Abstract We use the H-matrix technology to compute the approximate square root of a covariance matrix in linear cost. This allows us to generate normal and log-normal random fields on general point sets with optimal cost. We derive rigorous error estimates which show convergence of the method. Our approach requires only mild assumptions on the covariance function and on the point set. Therefore, it might be also a nice alternative to the circulant embedding approach which applies only to regular grids and stationary covariance functions.

Fast random field generation with H-matrices

Michael Feischl · Frances Y. Kuo · Ian H. Sloan

1 Introduction

Generating samples of random fields is a common bottleneck in simulation and modeling of real life phenomena as, e.g., structural vibrations [6], groundwater flow [8], and composite material behavior [1]. A standard approach is to truncate the Karhunen-Loève expansion of the random field. This can, particularly for rough fields with short correlation length, be very expensive, as many summands of the expansion have to be evaluated to compute a decent approximation. Often, it suffices to evaluate the random field only on some particular (quadrature) nodes. If the random field $\mathcal{Z}(\boldsymbol{x}, \omega)$ is Gaussian with given covariance function $\varrho(\cdot, \cdot)$, it is well-known that the evaluation at the quadrature nodes $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ can be done by computing the square-root of the corresponding covariance matrix $\boldsymbol{C} = (\varrho(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j \in \{1, \ldots, N\}} \in \mathbb{R}^{N \times N}$, i.e.,

$$\mathcal{Z}(\boldsymbol{x}_i, \omega) = \left(\boldsymbol{C}^{1/2} \boldsymbol{z}(\omega)\right)_i \quad \text{for all } i \in \{1, \ldots, N\},$$

where $\boldsymbol{z}(\omega) \in \mathbb{R}^N$ is a vector of i.i.d. standard normal random numbers. Since each evaluation requires a matrix-vector multiplication with $\boldsymbol{C}^{1/2}$, a direct approach requires $\mathcal{O}(N^2)$ operations for the multiplication plus $\mathcal{O}(N^3)$ operations for computing the square-root itself and thus is prohibitively expensive. An efficient method first proposed in [4,3] is *circulant embedding*, which employs fast FFT techniques to realize the factorization and the matrix-vector multiplication in $\mathcal{O}(N \log(N))$ operations. This approach, however, works solely for stationary covariance functions $\varrho(\boldsymbol{x}, \boldsymbol{y}) = \rho(|\boldsymbol{x} - \boldsymbol{y}|)$ and regular grids of quadrature nodes. Since non-stationary covariance functions are of great interest for the modeling of natural structures (e.g., porous rock, wood,...), and since finite element methods often use irregular grids, we propose a new method which removes both restrictions.

The idea is to approximate the covariance matrix $\boldsymbol{C}$ by an H^2-matrix, as described in, e.g, [2], and to use an iterative method to compute an approximation $\mathcal{Z}_{k,p}(\boldsymbol{z})$ (k and p are parameters of the methods, see below) to $\boldsymbol{C}^{1/2}\boldsymbol{z}$ for any $\boldsymbol{z} \in \mathbb{R}^N$. We therefore obtain the approximation to the random field by feeding the algorithms with i.i.d. standard normal random vectors $\boldsymbol{z}(\omega) \in \mathbb{R}^N$, i.e.,

$$\mathcal{Z}(\boldsymbol{x}_i, \omega) \approx \mathcal{Z}_{k,p}(\boldsymbol{z}(\omega))_i \quad \text{for all } i \in \{1, \ldots, N\}.$$

This is feasible since matrix-vector multiplication with H^2-matrices can be done in $\mathcal{O}(N)$ operations. The only assumption on the covariance function of the random field is that it is asymptotically smooth. We propose two iterative algorithms, each with individual advantages for smooth or rough random fields. This algorithms might also be of interest for the approximation of random fields with covariance kernels of random solutions of certain stochastic operator equations, as considered in [5].

The idea to use H-matrices for random field approximation has already been used indirectly in [16, 11], where the authors efficiently compute eigenfunctions of the covariance operator by use of H-matrix techniques.

M. Feischl, F.Y. Kuo, I.H. Sloan
School of Mathematics and Statistics, UNSW Sydney, NSW 2052
Tel.: +61-2-93857076
E-mail: m.feischl@unsw.edu.au, f.kuo@unsw.edu.au, i.sloan@unsw.edu.au

1.1 Notation

Throughout the text, $\alpha \lesssim \beta$ denotes $\alpha \leq C\beta$ for some generic constant $C > 0$ and $\alpha \simeq \beta$ means $\alpha \lesssim \beta$ and $\beta \lesssim \alpha$. The notation $|\cdot|$ has several unambiguous meanings: for vectors, it denotes the euclidean norm, while for sets, $|\cdot|$ is the natural measure, which is the Lebesgue measure (volume, area) for continuous sets and the counting measure (cardinality) for finite sets. The notation $\|\cdot\|_2$ is used for the spectral matrix norm and $|\boldsymbol{z}|_p := (\sum_{j=1}^{N} |\boldsymbol{z}_i|^p)^{1/p}$ for all $\boldsymbol{z} \in \mathbb{R}^N$ denotes the ℓ_p-norm. By $\mathcal{P}^k$ we denote the set of polynomials of maximal degree k. For brevity, we write $|\cdot| := |\cdot|_2$. We denote the maximal and minimal eigenvalues of a positive definite and symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ by

$$\lambda_{\max}(\boldsymbol{M}) := \sup_{\boldsymbol{z} \in \mathbb{R}^N \setminus \{0\}} \frac{|\boldsymbol{M}\boldsymbol{z}|}{|\boldsymbol{z}|} \quad \text{and} \quad \lambda_{\min}(\boldsymbol{M}) := \inf_{\boldsymbol{z} \in \mathbb{R}^N \setminus \{0\}} \frac{(\boldsymbol{M}\boldsymbol{z})^T \boldsymbol{z}}{|\boldsymbol{z}|^2}.$$

We denote the k-th component of a vector $\boldsymbol{v} \in \mathbb{R}^N$ by $\boldsymbol{v}_k$, whereas sequences of vectors are denoted by $\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots$.

2 Model Problem

Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space and let $D \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$ be a Lipschitz domain. We consider a random field which is normal or log-normal,

$$\mathcal{Z}(\boldsymbol{x}, \omega) \quad \text{or} \quad \exp(\mathcal{Z}(\boldsymbol{x}, \omega)) \quad \text{for all } \omega \in \Omega, \, \boldsymbol{x} \in D$$

for some zero-mean Gaussian random field $\mathcal{Z}(\cdot, \cdot)$ (note that the assumption on the mean is purely for brevity of presentation). The covariance function $\varrho \colon D \times D \to \mathbb{R}$ of $\mathcal{Z}(\cdot, \cdot)$ is assumed asymptotically smooth: that is, $\varrho \in C^\infty(\{(\boldsymbol{x}, \boldsymbol{y}) \in D \times D : \boldsymbol{x} \neq \boldsymbol{y}\})$ and there exist constants $c_1, c_2 > 0$ such that

$$|\partial_{\boldsymbol{x}}^\alpha \partial_{\boldsymbol{y}}^\beta \varrho(\boldsymbol{x}, \boldsymbol{y})| \leq c_1 (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|\alpha|_1 - |\beta|_1} |\alpha + \beta|_1! \quad \text{for all } \boldsymbol{x} \neq \boldsymbol{y} \in D, \tag{1}$$

for all multi-indices $\alpha, \beta \in \mathbb{N}_0^d$ with $|\alpha|_1 + |\beta|_1 \geq 1$. (The expert reader will notice that the original definition of asymptotically smooth includes a singularity order. As our covariance functions are always finite in value, we do not consider this.) The goal of this work is to derive an efficient method which evaluates the random field at certain (quadrature) points $\mathcal{N} \subseteq D$, where $\mathcal{N} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ is a finite set, i.e., we aim to approximate

$$\left(\mathcal{Z}(\boldsymbol{x}, \omega)\right)_{\boldsymbol{x} \in \mathcal{N}} \in \mathbb{R}^N \quad \text{or} \quad \left(\exp(\mathcal{Z}(\boldsymbol{x}, \omega))\right)_{\boldsymbol{x} \in \mathcal{N}} \in \mathbb{R}^N$$

for given $\omega \in \Omega$.

2.1 Examples of valid covariance functions

The condition above includes the important class of isotropic stationary covariance functions of Matérn form, e.g.,

$$\varrho(\boldsymbol{x}, \boldsymbol{y}) = \sigma^2 \frac{2^{1-\mu}}{\Gamma(\mu)} \left(\sqrt{2\mu} \frac{|\boldsymbol{x} - \boldsymbol{y}|_p}{\lambda}\right)^\mu K_\mu\left(\sqrt{2\mu} \frac{|\boldsymbol{x} - \boldsymbol{y}|_p}{\lambda}\right), \tag{2}$$

where $\Gamma(\cdot)$ is the gamma function, K_μ is the modified Bessel function of second kind, and $\lambda, \sigma > 0$, $\mu \in (0, \infty]$, $p \in \mathbb{N}$ are parameters. For $\mu = 1/2$, the above function takes the form

$$\varrho(\boldsymbol{x}, \boldsymbol{y}) = \sigma^2 \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|_p}{\lambda}\right)$$

and the limit case $\mu = \infty$ satisfies

$$\varrho(\boldsymbol{x}, \boldsymbol{y}) = \sigma^2 \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|_p^2}{2\lambda^2}\right).$$

Also much more general non-stationary, non-isotropic covariance functions, e.g.,

$$\varrho(\boldsymbol{x}, \boldsymbol{y}) := \sigma^2 \frac{\det(\boldsymbol{\Sigma_x})^{1/4} \det(\boldsymbol{\Sigma_y})^{1/4}}{\sqrt{2} \det(\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y})^{1/2}} \exp\left(-(\boldsymbol{x} - \boldsymbol{y})^T \frac{(\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y})^{-1}}{2} (\boldsymbol{x} - \boldsymbol{y}) \right). \tag{3}$$

satisfy the assumptions. Here, $\boldsymbol{\Sigma}_{(\cdot)} \colon D \to \mathbb{R}^{d \times d}$ is a smooth mapping into the symmetric positive definite matrices and $\sigma > 0$ is a parameter. This covariance function was first suggested in [12] to model spatially dependent anisotropies in a material.

Lemma 1 *The covariance functions from* (2) *satisfy* (1). *Assume the mapping* $\boldsymbol{x} \mapsto \boldsymbol{\Sigma_x}$ *satisfies (for any matrix norm* $\| \cdot \|$)

$$\sup_{\alpha \in \mathbb{N}^d} \sup_{\boldsymbol{x} \in D} \|\partial_{\boldsymbol{x}}^\alpha \boldsymbol{\Sigma_x}\| < \infty. \tag{4}$$

Then, the covariance function from (3) *is asymptotically smooth* (1).

We postpone the proof of the lemma to Appendix A.

3 Sampling the random field

By definition, $\mathcal{Z}(\boldsymbol{x}, \cdot)$, $\boldsymbol{x} \in \mathcal{N}$ is a Gaussian random field with covariance matrix $\boldsymbol{C} \in \mathbb{R}^{N \times N}$, $N = |\mathcal{N}|$, and $\boldsymbol{C}_{ij} = \varrho(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where we write $\mathcal{N} := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. The main goal of this section is to establish a new way to efficiently approximate $\boldsymbol{C}^{1/2} \boldsymbol{z}$ for given $\boldsymbol{z} \in \mathbb{R}^N$. Roughly, the strategy is to approximate $\boldsymbol{C}$ by an H^2-matrix and to benefit from the fast matrix-vector multiplication provided by it. This allows us to efficiently approximate $\boldsymbol{A} \boldsymbol{z}$ (without actually factorizing the matrix $\boldsymbol{C}$).

3.1 H^2-matrix approximation of the covariance matrix

Given the finite set of evaluation points $\mathcal{N} := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset D$, we approximate the covariance matrix $\boldsymbol{C} \in \mathbb{R}^{N \times N}$, $\boldsymbol{C}_{ij} := \varrho(\boldsymbol{x}_i, \boldsymbol{x}_j)$ by an H^2-matrix $\boldsymbol{C}_p$ via interpolation of order $p \in \mathbb{N}$.

In the following, we recall the definition of H^2-matrices and the approximation process as laid out in, e.g., [2]. The rough idea is to partition the index set of the covariance matrix into *far-field* blocks, which can be approximated efficiently by interpolation of the covariance function, and *near-field* blocks, which are stored exactly.

3.1.1 Block partitioning

For each subset $X \subseteq \mathcal{N}$, we denote by $B_X \subseteq \mathbb{R}^d$, the smallest axis-parallel box such that $X \subseteq B_X$. We build a binary tree of clusters in the following way. Let $X_{\text{root}} := \mathcal{N} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ denote the root of the tree which has level zero $\text{level}(X_{\text{root}}) = 0$ by definition. For each node of the tree X with $|X| > C_{\text{leaf}}$ for some cut-off constant $C_{\text{leaf}} \geq 2$ (usually $C_{\text{leaf}} \approx 20$), we define two sons of X as follows: Split B_X in half along its longest edge into $B_0 \cup B_1 = B_X$. Define $\text{sons}(X) := \{X_0, X_1\}$ with $X_0 := X \cap B_0$ and $X_1 := X \setminus X_0$ and set $\text{level}(X_i) = \text{level}(X) + 1$ for $i = 0, 1$. For a node X with $|X| \leq C_{\text{leaf}}$, we define $\text{sons}(X) := \emptyset$. This procedure generates a binary tree denoted by $\mathbb{T}_{\text{cl}}$ (where cl stands for *cluster*) and guarantees that its leaves satisfy $|X| \leq C_{\text{leaf}}$.

For a parameter $\eta > 0$, we consider the admissibility condition for axis parallel boxes $B, B' \subseteq \mathbb{R}^d$

$$\max\{\text{diam}(B), \text{diam}(B')\} \leq \eta \, \text{dist}(B, B'), \tag{5}$$

where the euclidean distance between the bounding boxes is defined by

$$\text{dist}(B, B') := \inf_{\boldsymbol{x} \in B, \boldsymbol{y} \in B'} |\boldsymbol{x} - \boldsymbol{y}|.$$

The condition (5) will be used to build the block-cluster tree $\mathbb{T} \subseteq \mathbb{T}_{cl} \times \mathbb{T}_{cl}$ as follows. The root of $\mathbb{T}$ is (X_{root}, X_{root}). For each node $(X, Y) \in \mathbb{T}$ of the tree, define sons(X, Y), the set of sons, as:

$$\begin{cases} \textbf{if } B_X \text{ and } B_Y \text{ satisfy (5) or if sons}(X) = \emptyset = \text{sons}(Y) \textbf{ set sons}(X, Y) = \emptyset \\ \textbf{else if } \text{sons}(Y) \neq \emptyset \text{ and sons}(X) = \emptyset \textbf{ set sons}(X, Y) = \{X\} \times \text{sons}(Y) \\ \textbf{else if } \text{sons}(X) \neq \emptyset \text{ and sons}(Y) = \emptyset \textbf{ set sons}(X, Y) = \text{sons}(X) \times \{Y\} \\ \textbf{else } \text{sons}(X) \neq \emptyset \text{ and sons}(Y) \neq \emptyset \textbf{ set sons}(X, Y) = \text{sons}(X) \times \text{sons}(Y) \end{cases}$$

We also define the level as level$(X_{root}, X_{root}) = 0$ and level$(X, Y) = \text{level}(X', Y') + 1$ for $(X, Y) \in$ sons(X', Y'). Further, we define

$$\mathbb{T}_{far} := \big\{ (X, Y) \in \mathbb{T} : \text{sons}(X, Y) = \emptyset \text{ and } B_X, B_Y \text{ satisfy (5)} \big\}$$

as well as

$$\mathbb{T}_{near} := \big\{ (X, Y) \in \mathbb{T} : \text{sons}(X, Y) = \emptyset \text{ and } B_X, B_Y \text{ do not satisfy (5)} \big\}.$$

Note that by definition of the block-cluster tree $\mathbb{T}$, the set $\mathbb{T}_{near} \cup \mathbb{T}_{far}$ contains all the leaves of $\mathbb{T}$. Moreover, we see that for each $(X, Y) \in \mathbb{T} \setminus (\mathbb{T}_{near} \cup \mathbb{T}_{far})$, there holds

$$X \times Y = \bigcup_{(X', Y') \in \text{sons}(X, Y)} X' \times Y'$$

Therefore, $\mathbb{T}_{near} \cup \mathbb{T}_{far}$ is a partition of $\mathcal{N} \times \mathcal{N}$ in the sense that each pair of points $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{N} \times \mathcal{N}$ for $1 \leq i, j \leq N$ is contained in exactly one $(X, Y) \in \mathbb{T}_{near} \cup \mathbb{T}_{far}$.

