ and $\mathbf{B}$ are Hermitian and positive-definite. Often in applications, an algebraic trick [such as in (29)] is required to form such an affine matrix function. We here provide two other closely related examples where such affine matrix function can be formulated.

5.1 Reproducing kernel Hilbert space

Let $\mathcal{H}_K$ be a reproducing kernel Hilbert space equipped with the reproducing kernel K that defines the function evaluation $f(\boldsymbol{x}) = \langle f, K(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}_K}$. Consider an infinite-dimensional generalized ridge regression on $\mathcal{H}_K$ to estimate $y = f(\boldsymbol{x})$ with the given training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ by the minimization problem (Hastie et al. 2001, Sect. 5.8.2)

$$\min_{f\in\mathcal{H}_K} \sum_{i=1}^{n} |y_i - f(\boldsymbol{x}_i)|^2 + \theta \|f\|^2_{\mathcal{H}_K}.$$

The solution to the above problem has the form $f(\cdot) = \sum_j \alpha_j K(\cdot, \boldsymbol{x}_j)$. For the finite-dimensional formulation, define the kernel matrix $\mathbf{K}$ with the components $K_{ij} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which is symmetric and positive-definite. Let $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_n]^\intercal$ and $\boldsymbol{y} := [y_1, \ldots, y_n]^\intercal$. The minimization problem in finite-dimensional setting becomes

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \theta \|\boldsymbol{\alpha}\|_{\mathbf{K}}^2,$$

where $\|\boldsymbol{\alpha}\|_{\mathbf{K}}^2 = \boldsymbol{\alpha}^\intercal \mathbf{K} \boldsymbol{\alpha}$. The optimal solution to the above problem is

$$\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \theta \mathbf{I})^{-1} \boldsymbol{y},$$

and the fitted values on the training points are $\hat{\boldsymbol{y}} = \mathbf{K}\hat{\boldsymbol{\alpha}} =: \mathbf{S}_\theta \boldsymbol{y}$ where the smoother matrix $\mathbf{S}_\theta$ is defined by $\mathbf{S}_\theta := \mathbf{K}(\mathbf{K} + \theta \mathbf{I})^{-1}$.

One may seek the optimal value for θ as the minimizer of the GCV function

$$V(\theta) := \frac{\frac{1}{n} \|(\mathbf{I} - \mathbf{S}_\theta)\boldsymbol{y}\|_2^2}{\left(\frac{1}{n} \operatorname{trace}(\mathbf{I} - \mathbf{S}_\theta)\right)^2}. \tag{32}$$

We recall that the expensive part of computing (32) is the term $\operatorname{trace}(\mathbf{S}_\theta)$. To apply our interpolation scheme, write $\mathbf{S}_\theta$ as the *Reinsch* form

$$\mathbf{S}_\theta = \theta^{-1}(\mathbf{K}^{-1} + \theta^{-1}\mathbf{I})^{-1}.$$

We realize that

$$\operatorname{trace}(\mathbf{S}_\theta) = tn(\tau_{-1}(t))^{-1},$$

where $t := \theta^{-1}$. The proposed interpolation method follows by using $\mathbf{A} = \mathbf{K}^{-1}$, $\mathbf{B} = \mathbf{I}$, and $\tau_{-1,0} = n/\operatorname{trace}(\mathbf{K})$.

5.2 Kernel-based GCV for mixed models

Another formulation of kernel-based GCV, for instance by Xu and Zhu (2009, Eqs. 9 and 10), yields a function of the form

$$V(h, \phi) = \frac{\frac{1}{n} \|(\mathbf{I} - \mathbf{H}(h, \phi))\boldsymbol{y}\|_2^2}{\left(\frac{1}{n} \operatorname{trace}(\mathbf{I} - \mathbf{H}(h, \phi))\right)^2}, \tag{33}$$

where

$$\begin{aligned}\mathbf{H}(h, \phi) = &\\ &\tilde{\mathbf{H}} + (\mathbf{I} - \tilde{\mathbf{H}})\mathbf{Z}\left(\mathbf{Z}^\intercal(\mathbf{I} - \tilde{\mathbf{H}})^\intercal(\mathbf{I} - \tilde{\mathbf{H}})\mathbf{Z} + \boldsymbol{\Sigma}\right)^{-1} \\ &\mathbf{Z}^\intercal(\mathbf{I} - \tilde{\mathbf{H}})^\intercal(\mathbf{I} - \tilde{\mathbf{H}}).\end{aligned} \tag{34}$$

In the above, the covariance $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\phi)$ is symmetric and positive-definite, the design matrix of random effects $\mathbf{Z}$ has full column-rank, and $\tilde{\mathbf{H}} = \tilde{\mathbf{H}}(h)$ is the smoother matrix when the random effects are absent. Optimal values of the parameters (h, ϕ) are sought by minimizing V.

It is possible to represent the term in the denominator of (33) by the trace of the inverse of a single matrix to be written as τ_{-1}. To do so, let $\mathbf{P} := \mathbf{I} - \tilde{\mathbf{H}}$ and $\mathbf{Y} := \mathbf{PZ}$. Using the Woodbury matrix identity and (34), we can represent the term inside the trace in (33) as

$$\begin{aligned}\mathbf{I} - \mathbf{H}(h, \phi) &= \left(\mathbf{I} - \mathbf{Y}(\mathbf{Y}^\intercal\mathbf{Y} + \boldsymbol{\Sigma})^{-1}\mathbf{Y}^\intercal\right)\mathbf{P} \\ &= \left(\mathbf{I} + \mathbf{Y}\boldsymbol{\Sigma}^{-1}\mathbf{Y}^\intercal\right)^{-1}\mathbf{P}.\end{aligned}$$

If $\mathbf{P}$ is positive-definite, then let $\mathbf{P} = \mathbf{L}\mathbf{L}^\intercal$ be the Cholesky decomposition of $\mathbf{P}$. By using the cyclic property of trace operator, we have

$$\begin{aligned}&\operatorname{trace}(\mathbf{I} - \mathbf{H}(h, \phi)) \\ &\quad = \operatorname{trace}\left(\mathbf{L}^\intercal(\mathbf{I} + \mathbf{Y}\boldsymbol{\Sigma}^{-1}\mathbf{Y}^\intercal)^{-1}\mathbf{L}\right) \\ &\quad = \operatorname{trace}\left((\mathbf{L}^{-1}\mathbf{L}^{-\intercal} + \mathbf{L}^{-1}\mathbf{Y}\boldsymbol{\Sigma}^{-1}\mathbf{Y}^\intercal\mathbf{L}^{-\intercal})^{-1}\right).\end{aligned} \tag{35}$$

Note that both matrices $\mathbf{A} := \mathbf{L}^{-1}\mathbf{L}^{-\intercal}$ and $\mathbf{B} := \mathbf{L}^{-1}\mathbf{Y}\boldsymbol{\Sigma}^{-1}\mathbf{Y}^\intercal\mathbf{L}^{-\intercal}$ are symmetric and positive-definite since they are in the Gramian form. To compute (35), the presented interpolation method can be applied for instance if $\boldsymbol{\Sigma}(\phi)$ is linear in its parameter. Such assumption is common, for instance when $\boldsymbol{\Sigma} = \phi\mathbf{K}$ where ϕ is variance and $\mathbf{K}$ is the correlation matrix. In such a case, the sum of two matrices in (35) becomes an affine function of $t := \phi^{-1}$ and $\operatorname{trace}(\mathbf{I} - \tilde{\mathbf{H}}(h, \phi))$ can be written as $\tau_{-1}(t)$.