3.1.2 Interpolation

The blocks $(X, Y) \in \mathbb{T}_{far}$ satisfy (5) and hence interpolation of the kernel function is highly accurate. This allows us to store the matrix very efficiently. Let $I(X) := \big\{ i \in \mathbb{N} : \boldsymbol{x}_i \in X \big\}$ denote the index set of X. The basic idea now is to replace $\boldsymbol{C}|_{I(X) \times I(Y)}$ by a low-rank approximation $\boldsymbol{V}^X \boldsymbol{M}^{XY} (\boldsymbol{V}^Y)^T$ with $\boldsymbol{V}^X \in \mathbb{R}^{|X| \times p^d}$, $\boldsymbol{M}^{XY} \in \mathbb{R}^{p^d \times p^d}$, and $\boldsymbol{V}^Y \in \mathbb{R}^{|Y| \times p^d}$, where p is the interpolation order. The three matrices are defined by Chebychev interpolation of the covariance function. To that end, let $\{q_1^X, \dots, q_{p^d}^X\}$ denote transformed, tensorial Chebychev nodes in B_X with the corresponding Lagrange basis functions $L_1^X, \dots, L_{p^d}^X \colon B_X \to \mathbb{R}$. Given $(X, Y) \in \mathbb{T}_{far}$, we may approximate

$$\varrho(\boldsymbol{x}, \boldsymbol{y}) \approx c_p^{XY}(\boldsymbol{x}, \boldsymbol{y}) := \sum_{n,m=1}^{p^d} \varrho(q_n^X, q_m^Y) L_n^X(\boldsymbol{x}) L_m^Y(\boldsymbol{y}) \quad \text{for all } \boldsymbol{x} \in X, \boldsymbol{y} \in Y.$$

For $i, j \in \{1, \dots, N\}$ and $n, m \in \{1, \dots, p^d\}$, this leads to

$$\boldsymbol{V}_{in}^X := L_n^X(\boldsymbol{x}_i), \quad \boldsymbol{V}_{jm}^Y := L_m^Y(\boldsymbol{x}_j), \text{ and } \boldsymbol{M}_{nm}^{XY} := \varrho(q_n^X, q_m^Y)$$

and hence

$$\boldsymbol{C}|_{I(X) \times I(Y)} \approx \boldsymbol{V}^X \boldsymbol{M}^{XY} (\boldsymbol{V}^Y)^T.$$

The admissibility condition (5) guarantees that the approximation error converges to zero exponentially in p, as we prove in Proposition 1 below. Further note that the Chebychev interpolation described above is exact on polynomials of degree p. Thus, for $X \in \mathbb{T}_{cl}$ and $\boldsymbol{x}_i \in X' \in \text{sons}(X)$, there holds with the transfer matrices $\boldsymbol{T}^{X'X} := (L_n^X(q_m^{X'}))_{mn} \in \mathbb{R}^{p^d \times p^d}$

$$\boldsymbol{V}_{in}^X := L_n^X(\boldsymbol{x}_i) = \sum_{m=1}^{p^d} L_n^X(q_m^{X'}) L_m^{X'}(\boldsymbol{x}_i) = \sum_{m=1}^{p^d} L_n^X(q_m^{X'}) \boldsymbol{V}_{im}^{X'} = (\boldsymbol{V}^{X'} \boldsymbol{T}^{X'X})_{in}.$$

Thus, it suffices to store $\boldsymbol{V}^X$ only for the leaves of $\mathbb{T}_{cl}$ together with the transfer matrices $\boldsymbol{T}^{X'X}$. This enables very efficient storage and arithmetics for H^2 matrices.

The capabilities of H^2-matrices which we employ in this work are summarized below in Proposition 1. To that end, we assume that the points $\mathcal{N}$ are approximately uniformly distributed, in the following sense.

5

Assumption 1 (quasi-uniform distribution) *We say that $\mathcal{N}$ is quasi-uniformly distributed if there exists a constant $C_{\mathrm{u}} > 0$ such that*

$$C_{\mathrm{u}}^{-1} N^{-1/d} \leq \min_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{N}} |\boldsymbol{x} - \boldsymbol{x}'| \leq \sup_{\boldsymbol{x} \in D} \min_{\boldsymbol{x}' \in \mathcal{N}} |\boldsymbol{x} - \boldsymbol{x}'| \leq C_{\mathrm{u}} N^{-1/d}.$$

Proposition 1 *Suppose we have a covariance matrix $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ and an asymptotically smooth kernel $\varrho(\cdot, \cdot)$ and recall Assumption 1 on approximate uniform distribution of $\mathcal{N}$. Then, there exists a constant $C_H > 0$ such that, for all $p \in \mathbb{N}_0$, the H^2-matrix $\boldsymbol{C}_p \in \mathbb{R}^{N \times N}$ constructed as above satisfies*

$$\|\boldsymbol{C} - \boldsymbol{C}_p\|_2 \leq \|\boldsymbol{C} - \boldsymbol{C}_p\|_F := \Big(\sum_{i,j=1}^{N} |\boldsymbol{C} - \boldsymbol{C}_p|_{ij}^2 \Big)^{1/2} \leq C_H N (\log(p) + 1)^{2d-1} \Big(\frac{\eta}{4c_2} \Big)^p. \qquad (6)$$

(The constant c_2 is defined in (1).) The H^2-matrix $\boldsymbol{C}_p$ is symmetric and can be stored using less than $C_H p^{2d} N$ memory units. Moreover, given any vector $\boldsymbol{x} \in \mathbb{R}^N$, it is possible to compute $\boldsymbol{C}_p x \in \mathbb{R}^N$ in less than $C_H p^{2d} N$ arithmetic operations. The constant C_H depends only on C_{leaf} and d. The matrix $\boldsymbol{C}_p$ is positive definite if p is sufficiently large such that

$$C_H N (\log(p) + 1)^{2d-1} \Big(\frac{\eta}{4c_2} \Big)^p < \lambda_{\min}(\boldsymbol{C}). \qquad (7)$$

We postpone the proof of the lemma to Appendix B.

3.2 Computing the square-root (Method 1)

Since $\boldsymbol{C}$ is positive definite in our case, a standard method is to compute the Cholesky factorization $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{C}$. This can be done using H^2-matrices in *almost* linear cost (analyzed in [10] for H-matrices, but the method transfers to H^2-matrices). However, to the authors' best knowledge, there is no complete error analysis available, and due to the complicated structure of the algorithm, the worst-case error estimate may be overly pessimistic. Therefore, we propose an iterative algorithm based on a variant of the Lanczos iteration. Note that polynomial or rational approximations of the square root (as pursued in, e.g., [17]) are doomed to fail since smooth random fields result in very badly conditioned covariance matrices $\boldsymbol{C}$ (see also the numerical experiments below). This implies that a polynomial approximation of the square root over the spectrum of $\boldsymbol{C}$ is very costly, whereas a rational approximation requires the inverse of $\boldsymbol{C}$ which is hard to compute due to the bad condition number.

The idea behind the algorithm below is as follows. Given a positive definite symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ and a vector $\boldsymbol{z} \in \mathbb{R}^N$, the aim is to compute efficiently an approximation to $\boldsymbol{M}^{1/2}\boldsymbol{z}$. For arbitrary $k \leq N$ define the order-k Krylov subspace of $\boldsymbol{M}$ and $\boldsymbol{z}$ as

$$\mathcal{K}_k := \mathrm{span}\{\boldsymbol{z}, \boldsymbol{M}\boldsymbol{z}, \boldsymbol{M}^2\boldsymbol{z}, \ldots, \boldsymbol{M}^{k-1}\boldsymbol{z}\}. \qquad (8)$$

Assuming $\mathcal{K}_k$ is k-dimensional, consider the orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{N \times k}$ whose columns are the orthonormal basis vectors of the Krylov subspace, i.e., $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_k$ and $\mathrm{range}(\boldsymbol{Q}) = \mathcal{K}_k$. Now define $\boldsymbol{U} \in \mathbb{R}^{k \times k}$ by

$$\boldsymbol{U} := \boldsymbol{Q}^T\boldsymbol{M}\boldsymbol{Q}.$$

If $k = N$ then $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_N$ and $\boldsymbol{Q}\boldsymbol{U}\boldsymbol{Q}^T = \boldsymbol{M}$, from which it follows that

$$\boldsymbol{M}^{1/2}\boldsymbol{z} = \boldsymbol{Q}\boldsymbol{U}^{1/2}\boldsymbol{Q}^T\boldsymbol{z}. \qquad (9)$$

The algorithm relies on explicit matrix multiplication to construct $\boldsymbol{U}$ and then a direct factorization of $\boldsymbol{U}$, thus for large N it is feasible only when $k \ll N$, in which case (9) does not hold exactly. However, as we show later it may hold to a good enough approximation. The following Lanczos type algorithm builds up progressively the columns of $\boldsymbol{Q}$ without fully computing $\mathcal{K}_k$ first.

Remark 1 In the following, we make frequent use of the QR-factorisation of matrices and therefore recall the most important facts: For a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times k}$ with $k \leq n \in \mathbb{N}$, there exists a QR-factorization $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ such that $\boldsymbol{Q} \in \mathbb{R}^{n \times k}$ and $\boldsymbol{R} \in \mathbb{R}^{k \times k}$. The columns of $\boldsymbol{Q}$ are orthonormal and for $1 \leq j \leq \mathrm{rank}(\boldsymbol{A})$, the first j columns of $\boldsymbol{Q}$ span the same linear space as the first j columns of $\boldsymbol{A}$. Moreover, $\boldsymbol{R}$ is upper triangular. If we restrict to positive diagonal entries of $\boldsymbol{R}$, the factorization is unique if $\mathrm{rank}(\boldsymbol{A}) = k$.

Algorithm 1 **Input:** *positive definite symmetric matrix* $\boldsymbol{M} \in \mathbb{R}^{N \times N}$, *vector* $\boldsymbol{z} \in \mathbb{R}^N$, *and maximal number of iterations* $k \in \mathbb{N}$.

1. *Compute Krylov subspace: Set* $\boldsymbol{Q}_1 := \boldsymbol{z}/|\boldsymbol{z}| \in \mathbb{R}^{N \times 1}$ *and* $k_0 = k$. *For* $j = 2, \dots, k$ *do:*
   (a) *Compute* $\widetilde{\boldsymbol{q}} := \boldsymbol{M}\boldsymbol{q}^{j-1} \in \mathbb{R}^N$, *where* $\boldsymbol{q}^{j-1}$ *is the* $(j-1)$-*th column of* $\boldsymbol{Q}_{j-1} \in \mathbb{R}^{N \times (j-1)}$.
   (b) *Compute QR-factorization* $\boldsymbol{Q}_j \in \mathbb{R}^{N \times j}$ *(with orthonormal columns)*, $\boldsymbol{R}_j \in \mathbb{R}^{j \times j}$ *(upper triangular) such that* $\boldsymbol{Q}_j \boldsymbol{R}_j = (\boldsymbol{Q}_{j-1}, \widetilde{\boldsymbol{q}}) \in \mathbb{R}^{N \times j}$.
   (c) *If* $(\boldsymbol{R}_j)_{jj} = 0$, *set* $k_0 = j - 1$ *and goto Step 2.*
2. *Compute* $\boldsymbol{U}_{k_0} := \boldsymbol{Q}_{k_0}^T \boldsymbol{M} \boldsymbol{Q}_{k_0} \in \mathbb{R}^{k_0 \times k_0}$.
3. *Compute* $\boldsymbol{U}_{k_0}^{1/2}$ *directly.*
4. *Return* $\boldsymbol{y} = \boldsymbol{Q}_{k_0} \boldsymbol{U}_{k_0}^{1/2} \boldsymbol{Q}_{k_0}^T \boldsymbol{z}$.

Output: *Approximation* $\boldsymbol{y} \approx \boldsymbol{M}^{1/2}\boldsymbol{z}$ *and number of steps* k_0.

Remark 2 Obviously, the orthogonal basis $\boldsymbol{q}^1, \dots, \boldsymbol{q}^k$ could also be generated by Gram-Schmidt orthogonalization. However, numerical experiments show that this is not stable with respect to roundoff errors. Moreover, also the classical Lanczos algorithm seems to be prone to rounding errors, especially for ill-conditioned matrices. Therefore, we propose to use the QR-factorization as above.

Remark 3 As proved in Lemma 4 below (and as is easily verified), a generic QR-algorithm produces $\boldsymbol{Q}_j$ which coincides with the first j columns of $\boldsymbol{Q}$ up to signs. For simplicity, we assume in the following that the QR-algorithm ensures that the diagonal entries of $\boldsymbol{R}_j$ are always non-negative. This guarantees that the first j columns of $\boldsymbol{Q}_{j+1}$ coincide with $\boldsymbol{Q}_j$. Thus, it suffices to store only the new column $\boldsymbol{q}^j$.

Theorem 1 *Let* $0 < \eta < 4c_2$ *and let* p *be sufficiently large such that* $\boldsymbol{C}_p$ *constructed from* $\boldsymbol{C}$ *as in Section 3.1 is positive definite (condition* (7) *is sufficient), and suppose Assumption 1 holds. Given* $\boldsymbol{z} \in \mathbb{R}^N$, *call Algorithm 1 with* $\boldsymbol{M} = \boldsymbol{C}_p$, $\boldsymbol{z}$, *and a maximal number of iterations* $k \in \mathbb{N}$. *The output of Algorithm 1 contains the approximation* $\mathcal{Z}_{k,p}(\boldsymbol{z}) := \boldsymbol{y} \in \mathbb{R}^N$ *to* $\boldsymbol{C}^{1/2}\boldsymbol{z}$ *and the step number* $k_0 \leq k$.

(i) *There holds with Kronecker's delta* $\delta_{i,j}$

$$\frac{|\boldsymbol{C}^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} \leq \delta_{k_0,k}\sqrt{2\|\boldsymbol{M}\|_2}\frac{4r^2}{r-1}r^{-k} + \frac{2C_{\mathrm{H}}N(\log(p)+1)^{2d-1}\left(\frac{\eta}{4c_2}\right)^p}{\max\{\lambda_{\min}(\boldsymbol{C}), \lambda_{\min}(\boldsymbol{C}_p)\}^{1/2}},$$

where C_{H}, η, c_2, *and* p *are as in Proposition 1, and*

$$r := \frac{\lambda_{\max}(\boldsymbol{C}_p) + \lambda_{\min}(\boldsymbol{C}_p)}{\lambda_{\max}(\boldsymbol{C}_p) - \lambda_{\min}(\boldsymbol{C}_p)} > 1.$$

(ii) *Let* $\lambda_{\max}(\boldsymbol{C}_p) = \lambda_1 > \lambda_2 > \dots > \lambda_M > 0$ *denote the distinct eigenvalues of* $\boldsymbol{C}_p$ *for some* $M \leq N$ *and assume*

$$|\lambda_i - \lambda_j| \leq \lambda_{\max}(\boldsymbol{C}_p)C_\kappa \kappa^{\min\{i,j\}} \quad \text{for all } 1 \leq i, j \leq M$$

for some $C_\kappa > 0$ *and* $0 < \kappa < 1$, *then*

$$\frac{|\boldsymbol{C}^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} \leq \delta_{k_0,k}3\sqrt{\lambda_{\max}(\boldsymbol{C}_p)C_\kappa}\,\kappa^{k/4} + 3\sqrt{2C_{\mathrm{H}}N}(\log(p)+1)^{d-1/2}\left(\frac{\eta}{4c_2}\right)^{p/2}.$$

The algorithm completes in $\mathcal{O}(k^3 p^{2d}N)$ *arithmetic operations and uses less than* $\mathcal{O}(kN)$ *storage.*

Remark 4 The theorem covers two regimes of covariance matrices. Whereas case (i) is the classical Lanczos convergence analysis for well-conditioned matrices, case (ii) considers ill-conditioned matrices with rapidly decaying eigenvalues. The numerical examples in Section 4 suggest that the error estimates might be more or less sharp, since Algorithm 1 performs remarkably well for smooth random fields (with rapidly decaying eigenvalues) and very rough random fields (with well-conditioned covariance matrices). Note that $k_0 < k$ (hence $\delta_{k_0,k} = 0$) implies that the condition in the if-clause 1(c) is true. This however is an exotic case, meaning that $\boldsymbol{z}$ lies some non-trivial invariant subspace of $\boldsymbol{C}_p$ with fewer than k dimensions. In this situation the algorithm computes $\boldsymbol{C}_p^{1/2}\boldsymbol{z}$ exactly and only the H-matrix approximation error remains. We note that by use of (16) instead of (17) in the proof below, it is possible to replace $(k+1)/4$ by $(k+1)/2$ and $p/2$ by p in the exponents in (ii) at the price of including the square-root of the minimal eigenvalue in the denominator as in (i).

Proof (Proof of Theorem 1) The cost estimate is proved as follows. The Krylov subspace loop of Algorithm 1 completes at most k iterations. In each iteration, we have one H^2-matrix-vector multiplication which needs $\mathcal{O}(p^{2d}N)$ operations. Moreover, the QR-factorization needs $\mathcal{O}(Nk^2)$ arithmetic operations. After the matrix $\boldsymbol{Q}_k$ is set up, we have k H^2-matrix-vector multiplications to compute $\boldsymbol{MQ}_k$ and k^2 scalar products to compute $\boldsymbol{U}_{k_0}$. In total, this needs $\mathcal{O}(N(k+k^2))$ arithmetic operations. The computation of $\boldsymbol{U}_{k_0}^{1/2}$ can be done in $\mathcal{O}(k^3)$ operations (see, e.g., [13] for the algorithm and the corresponding analysis). Finally, to compute $\boldsymbol{y}$, we have k scalar products, a matrix vector multiplication with a $(k \times k)$ matrix and a matrix-matrix multiplication of $(N \times k)$ and $(k \times k)$ matrices, all of which can be done in $\mathcal{O}(Nk^2)$ arithmetic operations.

To see (i), we employ the triangle inequality

$$
\begin{aligned}
\frac{|\boldsymbol{C}^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} &\leq \frac{|\boldsymbol{C}_p^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} + \frac{|\boldsymbol{C}^{1/2}\boldsymbol{z} - \boldsymbol{C}_p^{1/2}\boldsymbol{z}|}{|\boldsymbol{z}|} \\
&\leq \frac{|\boldsymbol{C}_p^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} + \|\boldsymbol{C}_p^{1/2} - \boldsymbol{C}^{1/2}\|_2.
\end{aligned}
\tag{10}
$$

For the first term on the right-hand side, Lemma 6 below proves

$$
\frac{|\boldsymbol{C}_p^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} \leq \delta_{k_0,k}\sqrt{2\|\boldsymbol{M}\|_2}\frac{4r^2}{r-1}r^{-k}.
$$

As shown in (16) of Lemma 2 below, the second term on the right-hand side of (10) is bounded by

$$
\|\boldsymbol{C}_p^{1/2} - \boldsymbol{C}^{1/2}\|_2 \leq 2\max\{\lambda_{\min}(\boldsymbol{C}), \lambda_{\min}(\boldsymbol{C}_p)\}^{-1/2}\|\boldsymbol{C}_p - \boldsymbol{C}\|_2.
\tag{11}
$$

Hence, (i) follows from Proposition 1. For (ii), we note that the combination of both estimates in Proposition 2 below shows for $\boldsymbol{U}_j := \boldsymbol{Q}_j^T \boldsymbol{MQ}_j$

$$
\min_{1 \leq j \leq k} \frac{|\boldsymbol{C}_p^{1/2}\boldsymbol{z} - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T\boldsymbol{z}|}{|\boldsymbol{z}|} \leq \delta_{k_0,k}3\sqrt{\lambda_{\max}(\boldsymbol{C}_p)C_\kappa}\,\kappa^{k/4}.
$$

We may eliminate the minimum in the error estimate since Algorithm 1 is essentially (up to roundoff errors) of Lanczos type, and for this algorithm, [7, Example 5.1] shows that the approximation error $|\boldsymbol{C}_p^{1/2}\boldsymbol{z} - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T\boldsymbol{z}|$ decreases monotonically in j. Since $\boldsymbol{Q}_{k_0}(\boldsymbol{U}_{k_0}^{1/2})\boldsymbol{Q}_{k_0}^T\boldsymbol{z} = \mathcal{Z}_{k,p}(\boldsymbol{z})$, the remainder of the proof then follows as for (i) but we use (17) instead of (16) of Lemma 2 below.