6 Conclusions

In many applications in statistics and machine learning, it is desirable to estimate the determinant and trace of the real powers of a one-parameter family of matrix functions $\mathbf{A} + t\mathbf{B}$ where the parameter t varies and the matrices $\mathbf{A}$ and $\mathbf{B}$ in the formulation remain unchanged. There exist many efficient techniques to estimate the determinant and trace of implicit matrices (such as inverse of a matrix), however, these methods are geared toward generic matrices. Using those methods, the computation of the determinant and trace of the parametric matrices should be repeated for each parameter value as the matrix is updated. To efficiently perform such computation for a wide range of parameter t, we presented in this work heuristic methods to interpolate the functions $t \mapsto \log\det(\mathbf{A} + t\mathbf{B})$ and $t \mapsto \operatorname{trace}((\mathbf{A} + t\mathbf{B})^p)$. The interpolation approach is based on sharp bounds for these functions using inequalities for a Schatten-type norm and

anti-norm. We proposed two types of interpolation functions, namely, interpolation with a linear combination of orthogonalized inverse-monimial basis functions, and interpolation with rational polynomials, which includes Padé approximation and Chebushev rational functions. We demonstrated that both functions can provide highly accurate interpolation over a wide range of t using very few interpolation points. The rational polynomials generally provide better results in the neighborhood of the origin of the parameter. In the regions away from the origin, choice of interpolation method is less important; namely we observed e.g., the interpolation with Chebyshev rational functions provide similar results to the orthogonalized inverse-monomials in (13) in such cases. All the presented interpolation methods can lead to one to two orders of magnitude savings in processing time in practical applications that require frequent evaluations of $\log\det(\mathbf{A}+t\mathbf{B})$ or $\mathrm{trace}((\mathbf{A}+t\mathbf{B})^p)$.

For applications where one is interested in values of $t \ll \tau_{p,0}$ (such as in Sect. 4.3 where the matrix was shifted due to being ill-conditioned) interpolation using (18) is recommended. One should keep in mind that there exists the possibility that (18) can become singular at its poles, but a slight rearrangement of the interpolant points t_i can be used to ensure these poles are outside the domain of interest. Although (18) provides accurate interpolation for a broad range of t, for a higher number of interpolation points (e.g 6 or more), relation (13) or (20) is preferred.

In closing, the presented interpolation method can be effectively utilized on large data, particularly with the powerful framework of randomized estimators of trace and log-determinant. A practical application of this method together with stochastic Lanczos quadrature on sparse matrices is given by Ameli and Shadden (2022c) to efficiently train Gaussian process regression. The interested reader may refer to Ameli and Shadden (2022b) where the interpolation scheme can be applied to massive data (e.g., $n \sim 2^{25}$) using the `imate` package.

Acknowledgements The authors acknowledge support from the National Science Foundation, Award No. 1520825, and American Heart Association, Award No. 18EIA33900046.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

Appendix A: Proofs

In Theorem 1, we show (7a) and (7b) for the operator (3) and $p \in (-\infty, 1) \setminus \{0\}$. The results for $p = 0$ follows by the continuity condition in (4).

Theorem 1 *Suppose $p \in (-\infty, 1) \setminus \{0\}$ and let the matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{n,n}(\mathbb{C})$ be Hermitian and positive semi-definite (positive definite if $p < 0$). Then*

$$\|\mathbf{A}+\mathbf{B}\|_p \geq \|\mathbf{A}\|_p + \|\mathbf{B}\|_p, \tag{A.1a}$$

$$\|\mathbf{A}-\mathbf{B}\|_p \leq \|\mathbf{A}\|_p - \|\mathbf{B}\|_p, \tag{A.1b}$$

provided that $\mathbf{A} \succeq \mathbf{B}$ ($\mathbf{A} \succ \mathbf{B}$ if $p < 0$) for (A.1b) to hold. In both (A.1a) and (A.1b), the equality is achieved if and only if $\mathbf{A} = c\mathbf{B}$ for $c \in \mathbb{R}_{\geq 0}$ ($c \in \mathbb{R}_{>0}$ if $p < 0$).

We prove Theorem 1 as follows.

Definition 1 (*Majorization*) For the n-tuple $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, we denote by $\boldsymbol{x}^{\downarrow} = (x_1^{\downarrow}, \ldots, x_n^{\downarrow})$ the tuple with the same components as $\boldsymbol{x}$, but sorted in decreasing order. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$. We say $\boldsymbol{x}$ weakly majorizes $\boldsymbol{y}$ from below (sub-majorizes) and indicate by $\boldsymbol{x} \succ_{\mathrm{w}} \boldsymbol{y}$ if and only if

$$\sum_{i=1}^{k} x_i^{\downarrow} \geq \sum_{i=1}^{k} y_i^{\downarrow}, \quad \text{for all } k = 1, \ldots, n.$$

Furthermore, if $\boldsymbol{x} \succ_{\mathrm{w}} \boldsymbol{y}$ and $\sum_{i=1}^{n} x_i^{\downarrow} = \sum_{i=1}^{n} y_i^{\downarrow}$, we say $\boldsymbol{x}$ majorizes $\boldsymbol{y}$ and indicate by $\boldsymbol{x} \succ \boldsymbol{y}$.

Proposition 2 *Suppose $p \in (-\infty, 1) \setminus \{0\}$, and let the matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{n,n}(\mathbb{C})$ be Hermitian and positive semi-definite (positive-definite if $p < 0$) with the n-tuple of eigenvalues $\lambda(\mathbf{A})$ and $\lambda(\mathbf{B})$, respectively. Then*

$$\|\mathbf{A}+\mathbf{B}\|_p \geq M_p(\lambda^{\downarrow}(\mathbf{A}) + \lambda^{\downarrow}(\mathbf{B})), \tag{A.2}$$

where M_p is the generalized mean defined in (5). The equality in the above holds if and only if $\lambda^{\downarrow}(\mathbf{A}+\mathbf{B}) = \lambda^{\downarrow}(\mathbf{A}) + \lambda^{\downarrow}(\mathbf{B})$.

Proof We proceed the proof for $p < 0$ as the case $p \in (0, 1)$ follows similarly. By Ky Fan eigenvalue inequality for Hermitian matrices (Zhang 2011, p. 356, Theorems 10.21)

$$\lambda(\mathbf{A}+\mathbf{B}) \prec \lambda^{\downarrow}(\mathbf{A}) + \lambda^{\downarrow}(\mathbf{B}). \tag{A.3}$$

Let $\mathcal{I} := [\lambda_{\min}, \lambda_{\max}]$, $\lambda_{\min} := \min\{\lambda_n^{\downarrow}(\mathbf{A}), \lambda_n^{\downarrow}(\mathbf{B})\}$ and $\lambda_{\max} := \max\{\lambda_1^{\downarrow}(\mathbf{A}), \lambda_1^{\downarrow}(\mathbf{B})\}$. Since $\mathbf{A}$ and $\mathbf{B}$ are Hermitian and at least positive semi-definite, we have $\mathcal{I} \subset \mathbb{R}_{\geq 0}$.

Define the convex function $f(t) := t^p$ on $\mathcal{I}$. Applying (A.3) to Theorem 5.A.1 of Marshall et al. (2011, p. 165) yields

$$f\left(\lambda(\mathbf{A}+\mathbf{B})\right) \prec_{\mathrm{w}} f\left(\lambda^{\downarrow}(\mathbf{A})+\lambda^{\downarrow}(\mathbf{B})\right). \tag{A.4}$$

(We note that for $p \in (0,1)$, the function f is concave and the direction of the above and subsequent inequalities are flipped instead). The above relation in particular, implies

$$\sum_{i=1}^{n} f\left(\lambda_i(\mathbf{A}+\mathbf{B})\right) \le \sum_{i=1}^{n} f\left(\lambda_i^{\downarrow}(\mathbf{A})+\lambda_i^{\downarrow}(\mathbf{B})\right). \tag{A.5}$$

Raising (A.5) to the power $\frac{1}{p}$ (which flips the direction of inequality if $p<0$) concludes (A.2).