3.3 Computing the square-root (Method 2)

The main drawback of Algorithm 1 is the additional storage requirements due to the necessity to store the matrix $\boldsymbol{Q}_k$. For this reason, we here follow a different approach, proposing a second algorithm that improves this situation.

The matrix sign function is defined for all square matrices $\widetilde{\boldsymbol{M}}$ with no pure imaginary eigenvalues as

$$
\text{sgn}(\widetilde{\boldsymbol{M}}) := \widetilde{\boldsymbol{M}}(\widetilde{\boldsymbol{M}}^2)^{-1/2}.
$$

The sign function $\text{sgn}(\widetilde{\boldsymbol{M}})$ can be computed using the Schultz iteration via

$$
\boldsymbol{M}_{k+1} = \frac{1}{2}\boldsymbol{M}_k(3\boldsymbol{I} - \boldsymbol{M}_k^2), \quad \boldsymbol{M}_0 = \widetilde{\boldsymbol{M}}.
\tag{12}
$$

The iterates $\boldsymbol{M}_k$ converge quadratically towards $\text{sgn}(\widetilde{\boldsymbol{M}})$ if $\|\boldsymbol{I} - \widetilde{\boldsymbol{M}}^2\|_2 < 1$ in any matrix norm (see [15, Theorem 5.2]). It is observed in [14], that all matrices $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ with only positive real eigenvalues satisfy

$$
\text{sgn}\begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{I} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \boldsymbol{M}^{1/2} \\ \boldsymbol{M}^{-1/2} & 0 \end{pmatrix},
$$

where $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ denotes the identity matrix, which opens the possibility to compute $\boldsymbol{M}^{1/2}$ via the sign function of the matrix By inserting

$$\widetilde{\boldsymbol{M}} := \begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{I} & 0 \end{pmatrix}.$$

By inserting this choice of $\widetilde{\boldsymbol{M}}$ into (12), we see that all iterates have the form

$$\boldsymbol{M}_k := \begin{pmatrix} 0 & \boldsymbol{A}_k \\ \boldsymbol{B}_k & 0 \end{pmatrix}.$$

As already observed in [14], this leads to the iteration

$$\boldsymbol{A}_{k+1} = \frac{1}{2}\boldsymbol{A}_k(3\boldsymbol{I} - \boldsymbol{B}_k\boldsymbol{A}_k), \quad \boldsymbol{B}_{k+1} = \frac{1}{2}\boldsymbol{B}_k(3\boldsymbol{I} - \boldsymbol{A}_k\boldsymbol{B}_k), \tag{13}$$

starting with $\boldsymbol{A}_0 = \boldsymbol{M}$ and $\boldsymbol{B}_0 = \boldsymbol{I} \in \mathbb{R}^{N \times N}$. The iterates $\boldsymbol{A}_k$ converge towards $\boldsymbol{M}^{1/2}$, which is what we aim to compute. The considerations above lead us to the following recursive form of the Schulz algorithm above, which uses only matrix vector multiplication. The subroutines `PartA` and `PartB` compute $\boldsymbol{A}_k\boldsymbol{z}$ and $\boldsymbol{B}_k\boldsymbol{z}$ respectively.

Algorithm 2 **Input:** *positive definite symmetric matrix* $\boldsymbol{M} \in \mathbb{R}^{N \times N}$, *vector* $\boldsymbol{z} \in \mathbb{R}^N$, *maximal number of iterations* $k \in \mathbb{N}$, *temporary storage vectors* $\boldsymbol{z}^j \in \mathbb{R}^N$, $j \in \{1, \ldots, k\}$, *and scaling factor* $0 < s < 2\|\boldsymbol{C}_p\|_2^{-1}$ *(the scaling factor ensures convergence of the algorithm).*
Main:

1. *Compute* $\boldsymbol{y} = \texttt{PartA}(s\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j)_{j=1}^k, k)$.
2. *Return* $\boldsymbol{y}/\sqrt{s}$.

   Output: *the approximation* $\boldsymbol{y} \approx \boldsymbol{M}^{1/2}\boldsymbol{z}$.
   Subroutines:
   $\texttt{PartA}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j), k)$:

(i) *If* $k = 0$, *return* $\boldsymbol{M}\boldsymbol{z}$.
(ii) *Compute* $\boldsymbol{z}^k := \texttt{PartA}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1)$ *and* $\boldsymbol{z}^k := \texttt{PartB}(\boldsymbol{M}, \boldsymbol{z}^k, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1)$.
(iii) *Compute* $\boldsymbol{z} := 3\boldsymbol{z} - \boldsymbol{z}^k$.
(iv) *Return* $\frac{1}{2}\texttt{PartA}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1)$.

   $\texttt{PartB}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j), k)$:

(i) *If* $k = 0$, *return* $\boldsymbol{z}$.
(ii) *Compute* $\boldsymbol{z}^k := \texttt{PartB}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1)$ *and* $\boldsymbol{z}^k := \texttt{PartA}(\boldsymbol{M}, \boldsymbol{z}^k, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1)$.
(iii) *Compute* $\boldsymbol{z} := 3\boldsymbol{z} - \boldsymbol{z}^k$.
(iv) *Return* $\frac{1}{2}\texttt{PartB}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1)$.

Remark 5 The extra storage vectors $(\boldsymbol{z}^j)_{j=1}^k$ are needed to avoid allocation of a new temporary storage vector in each call of either `PartA` are `PartB`. This would result in $\mathcal{O}(3^k)$ additional allocations. By supplying the additional storage vectors, we can exploit the fact that each level of recursion can share a single storage vector.

Theorem 2 *Suppose Assumption 1 holds and and let* $\boldsymbol{z} \in \mathbb{R}^N$. *If* $0 < \eta < 4c_2$ *and* p *is sufficiently large such that* $\boldsymbol{C}_p$ *constructed from* $\boldsymbol{C}$ *as in Section 3.1 is positive definite (condition (7) is sufficient), Algorithm 1 called with* $\boldsymbol{M} = \boldsymbol{C}_p$ *and* $0 < s < 2\|\boldsymbol{C}_p\|_2^{-1}$ *computes the approximation* $\mathcal{Z}_{k,p}(\boldsymbol{z}) := \boldsymbol{y} \in \mathbb{R}^N$ *such that*

$$\frac{|\boldsymbol{C}^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} \leq s^{-1/2}\kappa^{2^k} + \frac{2C_{\mathrm{H}}N(\log(p)+1)^{2d-1}\left(\frac{\eta}{4c_2}\right)^p}{\max\{\lambda_{\min}(\boldsymbol{C}), \lambda_{\min}(\boldsymbol{C}_p)\}^{1/2}},$$

where $\kappa := \max\{|1-s\lambda_{\max}(\boldsymbol{C}_p)|, |1-s\lambda_{\min}(\boldsymbol{C}_p)|\} < 1$. *The algorithm completes in* $\mathcal{O}(3^k p^{2d}N)$ *arithmetic operations and uses less than* kN *extra storage. The constant* C_{H} *is defined in Proposition 1.*

Remark 6 In contrast to Algorithm 1 which needs $\mathcal{O}(|\log_\kappa(\varepsilon)|N)$ extra storage (at least in case (ii)), we see that Algorithm 2 requires only $\mathcal{O}(\log|\log(\varepsilon)|N)$ additional storage for an error request of $\varepsilon > 0$.

Proof (Proof of Theorem 2)

First, we prove that `PartA` and `PartB` from Algorithm 2 correctly compute $\boldsymbol{A}_k\boldsymbol{z}$ and $\boldsymbol{B}_k\boldsymbol{z}$ from (13). This is done by induction on k. First, for $k = 0$, the output of `PartA` is obviously $\boldsymbol{M}\boldsymbol{z} = \boldsymbol{A}_0\boldsymbol{z}$ and the output of `PartB` is $\boldsymbol{z} = \boldsymbol{B}_0\boldsymbol{z}$. This confirms the case $k = 0$. Assume that `PartA` and `PartB` work correctly for $k \in \mathbb{N}$. By substitution of $\texttt{PartA}(\boldsymbol{M}, \boldsymbol{z}, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1) = \boldsymbol{A}_{k-1}\boldsymbol{z}$ and $\texttt{PartB}(\boldsymbol{M}, \boldsymbol{z}^k, (\boldsymbol{z}^j)_{j=1}^{k-1}, k-1) = \boldsymbol{B}_{k-1}\boldsymbol{z}^k$ in `PartA`, the variable $\boldsymbol{z}^k$ before step (iii) is given by $\boldsymbol{z}^k = \boldsymbol{B}_{k-1}\boldsymbol{A}_{k-1}\boldsymbol{z}$. Thus, step (iii)–(iv) correctly compute $\frac{1}{2}\boldsymbol{B}_{k-1}(3\boldsymbol{z} - \boldsymbol{B}_{k-1}\boldsymbol{A}_{k-1}\boldsymbol{z}) = \boldsymbol{A}_k\boldsymbol{z}$. During the execution of $\texttt{PartA}(\cdot,\cdot,\cdot,k)$, extra storage vector $\boldsymbol{z}^k$ is not accessed by other instances of the subroutines (the function calls to $\texttt{PartA}(\cdot,\cdot,\cdot,k-1)$ and $\texttt{PartB}(\cdot,\cdot,\cdot,k-1)$ access only $(\boldsymbol{z}^k)_{j=1}^{k-1}$). This ensures that the correct value of $\boldsymbol{z}^k$ is used at each point of the execution. Analogously, we argue that `PartB` works correctly and thus conclude the induction.

For the computational cost estimate, we prove by induction that each subroutine $\texttt{PartA}(\cdot,\cdot,\cdot,k)$ and $\texttt{PartB}(\cdot,\cdot,\cdot,k)$ requires less than

$$C\Big(3^k p^{2d} N + 2N \sum_{j=0}^{k-1} 3^j\Big) \tag{14}$$

operations for some universal constant $C \geq 1$ and all $k \in \mathbb{N}$. For $k = 0$, subroutine `PartA` performs an H^2-matrix-vector multiplication which, according to Proposition 1, costs less than $\mathcal{O}(p^{2d}N)$. Subroutine `PartB` just returns the vector $\boldsymbol{z}$. This shows (14) for $k = 0$ for both subroutines. Assume that (14) is correct for both subroutines for some $k > 0$. The fact that each subroutine $\texttt{PartA}(\cdot,\cdot,\cdot,k+1)$ and $\texttt{PartB}(\cdot,\cdot,\cdot,k+1)$ performs one scalar-vector multiplication and one vector addition as well as three calls to $\texttt{PartA}(\cdot,\cdot,\cdot,k)$ or $\texttt{PartB}(\cdot,\cdot,\cdot,k)$ shows that the cost of each subroutine $\texttt{PartA}(\cdot,\cdot,\cdot,k+1)$ and $\texttt{PartB}(\cdot,\cdot,\cdot,k+1)$ is bounded by

$$3C\Big(3^k p^{2d} N + 2N \sum_{j=0}^{k-1} 3^j\Big) + 2N = C\Big(3^{k+1} p^{2d} N + 2N \sum_{j=1}^{k} 3^j\Big) + 2N \leq C\Big(3^{k+1} p^{2d} N + 2N \sum_{j=0}^{k} 3^j\Big).$$

This concludes the proof of (14), which proves the cost estimate since

$$C\Big(3^k p^{2d} N + 2N \sum_{j=0}^{k-1} 3^j\Big) \leq C 3^k p^{2d} N + C 3^k N \leq 2C 3^k p^{2d} N.$$

To see the error estimate, we use (10) and note that Algorithm 2 is nothing else than a recursive version of the iteration (13). The scaling $s < 2\|\boldsymbol{C}_p\|_2^{-1}$ ensures $\kappa < 1$, since $\lambda \in \{\lambda_{\min}(\boldsymbol{C}_p), \lambda_{\max}(\boldsymbol{C}_p)\}$ satisfies $1 - s\lambda < 1$ (since $s, \lambda > 0$) as well as $s\lambda - 1 \leq s\|\boldsymbol{C}_p\|_2 - 1 < 2 - 1 = 1$. Thus, Lemma 8 shows

$$\frac{|\boldsymbol{C}_p^{1/2}\boldsymbol{z} - \mathcal{Z}_{k,p}(\boldsymbol{z})|}{|\boldsymbol{z}|} \leq s^{-1/2}\big(\max\{|1 - s\lambda_{\max}(\boldsymbol{C}_p)|, |1 - s\lambda_{\min}(\boldsymbol{C}_p)|\}\big)^{2^k} = s^{-1/2}\kappa^{2^k}.$$

We conclude the proof with the aid of (11) and Proposition 1.

4 Numerical experiments

All numerical experiments where computed in Matlab, by use of a Matlab-H^2-matrix library which can be downloaded under `software.michaelfeischl.net`. The authors are well aware that the Matlab implementation prohibits high-end performance. However, we wanted to demonstrate the feasibility of our algorithms and show the correct convergence rates, for which purpose the Matlab implementation is sufficient.

For the first example, we consider a covariance function of the form (3) with

$$\boldsymbol{\Sigma_x} := |\boldsymbol{x}|^2 \boldsymbol{I} \quad \text{and} \quad \boldsymbol{\Sigma_y} := |\boldsymbol{y}|^2 \boldsymbol{I}. \tag{15}$$
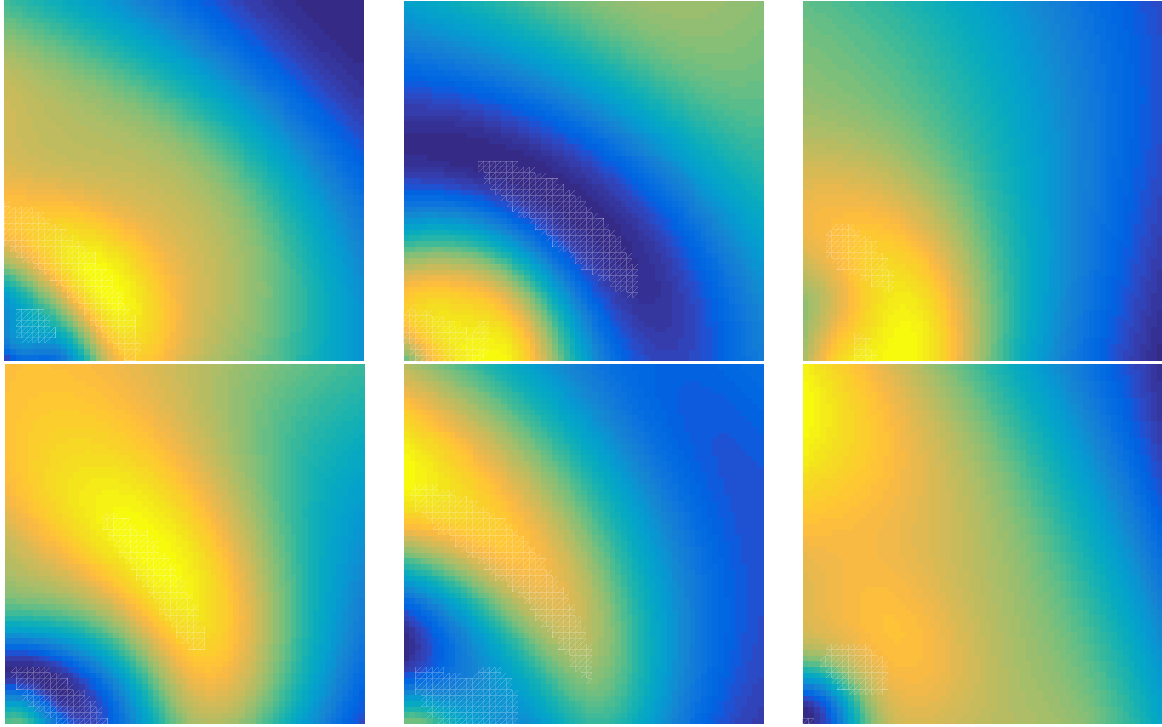
Fig. 1: Samples of $\mathcal{Z}$ with a non-stationary covariance function. We clearly observe the shorter covariance length (more variation) near the bottom left corner.
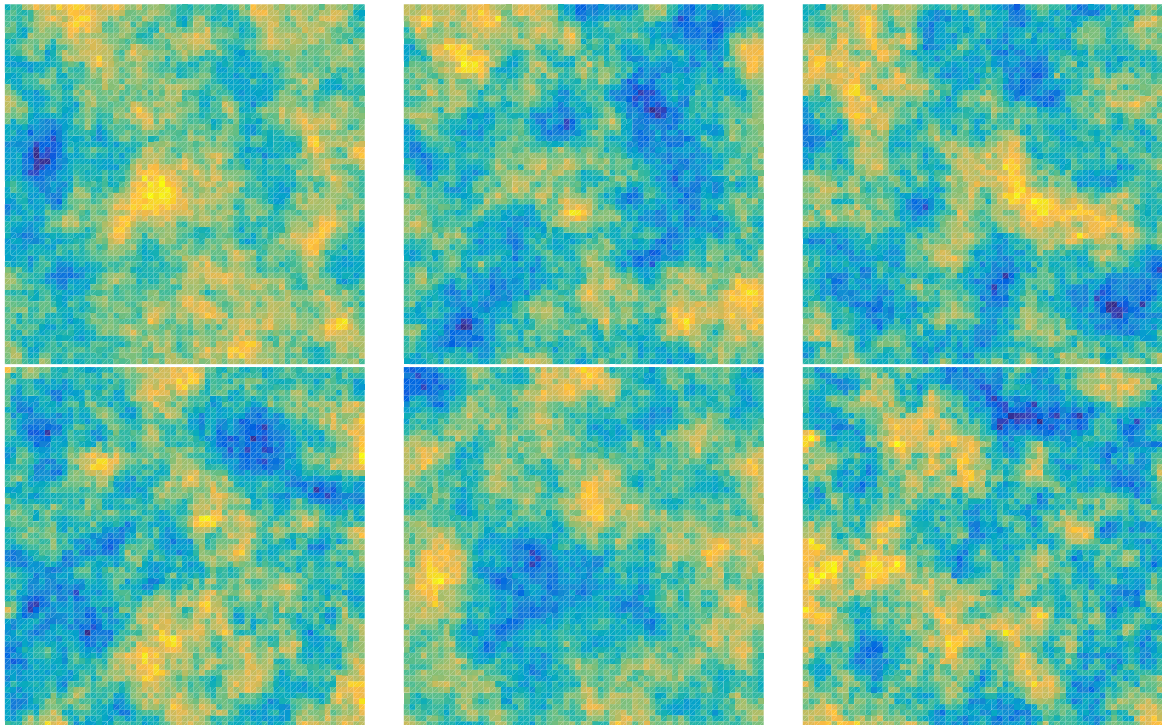