Also, the condition $\lambda(\mathbf{A}+\mathbf{B}) = \lambda^{\downarrow}(\mathbf{A})+\lambda^{\downarrow}(\mathbf{B})$ is sufficient for the equality in (A.5). To show the necessity condition, suppose in contrary that the equality in (A.5) holds. This equality condition together with (A.4) imply

$$f\left(\lambda(\mathbf{A}+\mathbf{B})\right) \prec f\left(\lambda^{\downarrow}(\mathbf{A})+\lambda^{\downarrow}(\mathbf{B})\right). \tag{A.6}$$

The above condition can be achieved if and only if either f is linear (Marshall et al. 2011, p. 166, Theorem 5.A.1.e), which is not, or if $\lambda(\mathbf{A}+\mathbf{B}) = \lambda^{\downarrow}(\mathbf{A})+\lambda^{\downarrow}(\mathbf{B})$. □

The equality condition in Proposition 2 is contingent on the following condition.

Lemma 3 *Let $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{n,n}(\mathbb{C})$ be Hermitian matrices with the n-tuple of eigenvalues $\lambda(\mathbf{A})$ and $\lambda(\mathbf{B})$, respectively. Then, $\lambda^{\downarrow}(\mathbf{A}+\mathbf{B}) = \lambda^{\downarrow}(\mathbf{A})+\lambda^{\downarrow}(\mathbf{B})$ only if $\mathbf{A}$ and $\mathbf{B}$ commute.*

Proof Using the Golden–Thompson inequality (Bhatia 1997, p. 261, Eq. IX.19) and Von Neumann's trace inequality (Mirsky 1975) respectively, we have

$$\begin{aligned}\mathrm{trace}\left(e^{\mathbf{A}+\mathbf{B}}\right) &\le \mathrm{trace}\left(e^{\mathbf{A}}e^{\mathbf{B}}\right) \le \sum_{i=1}^{n} e^{\lambda_i^{\downarrow}(\mathbf{A})}e^{\lambda_i^{\downarrow}(\mathbf{B})} \\ &= \sum_{i=1}^{n} e^{\lambda_i^{\downarrow}(\mathbf{A})+\lambda_i^{\downarrow}(\mathbf{B})} = \mathrm{trace}\left(e^{\mathbf{A}+\mathbf{B}}\right).\end{aligned}$$

But, the equality in Golden–Thompson inequality holds if and only if $\mathbf{A}$ and $\mathbf{B}$ commute (Petz 1994). □

Remark 5 The equality (2b) is a special case of (A.2) since $\mathbf{A}$ commutes with $\mathbf{B} = t\mathbf{I}$. This also applies to (2a) as it can be obtained from (2b) at $p \to 0$.

We also show superadditivity of the generalized mean function, M_p, for $p<1$.

Lemma 4 *M_p for $p<1$ is a concave function on $\mathbb{R}^n_{\ge 0}$ ($\mathbb{R}^n_{>0}$ if $p<0$).*

Proof We show the Hessian $\mathbf{H}$ of the function $M_p(\boldsymbol{x})$ is negative semi-definite. The component H_{ij} of the Hessian matrix $\mathbf{H}$ is

$$\begin{aligned}H_{ij} &:= \frac{\partial^2 M_p}{\partial x_i \partial x_j} \\ &= \frac{p-1}{n^{\frac{1}{p}}}\left(\sum_{k=1}^{n} x_k^p\right)^{\frac{1}{p}-2} x_i^{p-1}\left(\frac{\delta_{ij}}{x_i}\sum_{k=1}^{n} x_k^p - x_j^{p-1}\right),\end{aligned}$$

where δ_{ij} is the Kronecker delta function. The matrix $\mathbf{H}$ is negative semi-definite if and only if $\boldsymbol{w}^{\intercal}\mathbf{H}\boldsymbol{w} \le 0$ for all nonzero vectors $\boldsymbol{w} := (w_1, \ldots, w_n)$. The latter condition is equivalent to

$$\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j x_i^{p-1}\left(\frac{\delta_{ij}}{x_i}\sum_{k=1}^{n} x_k^p - x_j^{p-1}\right) \ge 0,$$

which simplifies to

$$\left(\sum_{j=1}^{n} w_j x_j^{p-1}\right)^2 \le \left(\sum_{k=1}^{n} x_k^p\right)\left(\sum_{i=1}^{n} w_i^2 x_i^{p-2}\right).$$