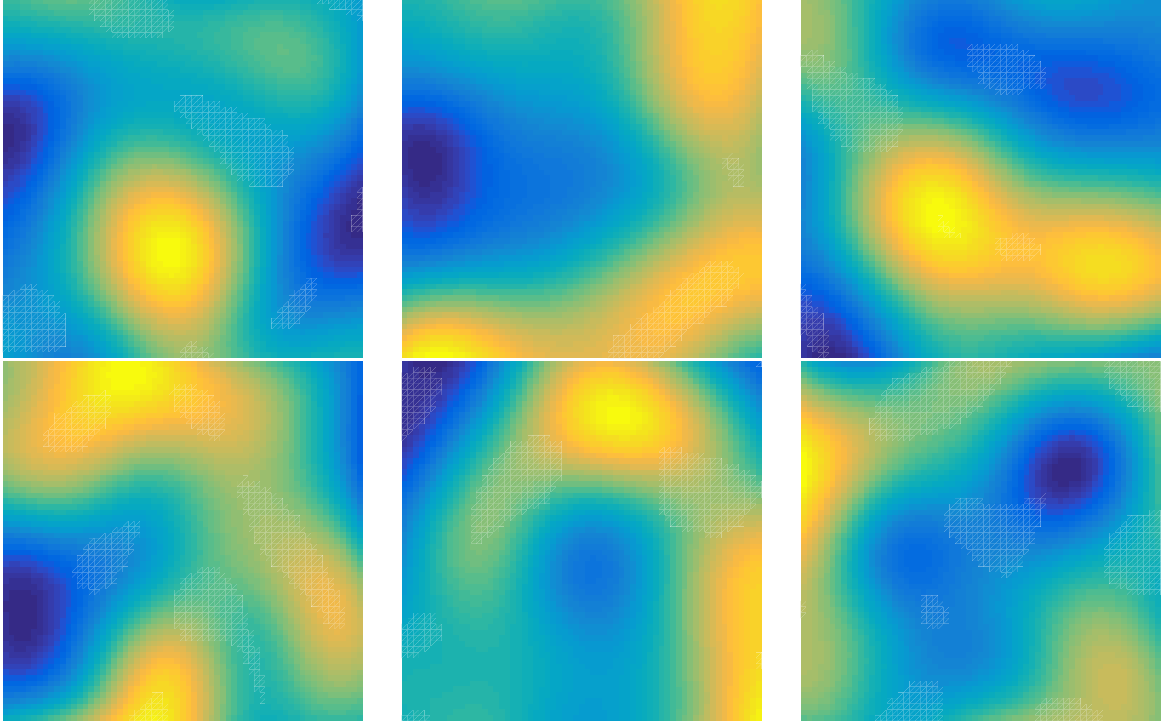Fig. 2: Samples of $\mathcal{Z}$ with a stationary covariance function from (2) with $p = 2$ and $\mu = 1/2$.

Fig. 3: Samples of $\mathcal{Z}$ with a stationary covariance function from (2) with $p = 2$ and $\mu = \infty$.

$m =$	5	6	7	8	9	10
$\lambda = 1$	2.0e+09	6.1e+16	8.6e+17	2.6e+19	1.8e+20	1.4e+20
$\lambda = 10^{-1}$	3.9e+07	5.5e+14	1.8e+17	8.4e+18	4.8e+20	4.6e+20
$\lambda = 10^{-2}$	6.5e+06	2.6e+12	2.7e+17	1.2e+19	3.3e+19	2.8e+20
$\lambda = 10^{-3}$	4.2e+06	9.4e+11	6.1e+17	4.2e+18	2.6e+19	1.1e+20

Table 1: Condition numbers of $\boldsymbol{C}$ for the covariance function from (2) with $\mathcal{N}$ being a Sobol point set with 2^m points.

We use Algorithm 1 to generate six samples on the unit square $D = [0,1]^2$ of the corresponding normal random field $\mathcal{Z}$ shown in Figure 1. Figure 2–3 show samples of the covariance functions from (2) with different parameters.

To illustrate the challenging nature of handling these covariance matrices, Table 1 shows condition numbers of $\boldsymbol{C}$ for different problem sizes and the Matérn covariance function (2).

For a performance comparison of Algorithm 1 and Algorithm 12, we consider the covariance function of the form (2) with $p = 2$, $\sigma = 1$, and varying $\mu \in \{1/2, \infty\}$, $\lambda \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$. We compute samples of $\mathcal{Z}(\boldsymbol{x}, \omega)$ on a Sobol pointset with 2^{10} points. The results are plotted in Figure 4 where we see the relative approximation error versus the computation time in seconds. We observe that with respect to computational time, Algorithm 1 is superior in almost all cases (particularly for smooth fields). However, keep in mind that according to Theorem 1, Algorithm 1 needs up to $\mathcal{O}(\log_{\kappa}(\varepsilon)N)$ extra storage, while Algorithm 2 uses only $\mathcal{O}(\log(\log(\varepsilon))N)$ extra storage units. (See Theorem 2, where the quadratic convergence shows that $k \simeq \log(\log(\varepsilon))$ is sufficient to reach a given accuracy $\varepsilon > 0$. However, we have to mention that k iterations of Algorithm 2 require $\mathcal{O}(3^k)$ arithmetic operations.)

Figure 5 compares the two algorithms with the direct matrix square root provided by Matlab. We evaluate $\mathcal{Z}(\boldsymbol{x}, \omega)$ on a Sobol pointset with size 2^m for $m \in \{1, \dots, 14\}$. The number of iterations in both algorithm is set such that the relative error is smaller than 10^{-10} for the example from above with $p = 2$, and varying $\mu \in \{1/2, \infty\}$, $\lambda \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$. We see that both, Algorithm 1–2, perform in linear time, whereas the direct approach comes closer to $\mathcal{O}(N^3)$. Even though our H^2-matrix library
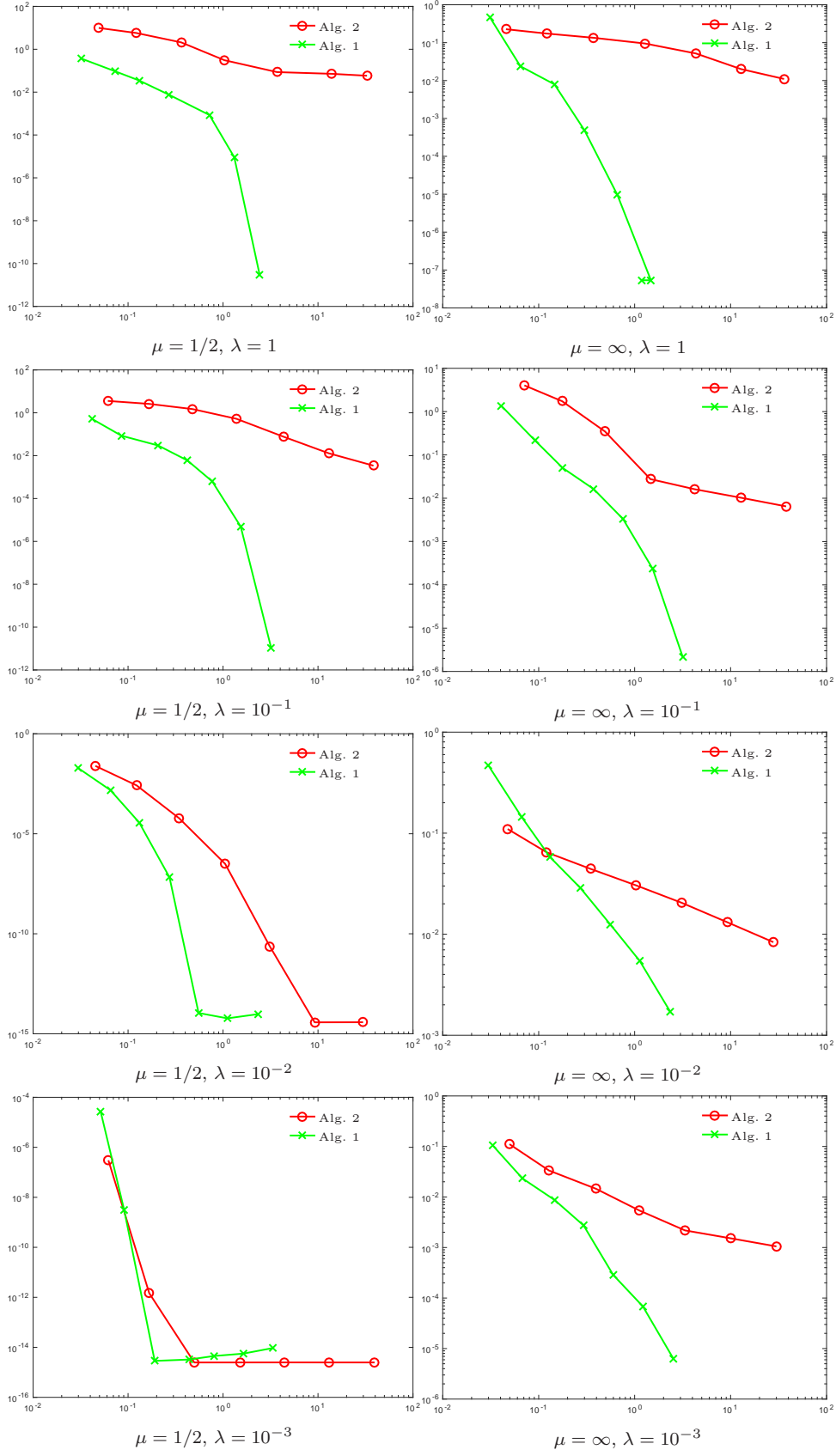
Fig. 4: Comparison of Algorithm 2 and Algorithm 1. We plot the relative error $|\mathcal{Z}_{k,p}(\boldsymbol{z}) - \boldsymbol{C}^{1/2}\boldsymbol{z}|/|\boldsymbol{z}|$ versus computation time in seconds.
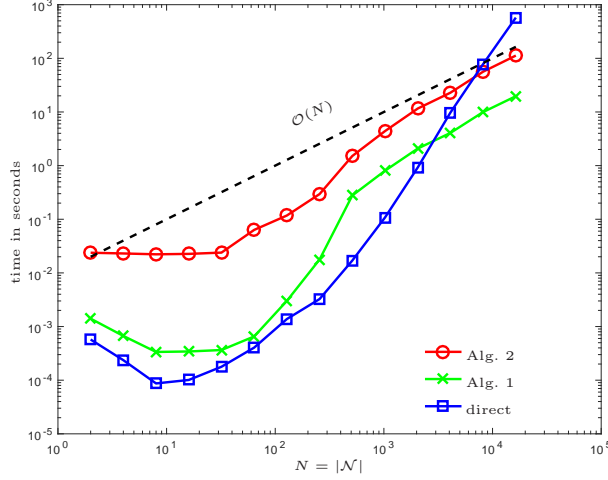
Fig. 5: Computational time in seconds versus the number of evaluation points N. The direct approach uses Matlab's `sqrtm` function.

is programmed entirely in Matlab (and thus nowhere near optimal performance), the breakthrough point at around $N = 10^3$ shows that also small problems benefit from the speed up.

5 Lemmas for the proof of Theorem 1

First, we state a slight generalization of a well-known result.

Lemma 2 *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{N \times N}$ be symmetric positive definite. Then, there holds*

$$\|\boldsymbol{A}^{1/2} - \boldsymbol{B}^{1/2}\|_2 \leq (\lambda_{\min}(\boldsymbol{A}) + \lambda_{\min}(\boldsymbol{B}))^{-1/2} \|\boldsymbol{A} - \boldsymbol{B}\|_2, \tag{16}$$

as well as

$$\|\boldsymbol{A}^{1/2} - \boldsymbol{B}^{1/2}\|_2 \leq 3 \|\boldsymbol{A} - \boldsymbol{B}\|_2^{1/2}. \tag{17}$$

Proof The estimate (16) is proved in [18, Lemma 2.2]. To obtain (17), let $\boldsymbol{U} \in \mathbb{R}^{N \times N}$ denote the orthonormal matrix that diagonalizes $\boldsymbol{A}$, i.e., $\boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U} = \boldsymbol{D}$ for a positive diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{N \times N}$. With $\boldsymbol{U} \boldsymbol{D}^{1/2} \boldsymbol{U}^T = \boldsymbol{A}^{1/2}$ and $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$, there holds for arbitrary $\alpha \geq 0$

$$\|\boldsymbol{A}^{1/2} - (\boldsymbol{A} + \alpha \boldsymbol{I})^{1/2}\|_2 = \|\boldsymbol{U}\boldsymbol{D}^{1/2}\boldsymbol{U}^T - \boldsymbol{U}(\boldsymbol{D} + \alpha\boldsymbol{I})^{1/2}\boldsymbol{U}^T\|_2$$
$$= \|\boldsymbol{D}^{1/2} - (\boldsymbol{D} + \alpha\boldsymbol{I})^{1/2}\|_2 = \max_{1 \leq i \leq N} \left| \sqrt{\boldsymbol{D}_{ii} + \alpha} - \sqrt{\boldsymbol{D}_{ii}} \right| \leq \sqrt{\alpha},$$

where we used $x + y \leq (\sqrt{x} + \sqrt{y})^2$ and hence $\sqrt{x + y} \leq \sqrt{x} + \sqrt{y}$ for $x, y \geq 0$ in the last estimate. With $\alpha := \|\boldsymbol{A} - \boldsymbol{B}\|_2$, (16) shows

$$\|(\boldsymbol{A} + \alpha \boldsymbol{I})^{1/2} - \boldsymbol{B}^{1/2}\|_2 \leq 2\alpha(\lambda_{\min}(\boldsymbol{A}) + \lambda_{\min}(\boldsymbol{B}) + \alpha)^{-1/2} \leq 2\sqrt{\alpha}.$$

The combination of the last two estimates concludes the proof of (17). $\qquad\square$

Lemma 3 *Let $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ be symmetric positive definite and assume that $0 < \kappa < 1$ and $C_\kappa > 0$ are such that the sequence of all distinct eigenvalues $\lambda_1 > \ldots > \lambda_M > 0 \in \mathbb{R}$ (for some $M \leq N$) of $\boldsymbol{M}$ satisfies $|\lambda_i - \lambda_j| \leq \lambda_1 C_\kappa \kappa^{\min\{i,j\}}$ for all $1 \leq i, j \leq M$. Given $1 \leq k \leq M$ and $\boldsymbol{z} \in \mathbb{R}^N$, define $\boldsymbol{Z} \in \mathbb{R}^{N \times k}$ by*

$$\boldsymbol{Z} := (\boldsymbol{z}, \lambda_1^{-1} \boldsymbol{M} \boldsymbol{z}, \lambda_1^{-2} \boldsymbol{M}^2 \boldsymbol{z}, \ldots, \lambda_1^{-(k-1)} \boldsymbol{M}^{k-1} \boldsymbol{z}). \tag{18}$$

Consider the QR-factorization $\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R}$, with $\boldsymbol{Q} \in \mathbb{R}^{N \times k}$ satisfying $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}_k$ and $\boldsymbol{R} \in \mathbb{R}^{k \times k}$ upper triangular with non-negative diagonal entries (note that if $\boldsymbol{Z}$ has full rank, this ensures uniqueness of $\boldsymbol{Q}$ and $\boldsymbol{R}$). Then the diagonal entries of $\boldsymbol{R}$ satisfy

$$\boldsymbol{R}_{nn} \leq |\boldsymbol{z}| C_\kappa^{n-1} \kappa^{(n-1)n/2} \quad \text{for all } 1 \leq n \leq k. \tag{19}$$

14

Proof Let $\boldsymbol{q}^i$, $i = 1, \ldots, k$ denote the orthonormal columns of $\boldsymbol{Q}$. By definition of the QR-factorization, there holds for $1 \le n \le k$

$$\lambda_1^{-(n-1)} \boldsymbol{M}^{n-1} \boldsymbol{z} = \sum_{i=1}^{n} \boldsymbol{R}_{in} \boldsymbol{q}^i.$$

Since the $\boldsymbol{q}^i$ are orthogonal, the best approximation (with respect to $|\cdot|$) of $\lambda_1^{-(n-1)} \boldsymbol{M}^{n-1} \boldsymbol{z}$ in $\mathrm{span}\{\boldsymbol{q}^1, \ldots, \boldsymbol{q}^{n-1}\}$ is given by $\sum_{i=1}^{n-1} \boldsymbol{R}_{in} \boldsymbol{q}^i$ for all $n \ge 2$. Therefore, we obtain

$$\boldsymbol{R}_{nn} = \left| \lambda_1^{-(n-1)} \boldsymbol{M}^{n-1} \boldsymbol{z} - \sum_{i=1}^{n-1} \boldsymbol{R}_{in} \boldsymbol{q}^i \right| \le \min_{\boldsymbol{v} \in \mathrm{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{n-2} \boldsymbol{z}\}} |\lambda_1^{-(n-1)} \boldsymbol{M}^{n-1} \boldsymbol{z} - \boldsymbol{v}|,$$

where we used $\mathrm{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{n-2} \boldsymbol{z}\} \subseteq \mathrm{span}\{\boldsymbol{q}^1, \ldots, \boldsymbol{q}^{n-1}\}$ by definition of the QR-factorization (see also Remark 1). We may choose $\boldsymbol{v} = p(\boldsymbol{M}) \boldsymbol{z}$, where $p(x)$ is the polynomial of degree $n - 2$ interpolating $f(x) := (x/\lambda_1)^{n-1}$ at the points $x = \lambda_1, \ldots, \lambda_{n-1}$. Since $\boldsymbol{M}$ is symmetric and positive definite, we may diagonalize it with an orthogonal matrix $\boldsymbol{U} \in \mathbb{R}^{N \times N}$, i.e., $\boldsymbol{U}^T \boldsymbol{M} \boldsymbol{U} = \boldsymbol{D}$ with a diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ containing the eigenvalues of $\boldsymbol{M}$. This allows us to conclude

$$\begin{aligned} \boldsymbol{R}_{nn} &\le \|f(\boldsymbol{M}) - p(\boldsymbol{M})\|_2 |\boldsymbol{z}| = \|\boldsymbol{U}^T (f(\boldsymbol{D}) - p(\boldsymbol{D})) \boldsymbol{U}\|_2 |\boldsymbol{z}| = \|f(\boldsymbol{D}) - p(\boldsymbol{D})\|_2 |\boldsymbol{z}| \\ &\le \max_{x \in \{\lambda_1, \ldots, \lambda_M\}} |f(x) - p(x)| |\boldsymbol{z}| = \max_{x \in \{\lambda_n, \ldots, \lambda_M\}} |f(x) - p(x)| |\boldsymbol{z}|. \end{aligned}$$

The function $f(x) - p(x)$ is a polynomial of degree $n - 1$ with known zeros $\lambda_1, \ldots, \lambda_{n-1}$ and thus reads

$$f(x) - p(x) = \alpha (x - \lambda_1) \cdots (x - \lambda_{n-1})$$

for some leading coefficient $\alpha \in \mathbb{R}$. Differentiation reveals $\alpha (n-1)! = f^{(n-1)}(x) = (n-1)! \lambda_1^{-(n-1)}$ and hence $\alpha = \lambda_1^{-(n-1)}$. This shows

$$\boldsymbol{R}_{nn} \le |\boldsymbol{z}| \max_{n \le i \le M} \prod_{j=1}^{n-1} \frac{|\lambda_i - \lambda_j|}{\lambda_1} = |\boldsymbol{z}| \prod_{j=1}^{n-1} \frac{|\lambda_M - \lambda_j|}{\lambda_1}.$$

By the decay assumption on the λ_i it follows that

$$\boldsymbol{R}_{nn} \le |\boldsymbol{z}| \prod_{j=1}^{n-1} (C_\kappa \kappa^j) = |\boldsymbol{z}| C_\kappa^{n-1} \kappa^{n(n-1)/2}. \tag{20}$$

This concludes the proof.

The next lemma shows that the matrices $\boldsymbol{Q}_j$ from Algorithm 1 are strongly tied to the matrices $\boldsymbol{Z} = \boldsymbol{Q} \boldsymbol{R}$ defined in Lemma 3.

Lemma 4 *Given $\boldsymbol{z} \in \mathbb{R}^N$ and let $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ be symmetric positive definite. Call Algorithm 1 with $\boldsymbol{M}$, $\boldsymbol{z}$, and $k \in \mathbb{N}$ to compute $k_0 \le k$ and $\boldsymbol{R}_j$, $\boldsymbol{Q}_j$ for all $1 \le j \le k_0$. Define $\boldsymbol{Z}, \boldsymbol{Q}, \boldsymbol{R}$ satisfying $\boldsymbol{Z} = \boldsymbol{Q} \boldsymbol{R}$ as in Lemma 3. Then, $\boldsymbol{Q}_j$ (as defined in Algorithm 1) for $1 \le j \le k_0$ satisfies $\boldsymbol{Q}_j = \boldsymbol{Q}|_{\{1, \ldots, N\} \times \{1, \ldots, j\}}$, i.e., the first j columns coincide and*

$$\mathrm{range}(\boldsymbol{Q}_j) = \mathrm{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{j-1} \boldsymbol{z}\} = \mathrm{range}(\boldsymbol{Z}|_{\{1, \ldots, N\} \times \{1, \ldots, j\}}) \tag{21}$$

for all $1 \le j \le k_0$. Moreover, $\boldsymbol{Z}$ has full rank if and only if $k_0 = k$.

Proof Let $\boldsymbol{q}^j$ denote the j-th column of $\boldsymbol{Q}_j$ and note that by definition of Algorithm 1 we have

$$(\boldsymbol{R}_j)_{jj} > 0 \quad \text{for all } 1 \le j \le k_0. \tag{22}$$

In order to prove (21), we first show

$$\mathrm{range}(\boldsymbol{Q}_j) = \mathrm{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{j-1} \boldsymbol{z}\} \tag{23}$$

for all $1 \leq j \leq k_0$ by induction. To that end, note that $\boldsymbol{Q}_1 = \boldsymbol{q}^1 = \boldsymbol{z}/|\boldsymbol{z}|$ and consequently (23) holds for $j = 1$. Assume (23) holds for all $1 \leq j < j_0 \leq k_0$. By construction of the matrices in Algorithm 1, we have

$$(\boldsymbol{Q}_{j_0-1}, \boldsymbol{M}\boldsymbol{q}^{j_0-1}) = \boldsymbol{Q}_{j_0}\boldsymbol{R}_{j_0}. \tag{24}$$

By the induction assumption, $\boldsymbol{q}^{j_0-1} \in \operatorname{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{j_0-2}\boldsymbol{z}\}$. Thus, (24) and the fact that $\boldsymbol{R}_{j_0}$ is regular (by (22)) imply

$$\operatorname{range}(\boldsymbol{Q}_{j_0}) = \operatorname{span}\{\operatorname{range}(\boldsymbol{Q}_{j_0-1}), \boldsymbol{M}\boldsymbol{q}^{j_0-1}\} \subseteq \operatorname{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{j_0-1}\boldsymbol{z}\}.$$

The fact that $\boldsymbol{Q}_{j_0}$ is orthogonal (and hence its range is j_0 dimensional) shows even equality, that is

$$\operatorname{range}(\boldsymbol{Q}_{j_0}) = \operatorname{span}\{\boldsymbol{z}, \ldots, \boldsymbol{M}^{j_0-1}\boldsymbol{z}\}. \tag{25}$$

This concludes the induction, and proves (23) for all $1 \leq j \leq k_0$. The second equation in (21) follows by definition of $\boldsymbol{Z}$.

To see the remainder of the statement, we first assume $k_0 = k$ and proceed to prove that $\boldsymbol{Z}$ has full rank. To that end, we apply (21) with $j = k$ to see that $\operatorname{range}(\boldsymbol{Z}) = \operatorname{range}(\boldsymbol{Q}_k)$ is k-dimensional and therefore $\boldsymbol{Z}$ has full rank.

For the converse implication, assume that $\boldsymbol{Z}$ has full rank. We prove $k_0 = k$ by induction. By construction, we have $(\boldsymbol{R}_1)_{11} = 1$ and thus $k_0 \geq 1$. Assume $k_0 \geq j_0$ for some $j_0 < k$. Then, since $(\boldsymbol{R}_j)_{jj} \neq 0$ for all $1 \leq j < j_0$, the identity (21) shows $\operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j\}}) = \operatorname{range}(\boldsymbol{Q}_j)$ for all $j < j_0$. From this, we argue that

$$\boldsymbol{q}^{j_0-1} \in \operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j_0-1\}}) \setminus \operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j_0-2\}}),$$

which, by definition of $\boldsymbol{Z} = (\boldsymbol{z}, \lambda_1^{-1}\boldsymbol{M}\boldsymbol{z}, \ldots, \lambda_1^{-(k-1)}\boldsymbol{M}^{k-1}\boldsymbol{z})$, shows that $\boldsymbol{q}^{j_0-1} = \sum_{i=0}^{j_0-2} \alpha_i \boldsymbol{M}^i \boldsymbol{z}$ for some $\alpha_i \in \mathbb{R}$ with $\alpha_{j_0-2} \neq 0$. Consequently, we obtain $\boldsymbol{M}\boldsymbol{q}^{j_0-1} = \sum_{i=0}^{j_0-2} \alpha_i \boldsymbol{M}^{i+1}\boldsymbol{z} \in \operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j_0\}}) \setminus \operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j_0-1\}})$. Since $\operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j_0-1\}}) = \operatorname{range}(\boldsymbol{Q}_{j_0-1})$, this implies the identity $\operatorname{range}((\boldsymbol{Q}_{j_0-1}, \boldsymbol{M}\boldsymbol{q}^{j_0-1})) = \operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,j_0\}})$ and therefore the matrix $(\boldsymbol{Q}_{j_0-1}, \boldsymbol{M}\boldsymbol{q}^{j_0-1})$ has full rank. Hence, (24) implies that $\boldsymbol{R}_{j_0}$ has full rank, which in particular implies $(\boldsymbol{R}_{j_0})_{j_0 j_0} \neq 0$ and thus $k_0 \geq j_0 + 1$. This concludes the induction and shows $k_0 = k$.

The following result proves that if Algorithm 1 terminates in less than k steps (due to the criterion in step 1(c)), the quantity $\boldsymbol{M}^{1/2}\boldsymbol{z}$ is computed exactly.

Lemma 5 *Let $\boldsymbol{z} \in \mathbb{R}^N$ and let $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ be symmetric positive definite. Call Algorithm 1 with $\boldsymbol{M}$, $\boldsymbol{z}$, and $k \in \mathbb{N}$ to compute $k_0 \leq k$ as well as $\boldsymbol{Q}_j$ for all $1 \leq j \leq k_0$. Define $\boldsymbol{U}_{k_0} = \boldsymbol{Q}_{k_0}^T \boldsymbol{M} \boldsymbol{Q}_{k_0}$ as in Algorithm 1. If $k_0 < k$, there holds*

$$\boldsymbol{M}^{1/2}\boldsymbol{z} = \boldsymbol{Q}_{k_0}\boldsymbol{U}_{k_0}^{1/2}\boldsymbol{Q}_{k_0}^T\boldsymbol{z}.$$

Proof If $k_0 < k$ then Lemma 4 shows that $\boldsymbol{Z}$ as defined in Lemma 3 does not have full rank. Moreover, the identity (21) shows that $\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,k_0\}}$ has full rank. By definition of $\boldsymbol{Z}$, this implies $\operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,k_0\}}) = \operatorname{range}(\boldsymbol{Z})$. Therefore, (21) shows

$$\begin{aligned}
\operatorname{range}(\boldsymbol{M}\boldsymbol{Q}_{k_0}) &= \operatorname{range}(\boldsymbol{M}\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,k_0\}}) \\
&\subseteq \operatorname{range}(\boldsymbol{Z}) = \operatorname{range}(\boldsymbol{Z}|_{\{1,\ldots,N\}\times\{1,\ldots,k_0\}}) = \operatorname{range}(\boldsymbol{Q}_{k_0}).
\end{aligned} \tag{26}$$

Let $\overline{\boldsymbol{Q}} \in \mathbb{R}^{N\times N}$ be an orthonormal matrix such that its first k_0 columns coincide with $\boldsymbol{Q}_{k_0}$, i.e., $\overline{\boldsymbol{Q}} = (\boldsymbol{Q}_{k_0}, \boldsymbol{Q}_\perp)$ for some orthonormal $\boldsymbol{Q}_\perp \in \mathbb{R}^{N\times(N-k_0)}$. We obtain

$$\boldsymbol{M}^{1/2} = \overline{\boldsymbol{Q}}\,\overline{\boldsymbol{Q}}^T \boldsymbol{M}^{1/2}\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{Q}}^T = \overline{\boldsymbol{Q}}\,(\overline{\boldsymbol{Q}}^T \boldsymbol{M}\overline{\boldsymbol{Q}})^{1/2}\,\overline{\boldsymbol{Q}}^T. \tag{27}$$

There holds

$$\overline{\boldsymbol{Q}}^T \boldsymbol{M}\overline{\boldsymbol{Q}} = \begin{pmatrix} \boldsymbol{Q}_{k_0}^T \boldsymbol{M}\boldsymbol{Q}_{k_0} & \boldsymbol{Q}_{k_0}^T \boldsymbol{M}\boldsymbol{Q}_\perp \\ \boldsymbol{Q}_\perp^T \boldsymbol{M}\boldsymbol{Q}_{k_0} & \boldsymbol{Q}_\perp^T \boldsymbol{M}\boldsymbol{Q}_\perp \end{pmatrix}.$$

The invariance property (26) shows $\boldsymbol{Q}_\perp^T \boldsymbol{M} \boldsymbol{Q}_{k_0} = \boldsymbol{0}$, and by symmetry also $\boldsymbol{Q}_{k_0}^T \boldsymbol{M} \boldsymbol{Q}_\perp = \boldsymbol{0}$. Therefore, we have

$$\overline{(\boldsymbol{Q}^T \boldsymbol{M} \overline{\boldsymbol{Q}})^{1/2}} = \begin{pmatrix} \boldsymbol{U}_{k_0}^{1/2} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{Q}_\perp^T \boldsymbol{M} \boldsymbol{Q}_\perp)^{1/2} \end{pmatrix}.$$

This and (27), together with $\boldsymbol{z} \in \mathrm{range}(\boldsymbol{Q}_{k_0})$, show $\boldsymbol{M}^{1/2} \boldsymbol{z} = \overline{\boldsymbol{Q}}\,(\overline{\boldsymbol{Q}}^T \boldsymbol{M} \overline{\boldsymbol{Q}})^{1/2}\,\overline{\boldsymbol{Q}}^T \boldsymbol{z} = \boldsymbol{Q}_{k_0} \boldsymbol{U}_{k_0}^{1/2} \boldsymbol{Q}_{k_0}^T \boldsymbol{z}$ and conclude the proof.

The following result is the main tool to prove Theorem 1 (i).

Lemma 6 *Let $\boldsymbol{z} \in \mathbb{R}^N$ and let $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ be symmetric positive definite. Call Algorithm 1 with $\boldsymbol{M}$, $\boldsymbol{z}$, and $k \in \mathbb{N}$ to compute $k_0 \le k$ as well as $\boldsymbol{Q}_j$ for all $1 \le j \le k_0$. Let $\boldsymbol{U}_{k_0} = \boldsymbol{Q}_{k_0}^T \boldsymbol{M} \boldsymbol{Q}_{k_0}$ be defined as in Algorithm 1. Then, there holds*

$$\frac{|\boldsymbol{M}^{1/2} \boldsymbol{z} - \boldsymbol{Q}_{k_0} \boldsymbol{U}_{k_0}^{1/2} \boldsymbol{Q}_{k_0}^T \boldsymbol{z}|}{|\boldsymbol{z}|} \le \begin{cases} \sqrt{2\|\boldsymbol{M}\|_2}\, \dfrac{4r^2}{r-1} r^{-k} & \text{if } k_0 = k, \\ 0 & \text{if } k_0 < k, \end{cases}$$

where

$$r := \frac{\lambda_{\max}(\boldsymbol{M}) + \lambda_{\min}(\boldsymbol{M})}{\lambda_{\max}(\boldsymbol{M}) - \lambda_{\min}(\boldsymbol{M})} > 1. \tag{28}$$

Proof The case $k_0 < k$ is covered in Lemma 5. Assume $k_0 = k$. Note that $\boldsymbol{Q}_k \boldsymbol{Q}_k^T$ is the identity on $\mathrm{range}(\boldsymbol{Q}_k)$. Lemma 4 shows that $\boldsymbol{M}^j \boldsymbol{z} \in \mathrm{range}(\boldsymbol{Q}_k)$ for all $0 \le j \le k-1$. Moreover, $\boldsymbol{z} \in \mathrm{range}(\boldsymbol{Q}_k)$ by construction. Hence, we have

$$\boldsymbol{M}^j \boldsymbol{z} = \boldsymbol{Q}_k \boldsymbol{Q}_k^T \boldsymbol{M}^j \boldsymbol{z} = \boldsymbol{Q}_k \boldsymbol{Q}_k^T \boldsymbol{M}^j \boldsymbol{Q}_k \boldsymbol{Q}_k^T \boldsymbol{z} = \boldsymbol{Q}_k (\boldsymbol{Q}_k^T \boldsymbol{M} \boldsymbol{Q}_k)^j \boldsymbol{Q}_k^T \boldsymbol{z} \quad \text{for all } 1 \le j \le k-1.$$

Thus, any polynomial $p \in \mathcal{P}^{k-1}$ of degree $k-1$ satisfies

$$p(\boldsymbol{M}) \boldsymbol{z} = \boldsymbol{Q}_k \boldsymbol{Q}_k^T p(\boldsymbol{M}) \boldsymbol{Q}_k \boldsymbol{Q}_k^T \boldsymbol{z} = \boldsymbol{Q}_k p(\boldsymbol{Q}_k^T \boldsymbol{M} \boldsymbol{Q}_k) \boldsymbol{Q}_k^T \boldsymbol{z} = \boldsymbol{Q}_k p(\boldsymbol{U}_k) \boldsymbol{Q}_k^T \boldsymbol{z}.$$