The above relation holds by the Cauchy–Schwarz inequality for the product two vectors with the components $x_j^{\frac{p}{2}}$ and $w_j x_j^{\frac{p}{2}-1}$. Thus, $\mathbf{H}$ is negative semi-definite and it concludes the proof. □

Proposition 5 *M_p for $p<1$ is superadditive on $\mathbb{R}^n_{\ge 0}$ ($\mathbb{R}^n_{>0}$ if $p<0$). That is, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n_{>0}$,*

$$M_p(\boldsymbol{x}+\boldsymbol{y}) \ge M_p(\boldsymbol{x}) + M_p(\boldsymbol{y}). \tag{A.7}$$

The equality in the above holds if and only if $\boldsymbol{x} = c\boldsymbol{y}$ where $c \ge 0$ ($c>0$ if $p<0$).

Proof Since by Lemma 4, the function M_p is concave, from the Jensen inequality (see e.g., Hardy et al. 1952, Sect. 3.12 we have $M_p(\frac{1}{2}(\boldsymbol{x}+\boldsymbol{y})) \ge \frac{1}{2}(M_p(\boldsymbol{x})+M_p(\boldsymbol{y}))$, which concludes (A.7). The Jensen inequality becomes an equality if $\boldsymbol{x} = \boldsymbol{y}$. But, since $M_p(c\boldsymbol{x}) = cM_p(\boldsymbol{x})$, the equality criterion can be extended to $\boldsymbol{x} = c\boldsymbol{y}$. □

We can now prove Theorem 1.

Proof of Theorem 1 We have $M_p(\lambda^{\downarrow}(\mathbf{A})) = \|\mathbf{A}\|_p$ and $M_p(\lambda^{\downarrow}(\mathbf{B})) = \|\mathbf{B}\|_p$. Applying Proposition 5 to $M_p(\lambda^{\downarrow}(\mathbf{A})+\lambda^{\downarrow}(\mathbf{B}))$ and using Proposition 2 concludes (A.1a). Also, applying (A.1a) to $\|\mathbf{B}+(\mathbf{A}-\mathbf{B})\|_p = \|\mathbf{A}\|_p$ concludes (A.1b).

The equality in Proposition 5 holds if and only if $\lambda^{\downarrow}(\mathbf{A}) = c\lambda^{\downarrow}(\mathbf{B})$ for some positive constant c. Also, by Lemma 3,

the equality in Proposition 2 holds if **A** and **B** commute, which means **A** and **B** have the same eigenspace (Horn and Johnson 1990, p. 50, Theorem 1.3.12). By combining these two conditions, equality in (A.1a) and (A.1b) is achieved when **A** is a scalar multiple of **B**. □

References

Ameli, S., Shadden, S.C.: GLearn, a high-performance python package for machine learning using Gaussian process. https://pypi.org/project/glearn/ (2022a)

Ameli, S., Shadden, S.C.: IMATE, a high-performance python package for implicit matrix trace estimation. https://pypi.org/project/imate/ (2022b)

Ameli, S., Shadden, S.C.: Noise estimation in Gaussian process regression. arXiv: 2206.09976 [cs.LG] (2022c)

Ameli, S., Shadden, S.C.: A singular Woodbury and pseudo-determinant matrix identities and application to Gaussian process regression. arXiv: 2207.08038 [math.ST] (2022d)

Avron, H., Toledo, S.: Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. J. ACM (2011). https://doi.org/10.1145/1944345.1944349

Bai, Z., Golub, G.H.: Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices. Ann. Numer. Math. **4**, 29–38 (1997)

Bai, Z., Fahey, G., Golub, G.: Some large-scale matrix computation problems. J. Comput. Appl. Math. **74**(1), 71–89 (1996). https://doi.org/10.1016/0377-0427(96)00018-0

Bekas, C., Curioni, A., Fedulova, I.: Low-cost data uncertainty quantification. Concurr. Comput. Pract. Exp. **24**(8), 908–920 (2012). https://doi.org/10.1002/cpe.1770

Bhatia, R.: Matrix Analysis, vol. 169. Springer, Berlin (1997). https://doi.org/10.1007/978-1-4612-0653-8

Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, Berlin (2006). https://doi.org/10.1117/1.2819119

Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996). https://doi.org/10.1137/1.9781611971484

Bourin, J.C., Hiai, F.: Norm and anti-norm inequalities for positive semi-definite matrices. Int. J. Math. **22**(08), 1121–1138 (2011). https://doi.org/10.1142/S0129167X1100715X

Bourin, J.C., Hiai, F.: Jensen and Minkowski inequalities for operator means and anti-norms. Linear Algebra Appl. **456**, 22–53 (2014). https://doi.org/10.1016/j.laa.2014.05.030. (**special Issue on Matrix Functions**)

Chaloner, K., Verdinelli, I.: Bayesian experimental design: a review. Stat. Sci. **10**(3), 273–304 (1995). https://doi.org/10.1214/ss/1177009939

Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Numer. Math. **31**(4), 377–403 (1978). https://doi.org/10.1007/BF01404567

Davis, T.A.: Direct Methods for Sparse Linear Systems. SIAM, Philadelphia (2006). https://doi.org/10.1137/1.9780898718881

Gibbs, M., MacKay, D.J.C.: Efficient implementation of Gaussian processes. Technical Report. Cavendish Laboratory, Cambridge (1997)

Girard, A.: A fast 'Monte-Carlo cross-validation' procedure for large least squares problems with noisy data. Numer. Math. **56**(1), 1–23 (1989). https://doi.org/10.1007/BF01395775

Golub, G.H., von Matt, U.: Generalized cross-validation for large-scale problems. J. Comput. Graph. Stat. **6**(1), 1–34 (1997). https://doi.org/10.1080/10618600.1997.10474725

Golub, G.H., Meurant, G.: Matrices, Moments and Quadrature with Applications. Princeton University Press, Princeton (2009)

Golub, G.H., Plemmons, R.J.: Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. Linear Algebra Appl. **34**, 3–28 (1980). https://doi.org/10.1016/0024-3795(80)90156-1

Golub, G.H., Strakoš, Z.: Estimates in quadratic formulas. Numer. Algorithms **8**(2), 241–268 (1994). https://doi.org/10.1007/BF02142693

Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979). https://doi.org/10.1080/00401706.1979.10489751

Guo, B., Shen, J., Wang, Z.: Chebyshev rational spectral and pseudospectral methods on a semi-infinite interval. Int. J. Numer. Methods Eng. **53**(1), 65–84 (2002). https://doi.org/10.1002/nme.392

Haber, E., Horesh, L., Tenorio, L.: Numerical methods for experimental design of large-scale linear ill-posed inverse problems. Inverse Probl. **24**(5), 055012 (2008). https://doi.org/10.1088/0266-5611/24/5/055012

Hardy, G.H., Littlewood, J.E., Pólya, G.: Inequalities. Cambridge Mathematical Library. Cambridge University Press, Cambridge (1952)

Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer, New York (2001)

Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1990). https://doi.org/10.1017/CBO9780511810817

Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Commun. Stat. Simul. Comput. **19**(2), 433–450 (1990). https://doi.org/10.1080/03610919008812866

Kalantzis, V., Bekas, C., Curioni, A., et al.: Accelerating data uncertainty quantification by solving linear systems with multiple right-hand sides. Numer. Algorithms **62**(4), 637–653 (2013). https://doi.org/10.1007/s11075-012-9687-2

Kent, J.T., Mohammadzadeh, M.: Global optimization of the generalized cross-validation criterion. Stat. Comput. **10**(3), 231–236 (2000). https://doi.org/10.1023/A:1008939510946

Lukas, M.A., de Hoog, F.R., Anderssen, R.S.: Efficient algorithms for robust generalized cross-validation spline smoothing. J. Comput. Appl. Math. **235**(1), 102–107 (2010). https://doi.org/10.1016/j.cam.2010.05.016

MacKay, D., Kay, D., Press, C.U.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)

Marshall, A.W., Olkin, I., Arnold, B.C.: Inequalities: Theory of Majorization and its Applications, vol. 143, 2nd edn. Springer, Berlin (2011). https://doi.org/10.1007/978-0-387-68276-1