This implies for all $p \in \mathcal{P}^{k-1}$

$$\begin{aligned} |\boldsymbol{M}^{1/2} \boldsymbol{z} &- \boldsymbol{Q}_k \boldsymbol{U}_k^{1/2} \boldsymbol{Q}_k^T z| \\ &\le |\boldsymbol{M}^{1/2} \boldsymbol{z} - \boldsymbol{Q}_k p(\boldsymbol{U}_k) \boldsymbol{Q}_k^T z| + |\boldsymbol{Q}_k p(\boldsymbol{U}_k) \boldsymbol{Q}_k^T z - \boldsymbol{Q}_k \boldsymbol{U}_k^{1/2} \boldsymbol{Q}_k^T z| \\ &\le |\boldsymbol{M}^{1/2} \boldsymbol{z} - p(\boldsymbol{M}) \boldsymbol{z}| + |\boldsymbol{Q}_k (p(\boldsymbol{U}_k) - \boldsymbol{U}_k^{1/2}) \boldsymbol{Q}_k^T \boldsymbol{z}| \\ &\le \big(\|\boldsymbol{M}^{1/2} - p(\boldsymbol{M})\|_2 + \|p(\boldsymbol{U}_k) - \boldsymbol{U}_k^{1/2}\|_2\big)|\boldsymbol{z}|. \end{aligned} \tag{29}$$

With $f(x) := \sqrt{(x+1)(\lambda_{\max}(\boldsymbol{M}) - \lambda_{\min}(\boldsymbol{M}))/2 + \lambda_{\min}(\boldsymbol{M})}$, the result [2, Lemma 4.14] proves

$$\min_{p \in \mathcal{P}^{k-1}} \|f - p\|_{L^\infty([-1,1])} \le \frac{2r^2}{r-1} r^{-k} \sup_{x \in \mathbb{C}_r} |f(x)|$$

with $r > 1$ from (28) and

$$\mathbb{C}_r := \Big\{ x \in \mathbb{C} \,:\, \Big(\frac{2\,\mathrm{real}(x)}{r + 1/r}\Big)^2 + \Big(\frac{2\,\mathrm{imag}(x)}{r - 1/r}\Big)^2 \le 1 \Big\}.$$

Since $x \in \mathbb{C}_r$ implies $|x| \le r$, straightforward calculations show

$$\begin{aligned} \sup_{x \in \mathbb{C}_r} |f(x)| &\le \sup_{|x| \le r} |f(x)| = \sup_{|x| \le r} \sqrt{|(x+1)(\lambda_{\max}(\boldsymbol{M}) - \lambda_{\min}(\boldsymbol{M}))/2 + \lambda_{\min}(\boldsymbol{M})|} \\ &\le \sup_{|x| \le r} \sqrt{(|x| + 1)(\lambda_{\max}(\boldsymbol{M}) - \lambda_{\min}(\boldsymbol{M}))/2 + \lambda_{\min}(\boldsymbol{M})} \\ &\le \sqrt{\lambda_{\max}(\boldsymbol{M}) + \lambda_{\min}(\boldsymbol{M})} \le \sqrt{2\|\boldsymbol{M}\|_2}, \end{aligned}$$

which implies the estimate $\min_{p \in \mathcal{P}^{k-1}} \|f - p\|_{L^\infty([-1,1])} \leq \sqrt{2\|\boldsymbol{M}\|_2} \frac{2r^2}{r-1} r^{-k}$. Hence, we obtain for $g(x) :=$ $\sqrt{x}$ (note that $x \mapsto (x+1)(\lambda_{\max}(\boldsymbol{M}) - \lambda_{\min}(\boldsymbol{M}))/2 + \lambda_{\min}(\boldsymbol{M})$ maps $[-1,1]$ onto $[\lambda_{\min}(\boldsymbol{M}), \lambda_{\max}(\boldsymbol{M})]$) also

$$\min_{p \in \mathcal{P}^{k-1}} \|g - p\|_{L^\infty([\lambda_{\min}(\boldsymbol{M}), \lambda_{\max}(\boldsymbol{M})])} \leq \sqrt{2\|\boldsymbol{M}\|_2} \frac{2r^2}{r-1} r^{-k}. \tag{30}$$

Let $\boldsymbol{U} \in \mathbb{R}^{N \times N}$ denote the orthonormal matrix ($\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$) that diagonalizes $\boldsymbol{M}$, i.e., $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ with $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ being the diagonal matrix containing the eigenvalues of $\boldsymbol{M}$. There holds $\boldsymbol{M}^{1/2} = \boldsymbol{U}\boldsymbol{D}^{1/2}\boldsymbol{U}^T$ as well as $p(\boldsymbol{M}) = \boldsymbol{U}p(\boldsymbol{D})\boldsymbol{U}^T$. This, (30), and invariance of the spectral norm $\|\cdot\|_2 = \|\boldsymbol{U}(\cdot)\boldsymbol{U}^T\|_2$ show

$$\min_{p \in \mathcal{P}^{k-1}} \|\boldsymbol{M}^{1/2} - p(\boldsymbol{M})\|_2 = \min_{p \in \mathcal{P}^{k-1}} \|\boldsymbol{D}^{1/2} - p(\boldsymbol{D})\|_2$$

$$= \min_{p \in \mathcal{P}^{k-1}} \max_{1 \leq i \leq N} |g(\boldsymbol{D}_{ii}) - p(\boldsymbol{D}_{ii})| \leq \sqrt{2\|\boldsymbol{M}\|_2} \frac{2r^2}{r-1} r^{-k}.$$

Since $\boldsymbol{U}_k$ is an orthogonal projection of $\boldsymbol{M}$, we have $\lambda_{\min}(\boldsymbol{M}) \leq \lambda_{\min}(\boldsymbol{U}_k) \leq \lambda_{\max}(\boldsymbol{U}_k) \leq \lambda_{\max}(\boldsymbol{M})$. Thus, repeating the above argument for $\boldsymbol{U}_k$ instead of $\boldsymbol{M}$ yields

$$\min_{p \in \mathcal{P}^{k-1}} \left(\|\boldsymbol{M}^{1/2} - p(\boldsymbol{M})\|_2 + \|p(\boldsymbol{U}_k) - \boldsymbol{U}_k^{1/2}\|_2 \right) \leq \sqrt{2\|\boldsymbol{M}\|_2} \frac{4r^2}{r-1} r^{-k}.$$

This in combination with (29) and Lemma 4 conclude the proof. $\qquad\square$

The next result quantifies the distance of $\text{range}(\boldsymbol{Q}_j)$ to $\text{range}(\boldsymbol{M}\boldsymbol{Q}_j)$ in terms of the projection $\boldsymbol{Q}_j\boldsymbol{Q}_j^T$ onto $\text{range}(\boldsymbol{Q}_j)$.

Lemma 7 *Assume the requirements of Lemma 3. Call Algorithm 1 with $\boldsymbol{M}$, $\boldsymbol{z}$, and $k \in \mathbb{N}$ to compute $k_0 \leq k$ as well as $\boldsymbol{Q}_j$ for all $1 \leq j \leq k_0$. Let $\boldsymbol{q}^j$ be the last column of $\boldsymbol{Q}_j$ for all $1 \leq j \leq k_0$. There holds for all $1 \leq j < k_0$*

$$\|\boldsymbol{M}\boldsymbol{Q}_j - \boldsymbol{Q}_j\boldsymbol{Q}_j^T\boldsymbol{M}\boldsymbol{Q}_j\|_2 = |(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{q}^j| \tag{31}$$

as well as

$$\min_{1 \leq i \leq j} \|\boldsymbol{M}\boldsymbol{Q}_i - \boldsymbol{Q}_i\boldsymbol{Q}_i^T\boldsymbol{M}\boldsymbol{Q}_i\|_2 \leq \lambda_{\max}(\boldsymbol{M}) C_\kappa \kappa^{(j+1)/2}.$$

Proof Recall $\boldsymbol{Z}, \boldsymbol{Q}, \boldsymbol{R}$ satisfying $\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R}$ from Lemma 3 with k replaced by k_0 in the call to Algorithm 1. By Lemma 4, $\boldsymbol{q}^j$ coincides with the j-th column of $\boldsymbol{Q}$ for all $1 \leq j \leq k_0$. Moreover, let $\boldsymbol{r}^j$ be the j-th column of $\boldsymbol{R}$ and let $\widetilde{\boldsymbol{r}}^j$ be the j-th column of $\boldsymbol{R}^{-1}$ from Lemma 3. All quantities are well-defined since $\boldsymbol{Z}$ has maximal rank k_0 by Lemma 4. For the first statement (31), note that $\text{range}(\boldsymbol{M}\boldsymbol{Q}_j) \subseteq \text{range}(\boldsymbol{Q}_{j+1})$ implies $\boldsymbol{M}\boldsymbol{Q}_j = \boldsymbol{Q}_{j+1}\boldsymbol{Q}_{j+1}^T\boldsymbol{M}\boldsymbol{Q}_j$. Moreover, due to Lemma 4, we have $\boldsymbol{Q}_{j+1} = (\boldsymbol{Q}_j, \boldsymbol{q}^{j+1})$, and hence $\boldsymbol{Q}_{j+1}\boldsymbol{Q}_{j+1}^T = \boldsymbol{q}^{j+1}(\boldsymbol{q}^{j+1})^T + \boldsymbol{Q}_j\boldsymbol{Q}_j^T$. Altogether, this shows

$$\|\boldsymbol{M}\boldsymbol{Q}_j - \boldsymbol{Q}_j\boldsymbol{Q}_j^T\boldsymbol{M}\boldsymbol{Q}_j\|_2 = \|\boldsymbol{Q}_{j+1}\boldsymbol{Q}_{j+1}^T\boldsymbol{M}\boldsymbol{Q}_j - \boldsymbol{Q}_j\boldsymbol{Q}_j^T\boldsymbol{M}\boldsymbol{Q}_j\|_2$$

$$= \|(\boldsymbol{q}^{j+1}(\boldsymbol{q}^{j+1})^T + \boldsymbol{Q}_j\boldsymbol{Q}_j^T)\boldsymbol{M}\boldsymbol{Q}_j - \boldsymbol{Q}_j\boldsymbol{Q}_j^T\boldsymbol{M}\boldsymbol{Q}_j\|_2$$

$$= \|\boldsymbol{q}^{j+1}(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{Q}_j\|_2 = |(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{q}^j|,$$

where the last step follows because $\boldsymbol{q}^{j+1}$ is orthogonal to $\boldsymbol{M}\boldsymbol{q}^i$, $i = 1, \ldots, j-1$. This proves (31).

To see the remaining statement, note that the definition of $\boldsymbol{Z}$ in (18) implies

$$(\boldsymbol{M}\boldsymbol{Z})|_{\{1,\ldots,N\} \times \{j\}} = \lambda_1 \boldsymbol{Z}|_{\{1,\ldots,N\} \times \{j+1\}} = \lambda_1(\boldsymbol{Q}\boldsymbol{R})|_{\{1,\ldots,N\} \times \{j+1\}} = \lambda_1\boldsymbol{Q}\boldsymbol{r}^{j+1}$$

as well as

$$\boldsymbol{q}^j = (\boldsymbol{Z}\boldsymbol{R}^{-1})|_{\{1,\ldots,N\} \times \{j\}} = \boldsymbol{Z}\widetilde{\boldsymbol{r}}^j.$$

The last two identities, and the fact that $(\boldsymbol{q}^{j+1})^T(\boldsymbol{MZ})|_{\{1,\dots,N\}\times\{i\}} = 0$ for all $1 \le i \le j-1$, imply

$$
\begin{aligned}
(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{q}^j &= (\boldsymbol{q}^{j+1})^T\boldsymbol{MZ}\widetilde{\boldsymbol{r}}^j = (\boldsymbol{q}^{j+1})^T(\boldsymbol{MZ})|_{\{1,\dots,N\}\times\{j\}}(\widetilde{\boldsymbol{r}}^j)_j \\
&= \lambda_1(\boldsymbol{q}^{j+1})^T\boldsymbol{Q}\boldsymbol{r}^{j+1}(\widetilde{\boldsymbol{r}}^j)_j = \lambda_1(\boldsymbol{r}^{j+1})_{j+1}(\widetilde{\boldsymbol{r}}^j)_j.
\end{aligned}
$$

The triangular structure of $\boldsymbol{R}$ implies $(\boldsymbol{R}^{-1})_{jj} = 1/\boldsymbol{R}_{jj}$ and hence $(\widetilde{\boldsymbol{r}}^j)_j = 1/\boldsymbol{R}_{jj}$ (where $\boldsymbol{R}_{jj} \ne 0$ by assumption). This shows

$$
(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{q}^j = \lambda_1 \frac{\boldsymbol{R}_{(j+1)(j+1)}}{\boldsymbol{R}_{jj}}. \tag{32}
$$

With Lemma 3, we have

$$
\frac{\boldsymbol{R}_{(j+1)(j+1)}}{\boldsymbol{R}_{jj}}\frac{\boldsymbol{R}_{jj}}{\boldsymbol{R}_{(j-1)(j-1)}}\cdots\frac{\boldsymbol{R}_{22}}{\boldsymbol{R}_{11}}\boldsymbol{R}_{11} = \boldsymbol{R}_{(j+1)(j+1)} \le |\boldsymbol{z}|C_\kappa^j\kappa^{(j+1)j/2}. \tag{33}
$$

Moreover, we know $\boldsymbol{R}_{11} = |\boldsymbol{q}^1\boldsymbol{R}_{11}| = |\boldsymbol{z}|$. This implies that at least one of the fractions on the left-hand side of (33) must be smaller than the j-th root of the right hand side of (33) divided by $|\boldsymbol{z}|$ and hence

$$
\min_{1\le i\le j}\frac{\boldsymbol{R}_{(i+1)(i+1)}}{\boldsymbol{R}_{ii}} \le C_\kappa\kappa^{(j+1)/2}.
$$

With this, (32), and (31), we obtain

$$
\min_{1\le i\le j}\|\boldsymbol{M}\boldsymbol{Q}_i - \boldsymbol{Q}_i\boldsymbol{Q}_i^T\boldsymbol{M}\boldsymbol{Q}_i\|_2 \le \lambda_1 C_\kappa\kappa^{(j+1)/2}.
$$

This concludes the proof.

The following proposition is the main tool to prove Theorem 1 (ii).

Proposition 2 *Let $\boldsymbol{z} \in \mathbb{R}^N$ and let $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ be symmetric positive definite. Call Algorithm 1 with $\boldsymbol{M}$, $\boldsymbol{z}$, and $k \in \mathbb{N}$ to compute $k_0 \le k$ as well as $\boldsymbol{Q}_j$ for all $1 \le j \le k_0$. Then, $\boldsymbol{U}_j := \boldsymbol{Q}_j^T\boldsymbol{M}\boldsymbol{Q}_j$ satisfies the error bound*

$$
\frac{|\boldsymbol{M}^{1/2}\boldsymbol{z} - \boldsymbol{Q}_j\boldsymbol{U}_j^{1/2}\boldsymbol{Q}_j^T\boldsymbol{z}|}{|\boldsymbol{z}|} \le \begin{cases} \min\left\{\dfrac{|(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{q}^j|}{\sqrt{\lambda_{\min}(\boldsymbol{M})}}, 3\sqrt{|(\boldsymbol{q}^{j+1})^T\boldsymbol{M}\boldsymbol{q}^j|}\right\} & 1 \le j < k_0, \\ 0 & j = k_0 \text{ and } k_0 < k \end{cases}
$$

and we have the a priori estimate

$$
\min_{1\le i\le j}|(\boldsymbol{q}^{i+1})^T\boldsymbol{M}\boldsymbol{q}^i| \le \lambda_1 C_\kappa\kappa^{(j+1)/2}
$$

for all $1 \le j < k_0$.

Proof The case $k_0 < k$ and $j = k_0$ is trivially covered in Lemma 5. For the other cases, let $\overline{\boldsymbol{Q}} \in \mathbb{R}^{N\times N}$ be orthonormal such that the first j columns coincide with $\boldsymbol{Q}_j$, i.e., $\overline{\boldsymbol{Q}} = (\boldsymbol{Q}_j, \boldsymbol{Q}_\perp)$ for some orthonormal $\boldsymbol{Q}_\perp \in \mathbb{R}^{N\times(N-j)}$. Then, we write

$$
\overline{\boldsymbol{Q}}^T\boldsymbol{M}\overline{\boldsymbol{Q}} = \begin{pmatrix} \boldsymbol{U}_j & \boldsymbol{S}^T \\ \boldsymbol{S} & \boldsymbol{T} \end{pmatrix}
$$

for matrices $\boldsymbol{S} = \boldsymbol{Q}_\perp^T\boldsymbol{M}\boldsymbol{Q}_j \in \mathbb{R}^{(N-j)\times j}$, $\boldsymbol{T} \in \mathbb{R}^{(N-j)\times(N-j)}$. This means that

$$
\left\|\overline{\boldsymbol{Q}}^T\boldsymbol{M}\overline{\boldsymbol{Q}} - \begin{pmatrix} \boldsymbol{U}_j & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T} \end{pmatrix}\right\|_2 \le \|\boldsymbol{S}\|_2.
$$

Lemma 2 then implies

$$
\left\|(\overline{\boldsymbol{Q}}^T\boldsymbol{M}\overline{\boldsymbol{Q}})^{1/2} - \begin{pmatrix} \boldsymbol{U}_j^{1/2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^{1/2} \end{pmatrix}\right\|_2 \le \min\left\{\lambda_{\min}(\boldsymbol{M})^{-1/2}\|\boldsymbol{S}\|_2, 3\sqrt{\|\boldsymbol{S}\|_2}\right\}. \tag{34}
$$

Since $\boldsymbol{I} - \boldsymbol{Q}_j\boldsymbol{Q}_j^T = \boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^T$, we have

$$\|\boldsymbol{S}\|_2 = \|\boldsymbol{Q}_\perp^T\boldsymbol{M}\boldsymbol{Q}_j\|_2 = \|\boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^T\boldsymbol{M}\boldsymbol{Q}_j\|_2 = \|\boldsymbol{M}\boldsymbol{Q}_j - \boldsymbol{Q}_j\boldsymbol{Q}_j^T\boldsymbol{M}\boldsymbol{Q}_j\|_2.$$

With $(\overline{\boldsymbol{Q}}^T\boldsymbol{M}\overline{\boldsymbol{Q}})^{1/2} = \overline{\boldsymbol{Q}}^T\boldsymbol{M}^{1/2}\overline{\boldsymbol{Q}}$ and since the ranges of $\boldsymbol{Q}_j$ and $\boldsymbol{Q}_\perp$ are orthogonal, we have $\boldsymbol{Q}_\perp^T\boldsymbol{Q}_j\boldsymbol{Q}_j^T = \boldsymbol{0}$ and

$$
\begin{aligned}
\|\boldsymbol{M}^{1/2}&\boldsymbol{Q}_j\boldsymbol{Q}_j^T - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T\|_2 \\
&= \|\boldsymbol{M}^{1/2}\boldsymbol{Q}_j\boldsymbol{Q}_j^T - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T\boldsymbol{Q}_j\boldsymbol{Q}_j^T - \boldsymbol{Q}_\perp(\boldsymbol{T}^{1/2})\boldsymbol{Q}_\perp^T\boldsymbol{Q}_j\boldsymbol{Q}_j^T\|_2 \\
&\leq \|\boldsymbol{M}^{1/2} - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T - \boldsymbol{Q}_\perp(\boldsymbol{T}^{1/2})\boldsymbol{Q}_\perp^T\|_2 \\
&= \|\overline{\boldsymbol{Q}}^T\left(\boldsymbol{M}^{1/2} - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T - \boldsymbol{Q}_\perp(\boldsymbol{T}^{1/2})\boldsymbol{Q}_\perp^T\right)\overline{\boldsymbol{Q}}\|_2 \\
&= \left\|(\overline{\boldsymbol{Q}}^T\boldsymbol{M}\overline{\boldsymbol{Q}})^{1/2} - \begin{pmatrix} \boldsymbol{U}_j^{1/2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}^{1/2} \end{pmatrix}\right\|_2.
\end{aligned}
\tag{35}
$$

The combination of (34) and (35) shows

$$\|\boldsymbol{M}^{1/2}\boldsymbol{Q}_j\boldsymbol{Q}_j^T - \boldsymbol{Q}_j(\boldsymbol{U}_j^{1/2})\boldsymbol{Q}_j^T\|_2 \leq \min\left\{\lambda_{\min}(\boldsymbol{M})^{-1/2}\|\boldsymbol{S}\|_2, 3\sqrt{\|\boldsymbol{S}\|_2}\right\}.$$

We conclude the proof with $\boldsymbol{z} = \boldsymbol{Q}_j\boldsymbol{Q}_j^T\boldsymbol{z}$ due to $\boldsymbol{z} \in \mathrm{range}(\boldsymbol{Q}_j)$ and Lemma 7.

6 Lemma for the proof of Theorem 2

The following lemma is the main tool for the proof of Theorem 2.

Lemma 8 *Let $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ be symmetric positive definite. Then, the iteration (13) with initial values $\boldsymbol{A}_0 = s\boldsymbol{M}$ and $\boldsymbol{B}_0 = \boldsymbol{I}$ satisfies*

$$\|\boldsymbol{M}^{1/2} - s^{-1/2}\boldsymbol{A}_k\|_2 \leq s^{-1/2}\left(\max\{|1 - s\lambda_{\max}(\boldsymbol{M})|, |1 - s\lambda_{\min}(\boldsymbol{M})|\}\right)^{2^k} \tag{36}$$

for all $k \in \mathbb{N}$ and all $s > 0$. The minimum bound is attained at $s = 2/(\lambda_{\min}(\boldsymbol{M}) + \lambda_{\max}(\boldsymbol{M}))$ such that $\max\{|1 - s\lambda_{\max}(\boldsymbol{M})|, |1 - s\lambda_{\min}(\boldsymbol{M})|\} = 1 - 2\lambda_{\min}(\boldsymbol{M})/(\lambda_{\min}(\boldsymbol{M}) + \lambda_{\max}(\boldsymbol{M}))$.

Proof Straightforward calculations show

$$\max\left\{\|\boldsymbol{M}^{1/2} - \boldsymbol{A}_k\|_2, \|\boldsymbol{M}^{-1/2} - \boldsymbol{B}_k\|_2\right\} = \left\|\begin{pmatrix} 0 & \boldsymbol{M}^{1/2} \\ \boldsymbol{M}^{-1/2} & 0 \end{pmatrix} - \begin{pmatrix} 0 & \boldsymbol{A}_k \\ \boldsymbol{B}_k & 0 \end{pmatrix}\right\|_2.$$

The result [15, Theorem 5.2] shows $\|\boldsymbol{I} - \boldsymbol{X}_n^2\|_2 < \|\boldsymbol{I} - \boldsymbol{X}_0^2\|_2^{(e_1+e_2+1)^n}$ for all $n \in \mathbb{N}$, where $\boldsymbol{X}_{n+1} = -\boldsymbol{X}_n P_{e_1 e_2}(\boldsymbol{I} - \boldsymbol{X}_n^2)Q_{e_1 e_2}^{-1}(\boldsymbol{I} - \boldsymbol{X}_n^2)$ and $\boldsymbol{X}_0$ has no purely imaginary eigenvalues. Here $P_{e_1 e_2}/Q_{e_1 e_2}$ is the (e_1/e_2)-Padé approximant to $(1-x)^{-1/2}$. We obtain from [15, Table 1] that for $e_1 = 1$ and $e_2 = 0$, $\boldsymbol{X}_n$ satisfies the Schultz iteration (12) and thus we may use the result with

$$\boldsymbol{X}_0 = \begin{pmatrix} 0 & \boldsymbol{M} \\ \boldsymbol{I} & 0 \end{pmatrix}$$

to show

$$\left\|\begin{pmatrix} 0 & \boldsymbol{M}^{1/2} \\ \boldsymbol{M}^{-1/2} & 0 \end{pmatrix} - \begin{pmatrix} 0 & \boldsymbol{A}_k \\ \boldsymbol{B}_k & 0 \end{pmatrix}\right\|_2 < \left\|\begin{pmatrix} \boldsymbol{I} - \boldsymbol{M} & 0 \\ 0 & \boldsymbol{I} - \boldsymbol{M} \end{pmatrix}\right\|_2^{2^k} = \|\boldsymbol{I} - \boldsymbol{M}\|_2^{2^k}$$

for all $k \in \mathbb{N}$. By scaling of $\boldsymbol{M}$, we may minimize the right-hand side. To that end, we observe that the spectrum satisfies $\sigma(\boldsymbol{I} - s\boldsymbol{M}) \subset [1 - s\lambda_{\max}(\boldsymbol{M}), 1 - s\lambda_{\min}(\boldsymbol{M})]$. The fact $\|\boldsymbol{I} - s\boldsymbol{M}\|_2 \leq \max\{|1 - s\lambda_{\max}(\boldsymbol{M})|, |1 - s\lambda_{\min}(\boldsymbol{M})|\}$ proves (36). A straightforward optimization of $s > 0$ concludes the proof.
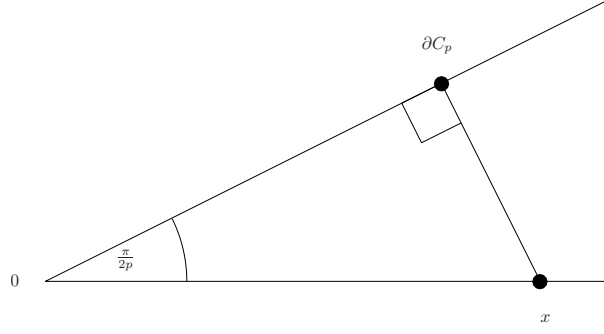
Fig. 6: The situation of the proof of Lemma 10. The distance between ∂C_p and x is $x\sin(\pi/(2p))$.

A Proof of Lemma 1

The following lemma is an elementary statement on holomorphic functions

Lemma 9 *Let $f\colon O \to \mathbb{C}$ be a continuous function on the domain $O \subset \mathbb{C}^n$ which is holomorphic in O in all variables $\boldsymbol{x}_i$, $i \in \{1,\ldots,n\}$, i.e.,*

$$\boldsymbol{x}_i \mapsto f(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i,\ldots,\boldsymbol{x}_n)$$

is holomorphic in $\{\boldsymbol{x}_i \in \mathbb{C} : (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i,\ldots,\boldsymbol{x}_n) \in O\}$ for all $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},\ldots,\boldsymbol{x}_n \in \mathbb{C}$. Then, for all multi-indices $\alpha \in \mathbb{N}_0^n$, the function $\partial_{\boldsymbol{x}}^{\alpha} f$ is holomorphic in O in all variables $\boldsymbol{x}_i$, $i \in \{1,\ldots,n\}$ as defined above.

Proof The result is proved by induction on $|\alpha|_1$. Obviously, for $|\alpha|_1 = 0$, $\partial_{\boldsymbol{x}}^{\alpha} f = f$ and the statement is true. Assume the statement holds for all $|\alpha|_1 \le k$ and choose some $\alpha \in \mathbb{N}_0^n$ with $|\alpha|_1 = k+1$. Then, we have for some $i \in \{1,\ldots,n\}$ and some $\alpha_0 \in \mathbb{N}_0^n$ with $|\alpha_0|_1 = k$ that

$$\partial_{\boldsymbol{x}}^{\alpha} f = \partial_{\boldsymbol{x}_i} \partial_{\boldsymbol{x}}^{\alpha_0} f.$$

Since, $\partial_{\boldsymbol{x}}^{\alpha_0} f$ is holomorphic in O in all variables by the induction hypothesis, obviously $\partial_{\boldsymbol{x}}^{\alpha} f$ is holomorphic in O at least in $\boldsymbol{x}_i$ (derivatives of holomorphic functions are holomorphic). To prove the statement for all other variables, we may employ Cauchy's integral formula to obtain

$$\partial_{\boldsymbol{x}}^{\alpha} f(\boldsymbol{x}) = \partial_{\boldsymbol{x}_i} \partial_{\boldsymbol{x}}^{\alpha_0} f = \frac{1}{2\pi i} \int_{\partial B_\varepsilon(\boldsymbol{x}_i)} \frac{\partial_{\boldsymbol{x}}^{\alpha_0} f(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{i-1},\boldsymbol{z},\boldsymbol{x}_{i+1},\ldots,\boldsymbol{x}_n)}{(\boldsymbol{z}-\boldsymbol{x}_i)^2} \, \mathrm{d}\boldsymbol{z},$$

for some $\varepsilon > 0$ with $B_\varepsilon(\boldsymbol{x}_i) \subset \mathbb{C}$ being the ball with radius ε. The integrand is holomorphic in all variables $\boldsymbol{x}_j$, $j \ne i$. Hence, we conclude that $\partial_{\boldsymbol{x}}^{\alpha} f(\boldsymbol{x})$ is holomorphic in all variables and prove the assertion.

The following result is elementary but technical.

Lemma 10 *For $n,p \in \mathbb{N}$, define the set $M := \{\boldsymbol{x} \in \mathbb{C}^n : \mathrm{real}(\sum_{i=1}^n \boldsymbol{x}_i^p) \le 0\}$. Then, there holds $(\mathbb{R}^n)_+ := \{\boldsymbol{x} \in \mathbb{R}^n \setminus \{0\} : \boldsymbol{x}_i \ge 0\} \cap M = \emptyset$ and*

$$\mathrm{dist}(M, \boldsymbol{x}) \ge |\sin(\frac{\pi}{2p})|\|\boldsymbol{x}\| \quad \text{for all } \boldsymbol{x} \in (\mathbb{R}^n)_+.$$

Proof Let $\boldsymbol{x} \in (\mathbb{R}^n)_+$, then we have $\sum_{i=1}^n \boldsymbol{x}_i^p > 0$ and hence $\boldsymbol{x} \notin M$. It is easy to see that the cone $C_p := \{r\exp(i\phi) : r > 0, \phi \in (-\frac{\pi}{2p}, \frac{\pi}{2p})\} \subset \mathbb{C}$ satisfies $\mathrm{real}(x^p) > 0$ for all $x \in C_p$. Thus, we have that

$$C_p^n := \Big(\prod_{i=1}^n (\{0\} \cup C_p) \Big) \setminus \{0\} \subset \mathbb{C}^n$$

satisfies $C_p^n \cap M = \emptyset$. Moreover, a simple geometric argument (see Figure 6) shows that all $x > 0$ satisfy

$$\mathrm{dist}(x, \partial C_p) = x\sin(\pi/(2p)).$$

Since $(\mathbb{R}^n)_+ \subseteq C_p^n$, this implies

$$\mathrm{dist}(M, \boldsymbol{x}) \ge \mathrm{dist}(\partial C_p^n, \boldsymbol{x}) = \Big(\sum_{i=1}^n \boldsymbol{x}_i^2 \sin(\pi/(2p))^2 \Big)^{1/2} = |\sin(\pi/(2p))|\|\boldsymbol{x}\|.$$

This concludes the proof.

Products of asymptotically smooth functions are again asymptotically smooth. This is shown in the next lemma.

21

Lemma 11 *Given two functions $f, g\colon D \times D \to \mathbb{R}$ which are asymptotically smooth (1). Then, also their product fg satisfies (1).*

Proof To simplify the notation, we consider f, g as functions of one variable $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in D \times D \subset \mathbb{R}^{2d}$. For multi-indices $\alpha, \beta \in \mathbb{N}^{2d}$, define

$$\binom{\alpha}{\beta} := \prod_{i=1}^{2d} \binom{\alpha_i}{\beta_i}.$$

Note that there holds $\binom{\alpha}{\beta} \leq \binom{|\alpha|_1}{|\beta|_1}$. This follows from the basic combinatorial fact that the number of possible choices of β_i elements out of a set of α_i elements for all $i = 1, \ldots, 2d$ is smaller than the number of choices of $|\beta|_1$ elements out of a set of $|\alpha|_1$ elements.

The Leibniz formula together with the definition of asymptotically smooth function (1) show for $\alpha \in \mathbb{N}^{2d}$

$$
\begin{aligned}
|\partial_{\boldsymbol{z}}^\alpha (fg)|(\boldsymbol{z}) &\leq \sum_{\substack{\beta \in \mathbb{N}_0^{2d} \\ \beta \leq \alpha}} \binom{\alpha}{\beta} |\partial_{\boldsymbol{z}}^\beta f|(\boldsymbol{z}) |\partial_{\boldsymbol{z}}^{\alpha-\beta} g|(\boldsymbol{z}) \\
&\leq \sum_{\substack{\beta \in \mathbb{N}_0^{2d} \\ \beta \leq \alpha}} \binom{|\alpha|_1}{|\beta|_1} c_1 (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|\beta|_1} |\beta|_1! c_1 (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|\alpha|_1 + |\beta|_1} (|\alpha|_1 - |\beta|_1)! \\
&\leq \sum_{\substack{\beta \in \mathbb{N}_0^{2d} \\ \beta \leq \alpha}} c_1^2 (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|\alpha|_1} |\alpha|_1! \\
&\leq (|\alpha|_1 + 1)^{2d} c_1^2 (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|\alpha|_1} |\alpha|_1! \\
&\lesssim c_1^2 (\widetilde{c}_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|\alpha|_1} |\alpha|_1!,
\end{aligned}
$$

where we used $(|\alpha|_1 + 1)^{2d} \leq (2d \exp(2d))^{|\alpha|_1}$ and $\widetilde{c}_2 = c_2 / (2d \exp(2d))$. This concludes the proof.

The final lemma of this section proves the concatenations of certain asymptotically smooth functions are asymptotically smooth.

Lemma 12 *Let $g\colon D \times D \to \mathbb{R}$ be asymptotically smooth (1) with constants $c_1, c_2 > 0$.*
(i) If $c_g := \sup_{\boldsymbol{x} \in D \times D} g(\boldsymbol{x}) < \infty$. Then, $\exp \circ g$ satisfies (1) with constants $\widetilde{c}_1 := \exp(c_g)$ and $\widetilde{c}_2 := c_2 / (2 \max\{1, c_1\})$.
(ii) If g satisfies $\partial_{\boldsymbol{x}}^\alpha \partial_{\boldsymbol{y}}^\beta g(\boldsymbol{x}, \boldsymbol{y}) \leq C_g$ for all $\alpha, \beta \in \mathbb{N}_0^d$ and some $C_g < \infty$ as well as $g(\boldsymbol{x}, \boldsymbol{y}) \geq C_g^{-1} |\boldsymbol{x} - \boldsymbol{y}|$, then, $g^{1/q}$ satisfies (1) with $\widetilde{\varrho}_1 = 1/2$ and $\widetilde{\varrho}_2 = C_g^{-1}$ for all $q \in \mathbb{N}$.
(iii) If g satisfies the assumptions from (ii) and additionally $g(\boldsymbol{x}, \boldsymbol{y}) \geq c_0 > 0$ for all $\boldsymbol{x}, \boldsymbol{y} \in D$, then $g^{-1/q}$ satisfies (1) for all $q \in \mathbb{N}$.