Mirsky, L.: A trace inequality of John von Neumann. Monatsh. Math. **79**, 303–306 (1975). https://doi.org/10.1007/BF01647331

Mitrinović, D.S., Vasić, P.M.: Analytic Inequalities. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete. Springer, Berlin (1970). https://doi.org/10.1007/978-3-642-99970-3

Newbery, A.C.R., Garrett, T.S.: Interpolation with minimized curvature. Comput. Math. Appl. **22**(1), 37–43 (1991). https://doi.org/10.1016/0898-1221(91)90023-W

Niessner, H., Reichert, K.: On computing the inverse of a sparse matrix. Int. J. Numer. Methods Eng. **19**(10), 1513–1526 (1983). https://doi.org/10.1002/nme.1620191009

Petz, D.: A survey of certain trace inequalities. Banach Center Publ. **30**(1), 287–298 (1994). https://doi.org/10.4064/-30-1-287-298

Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, MIT Press, Cambridge (2006)

Saibaba, A.K., Alexanderian, A., Ipsen, I.C.F.: Randomized matrix-free trace and log-determinant estimators. Numer. Math. **137**(2), 353–395 (2017). https://doi.org/10.1007/s00211-017-0880-z

Seber, G., Lee, A.: Linear Regression Analysis. Wiley Series in Probability and Statistics, Wiley, New York (2012). https://doi.org/10.1002/9780471722199

Stathopoulos, A., Laeuchli, J., Orginos, K.: Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. SIAM J. Sci. Comput. **35**(5), S299–S322 (2013). https://doi.org/10.1137/120881452

Stewart, G.W.: Matrix Algorithms: Volume 1: Basic Decompositions. SIAM, Philadelphia (1998). https://doi.org/10.1137/1.9781611971408

Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997). https://doi.org/10.1023/A:1008202821328

Takahashi, K., Fagan, J., Chen, M.S.: Formation of a sparse bus impedance matrix and its application to short circuit study. In: 8th Power Industry Computer Application Conference Proceedings. IEEE Power Engineering Society, pp 63–69 (1973)

Tang, J.M., Saad, Y.: A probing method for computing the diagonal of a matrix inverse. Numer. Linear Algebra Appl. **19**(3), 485–501 (2012). https://doi.org/10.1002/nla.779

Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res. **1**, 211–244 (2001). https://doi.org/10.1162/15324430152748236

Ubaru, S., Saad, Y.: Fast methods for estimating the numerical rank of large matrices. In: Proceedings of the 33rd International Conference on Machine Learning—Volume 48. JMLR.org, ICML'16, pp. 468–477 (2016)

Ubaru, S., Saad, Y.: Applications of trace estimation techniques. In: Kozubek, T., Čermák, M., Tichý, P., et al (eds.) High Performance Computing in Science and Engineering. Springer, Cham, pp 19–33 (2018) https://doi.org/10.1007/978-3-319-97136-0_2

Ubaru, S., Chen, J., Saad, Y.: Fast estimation of tr($f(a)$) via stochastic Lanczos quadrature. SIAM J. Matrix Anal. Appl. **38**(4), 1075–1099 (2017). https://doi.org/10.1137/16M1104974

Wahba, G.: Practical approximate solutions to linear operator equations when the data are noisy. SIAM J. Numer. Anal. **14**(4), 651–667 (1977). https://doi.org/10.1137/0714044

Wu, L., Laeuchli, J., Kalantzis, V., et al.: Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse. J. Comput. Phys. **326**, 828–844 (2016). https://doi.org/10.1016/j.jcp.2016.09.001

Xu, W., Zhu, L.: Kernel-based generalized cross-validation in non-parametric mixed-effect models. Scand. J. Stat. **36**(2), 229–247 (2009). https://doi.org/10.1111/j.1467-9469.2008.00625.x

Zhang, F.: Matrix Theory: Basic Results and Techniques, 2nd edn. Springer, New York (2011). https://doi.org/10.1007/978-1-4614-1099-7