Proof To simplify the notation, we consider g as a function of one variable $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in D \times D \subset \mathbb{R}^{2d}$. Define the set of all partitions of $\{1, \ldots, n\}$ as

$$\Pi(n) := \Big\{ P \subseteq 2^{\{1,\ldots,n\}} \ : \ S \cap S' = \emptyset \text{ or } S = S' \text{ for all } S, S' \in P, \ \bigcup_{S \in P} S = \{1, \ldots, n\} \Big\}.$$

For a multi-index $\alpha \in \mathbb{N}^{2d}$, we define $\widetilde{\alpha} \in \{1, \ldots, 2d\}^n$ by $\widetilde{\alpha}_i = j$ for all $1 + \sum_{k=1}^{j-1} \alpha_k \leq i \leq \sum_{k=1}^{j} \alpha_k$ and all $1 \leq j \leq 2d$ (e.g., $\alpha = (2, 3, 1, 1)$ yields $\widetilde{\alpha} = (1, 1, 2, 2, 2, 3, 4)$). With $n = |\alpha|_1$ and some $S \in P \in \Pi(n)$, we define

$$\partial_{\boldsymbol{z}}^S g(\boldsymbol{z}) = \Big(\prod_{i \in S} \partial_{\boldsymbol{z}_{\widetilde{\alpha}_i}} \Big) g(\boldsymbol{z}).$$

(the definition implies $\partial_{\boldsymbol{z}}^{\{1,\ldots,n\}} g(\boldsymbol{z}) = \partial_{\boldsymbol{z}}^\alpha g(\boldsymbol{z})$.) With those definitions and given a function $f\colon \mathbb{R} \to \mathbb{R}$, Faà di Bruno's formula reads for a multi-index $\alpha \in \mathbb{N}^{2d}$

$$\partial_{\boldsymbol{z}}^\alpha (f \circ g)(\boldsymbol{z}) = \sum_{P \in \Pi(|\alpha|_1)} (\partial_x^{|P|} f) \circ g(\boldsymbol{z}) \prod_{S \in P} \partial_{\boldsymbol{z}}^S g(\boldsymbol{z}). \tag{37}$$

For (i), Faà di Bruno's formula (37) and $\partial_x^{|P|} \exp = \exp$ show for all multi indices $\alpha \in \mathbb{N}^{2d}$ with $n = |\alpha|_1$ that

$$\partial_{\boldsymbol{x}}^\alpha (\exp \circ g)(\boldsymbol{z}) = \sum_{P \in \Pi(n)} \exp \circ g(\boldsymbol{z}) \prod_{S \in P} \partial_{\boldsymbol{z}}^S g(\boldsymbol{z}).$$

The definition of asymptotically smooth (1) and $\|g\|_{L^\infty(D \times D)} = c_g$ imply

$$
\begin{aligned}
|\partial_{\boldsymbol{x}}^\alpha (\exp \circ g(\boldsymbol{z}))| &\leq \exp(c_g) \sum_{P \in \Pi(n)} \prod_{S \in P} c_1 (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-|S|} |S|! \\
&\leq \exp(c_g) \sum_{P \in \Pi(n)} (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-\sum_{S \in P} |S|} c_1^{|P|} \prod_{S \in P} |S|! \\
&\leq \exp(c_g) \max\{1, c_1\}^n (c_2 |\boldsymbol{x} - \boldsymbol{y}|)^{-n} \sum_{P \in \Pi(n)} \prod_{S \in P} |S|!.
\end{aligned}
$$

With $f(x) := (1-x)^{-1}$, $x \in \mathbb{R} \setminus \{1\}$, we have $\partial_x^k f(x) = k!(1-x)^{-1-k}$. Hence, the last factor can be written, using Faà di Bruno's formula again, as

$$\sum_{P \in \Pi(n)} \prod_{S \in P} |S|! = \sum_{P \in \Pi(n)} \exp \circ f(0) \prod_{S \in P} \partial_x^{|S|} f(0) = \partial_x^n (\exp \circ f)(0).$$

As the function $h(x) := \exp((1-x)^{-1})$, $x \in \mathbb{C}$ is holomorphic at least for $|x| < 1$, Cauchy's integral formula shows

$$|\partial_x^n h(0)| = \frac{n!}{2\pi} \Big| \int_{|z|=1/2} \frac{h(z)}{z^{n+1}} \, \mathrm{d}z \Big| \leq n! 2^n \exp(2).$$

Altogether, we conclude the proof of (i) by

$$|\partial_{\boldsymbol{z}}^{\alpha}(\exp \circ g(\boldsymbol{z}))| \leq \exp(c_g) \Big(\frac{c_2}{2 \max\{1, c_1\}} |\boldsymbol{x} - \boldsymbol{y}| \Big)^{-n} n!.$$

For (ii), Faà di Bruno's formula (37) shows again for $q > 1$

$$|\partial_{\boldsymbol{z}}^{\alpha}(g^{1/q})(\boldsymbol{z})| \leq \sum_{P \in \Pi(n)} |P|! |g(\boldsymbol{z})|^{1/q-|P|} \prod_{S \in P} C_g \leq C_g^n |\boldsymbol{x} - \boldsymbol{y}|^{-|n|} \sum_{P \in \Pi(n)} |P|!,$$

where we used $f(x) := x^{1/q}$ and $|\partial_x^{|P|} f(x)| = |(1/q)(1/q-1)(1/q-2) \cdots (1/q - |P| + 1)||x|^{1/q-|P|} \leq |P|! |x|^{1/q-|P|}$ as well as the boundedness assumption on the derivatives of g from (ii). With $r(x) := \exp(x) - 1$ and $f(x) := (1-x)^{-1}$, $x \in \mathbb{R}$, the last factor satisfies

$$\sum_{P \in \Pi(n)} |P|! = \sum_{P \in \Pi(n)} (\partial_x^{|P|} f) \circ r(0) \prod_{S \in P} (\partial_x^{|S|} r)(0) = \partial_x^n (f \circ r)(0).$$

The function $h(x) := f \circ r(x) = (2 - \exp(x))^{-1}$, $x \in \mathbb{C}$ is holomorphic at least for $|x| \leq 1/2$. As above, this implies

$$\partial_x^n (f \circ r)(0) \leq n! 2^n$$

and thus concludes the proof of (ii).

For (iii), we conclude the proof as for (ii) by use of the estimate $g(z)^{-1/q-|P|} \leq c_0^{-1-n}$.

At last, we are ready to prove Lemma 1 which states that the covariance functions from (2) and (3) are asymptotically smooth (1).

Proof (Proof of Lemma 1) To see (1), consider $\varrho(\cdot, \cdot)$ from (2). We define for complex variables $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathbb{C}$

$$d(\boldsymbol{x} - \boldsymbol{y}) = \Big(\sum_{i=1}^d (\boldsymbol{x}_i - \boldsymbol{y}_i)^p \Big)^{1/p} \in \mathbb{C},$$

whenever $(\cdot)^{1/p}$ is defined in $\mathbb{C}$. and consider $\widetilde{\varrho}(\boldsymbol{x}, \boldsymbol{y})$ which is $\varrho(\boldsymbol{x}, \boldsymbol{y})$ from (2) but with $d(\boldsymbol{x} - \boldsymbol{y})$ instead of $|\boldsymbol{x} - \boldsymbol{y}|_p$. With the notation of Lemma 10, the above sum has positive real part in $O := \{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{C}^{2d} : \boldsymbol{x} - \boldsymbol{y} \notin M \}$. Thus, the function $(\boldsymbol{x}, \boldsymbol{y}) \mapsto d(\boldsymbol{x} - \boldsymbol{y})$ is holomorphic in each variable in O. Since for $a > 0$, $\boldsymbol{x} \mapsto \boldsymbol{x}^\mu K_\mu(a\boldsymbol{x})$ is a holomorphic function on $\mathbb{C} \setminus (\mathbb{R}_- \cup \{0\})$, and $d(\boldsymbol{x} - \boldsymbol{y})$ has positive real part, we deduce that $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \widetilde{\varrho}(\boldsymbol{x}, \boldsymbol{y})$ is holomorphic in each variable in O. Thus, Lemma 9 proves that $\partial_{\boldsymbol{x}}^{\alpha} \partial_{\boldsymbol{y}}^{\beta} \widetilde{\varrho}(\boldsymbol{x}, \boldsymbol{y})$ is holomorphic in O in all variables $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$. Therefore, Cauchy's integral formula applied in all variables shows

$$\partial_{\boldsymbol{x}}^{\alpha} \partial_{\boldsymbol{y}}^{\beta} \widetilde{\varrho}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\prod_{i=1}^d \alpha_i! \beta_i!}{(2\pi i)^{2d}} \int_{\partial B_{\boldsymbol{x},1}} \cdots \int_{\partial B_{\boldsymbol{x},d}} \int_{\partial B_{\boldsymbol{y},1}} \cdots \int_{\partial B_{\boldsymbol{y},d}} \frac{\widetilde{\varrho}(s,t)}{\prod_{i=1}^d (s_i - \boldsymbol{x}_i)^{\alpha_i+1} (t_i - \boldsymbol{y}_i)^{\beta_i+1}} \, \mathrm{d}t \, \mathrm{d}s.$$

The balls $B_{\boldsymbol{x},i}$ and $B_{\boldsymbol{y},i}$ have to be chosen such that $\prod_{i=1}^d B_{\boldsymbol{x},i} \times \prod_{i=1}^d B_{\boldsymbol{y},i} \subset O$. With Lemma 10, and for $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{2d}$ such that $\boldsymbol{x} - \boldsymbol{y} \in (\mathbb{R}^n)_+$ (note that Lemma 10 implies $(\boldsymbol{x}, \boldsymbol{y}) \in O$), this can be achieved by setting $B_{\boldsymbol{x},i} := B_\varepsilon(\boldsymbol{x}_i)$ and $B_{\boldsymbol{y},i} := B_\varepsilon(\boldsymbol{y}_i)$ with $\varepsilon := \sin(\pi/(2p)) |\boldsymbol{x} - \boldsymbol{y}|/(2d+1)$. From this, we obtain the estimate

$$|\partial_{\boldsymbol{x}}^{\alpha} \partial_{\boldsymbol{y}}^{\beta} \varrho(\boldsymbol{x}, \boldsymbol{y})| = |\partial_{\boldsymbol{x}}^{\alpha} \partial_{\boldsymbol{y}}^{\beta} \widetilde{\varrho}(\boldsymbol{x}, \boldsymbol{y})| \lesssim \frac{\alpha! \beta! (2d+1)^{|\alpha|_1+|\beta|_1}}{|\boldsymbol{x} - \boldsymbol{y}|^{|\alpha|_1+|\beta|_1}} \max_{(s,t) \in D \times D} |\widetilde{\varrho}(s,t)| \tag{38}$$

for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{2d}$ such that $\boldsymbol{x} - \boldsymbol{y} \in (\mathbb{R}^n)_+$, where the first equality follows from $d(\boldsymbol{x} - \boldsymbol{y}) = |\boldsymbol{x} - \boldsymbol{y}|_p$ for all $\boldsymbol{x} - \boldsymbol{y} \in (\mathbb{R}^n)_+$. To remove the restriction $\boldsymbol{x} - \boldsymbol{y} \in (\mathbb{R}^n)_+$, consider $b \in \{0, 1\}^d$ and define the function

$$F_b(\boldsymbol{x}, \boldsymbol{y}) := ((-1)^{b_1} \boldsymbol{x}_1, \ldots, (-1)^{b_d} \boldsymbol{x}_d, (-1)^{b_1} \boldsymbol{y}_1, \ldots, (-1)^{b_d} \boldsymbol{y}_d).$$

Since we consider $\varrho(\cdot, \cdot)$ from (2), there holds $\varrho \circ F_b = \varrho$. Since for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ with $\boldsymbol{x} \neq \boldsymbol{y}$, there exists some $b \in \{0, 1\}^d$ such that $(\boldsymbol{x}_b, \boldsymbol{y}_b) := F_b(\boldsymbol{x}, \boldsymbol{y})$ satisfies $\boldsymbol{x}_b - \boldsymbol{y}_b \in (\mathbb{R}^n)_+$, we prove (38) for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ with $\boldsymbol{x} \neq \boldsymbol{y}$. Finally, the fact $\alpha! \beta! \leq |\alpha + \beta|_1!$, proves that $\varrho(\cdot, \cdot)$ from (2) is asymptotically smooth (1).

Next, consider the covariance function $\varrho(\cdot, \cdot)$ from (3). By definition $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is continuous on $\overline{D}$. Hence, $\det(\boldsymbol{\Sigma}_{\boldsymbol{x}}) \geq c_0 > 0$ for all $\boldsymbol{x} \in D$. The assumption (4) implies that also $\det(\boldsymbol{\Sigma}_{\boldsymbol{x}})$ has bounded derivatives in the sense of (4) (since $\det(\boldsymbol{\Sigma}_{\boldsymbol{x}})$ is a polynomial in the matrix entries of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$). Thus, Lemma 12 shows that the functions $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \det(\boldsymbol{\Sigma}_{\boldsymbol{x}})^{1/4}$, $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \det(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{1/4}$, and $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \det(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{y}})^{-q}$, $q \in \{1/2, 1\}$ satisfy (1). With $\boldsymbol{\Sigma}_{\boldsymbol{x}}$, also all functions $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{x}}$ defined by considering only sub-matrices of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ satisfy (4). Thus, Cramer's rule and Lemma 11 show that the map $(\boldsymbol{x}, \boldsymbol{y}) \mapsto ((\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1})_{i,j}$ for all $i, j \in \{1, \ldots, d\}$ satisfies (1). From this, we conclude (again with Lemma 11), that $(\boldsymbol{x}, \boldsymbol{y}) \mapsto (\boldsymbol{x} - \boldsymbol{y})^T (\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1} (\boldsymbol{x} - \boldsymbol{y})$ as sum and product of asymptotically smooth functions is asymptotically smooth (1). Finally, Lemma 12 shows that $\varrho(\boldsymbol{x}, \boldsymbol{y})$ satisfies (1). This concludes the proof.

B Proof of Proposition 1

The following lemmas state facts about the H^2-matrix block partitioning, which are well-known but cannot be found explicitly in the literature.

Lemma 13 *Under Assumption 1, there exists a constant $C_B > 0$ which depends only on d, C_u, D, and $B_{X_{\mathrm{root}}}$ such that all $X \in \mathbb{T}_{\mathrm{cl}}$ satisfy*

$$\mathrm{diam}(B_X)^d \leq C_B|B_X|, \tag{39a}$$

$$C_B^{-1}N|B_X| - 1 \leq |X| \leq 1 + C_BN|B_X|, \tag{39b}$$

$$|B_X| = 2^{-\mathrm{level}(X)}|B_{X_{\mathrm{root}}}|. \tag{39c}$$

Moreover, all $(X,Y) \in \mathbb{T}$ satisfy

$$C_{BB}^{-1}\mathrm{diam}(B_X) \leq \mathrm{diam}(B_Y) \leq C_{BB}\mathrm{diam}(B_X), \tag{40}$$

where $C_{BB} > 0$ depends only on C_B, C_{leaf}, and D.

Proof The first estimate (39a) follows from the fact that always the longest edge of a bounding box is halved. This means that the ratio $L_{\max}/L_{\min}$ of the maximal and the minimal side length of a bounding box B_X stays bounded in terms of the corresponding ratio for $B_{X_{\mathrm{root}}}$. Therefore, we have

$$\mathrm{diam}(B_X)^d \leq (\sqrt{d}L_{\max})^d \lesssim d^{d/2}L_{\min}^d \leq d^{d/2}|B_X|.$$

To see the second estimate (39b), consider a given bounding box B with side lengths $L_1, \ldots, L_d$. Due to Assumption 1 the balls $Q_{\boldsymbol{x}}$ with centre $\boldsymbol{x}$ and radius $C_u^{-1}N^{-1/d}/2$ for all $\boldsymbol{x} \in \mathcal{N}$ do not overlap. All balls $Q_{\boldsymbol{x}}$ with $\boldsymbol{x} \in B$ are containted in a box with sidelengths $L_{\max} + C_u^{-1}N^{-1/d}$. Thus, the number m_B of $\boldsymbol{x} \in \mathcal{N}$ contained in B can be bounded by

$$m_B \lesssim \frac{(L_{\max} + C_u^{-1}N^{-1/d})^d}{C_u^{-d}/(N2^d)} \leq \frac{dL_{\max}^d}{C_u^{-d}/(N2^d)} + d2^d.$$

Since $m_B \leq 1$ if $L_{\max} < C_u^{-1}N^{-1/d}/2$ and since $L_{\max}^d \simeq |B|$, we may improve the estimate to

$$m_B \leq 1 + C_B|B|N,$$

where C_B depends only on d and C_u. On the other hand, Assumption 1 implies that any ball with radius $C_uN^{-1/d}$ contains at least one point $\boldsymbol{x} \in \mathcal{N}$. Since each such ball fits inside a box with sidelength $2C_uN^{-1/d}$, we obtain

$$m_B \gtrsim \left\lfloor \frac{L_{\min}^d}{2^d C_u^{-d}/N} \right\rfloor$$

points of $\mathcal{N}$. This allows us to estimate $m_B \geq C_B^{-1}|B|N - 1$ and conclude (39b). The estimate (39c) follows from the fact $|B_X| = |B_{X'}|/2$ for all $X \in \mathrm{sons}(X')$. For (40), we observe with (39b) that

$$\frac{|X| - 1}{N} \lesssim |B_X| \lesssim \frac{|X| + 1}{N}$$

for all $X \in \mathbb{T}_{\mathrm{cl}}$ with hidden constants depending only on C_B. Thus, with (39c), we have for all $X \in \mathbb{T}_{\mathrm{cl}}$ with $X \in \mathrm{sons}(X')$ that

$$2^{-\mathrm{level}(X)} \geq 2^{-\mathrm{level}(X')}/2 \simeq |B_{X'}| \gtrsim C_{\mathrm{leaf}}/N.$$

Moreover, if additionally $\mathrm{sons}(X) = \emptyset$, we have even $2^{-\mathrm{level}(X)} \simeq |B_x| \lesssim C_{\mathrm{leaf}}/N$. By definition of the block-tree $\mathbb{T}$, a level difference between X and Y for $(X,Y) \in \mathbb{T}$ can only happen, if $\mathrm{sons}(X) = \emptyset$ or $\mathrm{sons}(Y) = \emptyset$. Assume $\mathrm{sons}(X) = \emptyset$. In this case, we have $\mathrm{level}(Y) \geq \mathrm{level}(X)$. Then, we have

$$2^{-\mathrm{level}(X)} \simeq C_{\mathrm{leaf}}/N \lesssim |Y|/N \simeq 2^{-\mathrm{level}(Y)},$$

with hidden constants depending only on C_B and D. This implies $\mathrm{level}(Y) \leq \mathrm{level}(X) + C$ for some constant $C > 0$ which depends only on C_{leaf}, D, and C_B from (39). From this we derive (40) by use of (39).

Lemma 14 *Given the definition of $\mathbb{T}_{\mathrm{far}}$ in Section 3.1, there exists a constant $C > 0$ such that all $(X,Y) \in \mathbb{T}_{\mathrm{far}}$ satisfy*

$$C^{-1}\mathrm{diam}(B_X) \leq \mathrm{dist}(B_X, B_Y) \leq C\,\mathrm{diam}(B_X). \tag{41}$$

Proof By Lemma 13, we have

$$\max\{\mathrm{diam}(B_X), \mathrm{diam}(B_Y)\} \simeq 2^{-\mathrm{level}(X)/d}.$$

For $(X,Y) \in \mathrm{sons}(X', Y')$, we obtain additionally

$$\mathrm{dist}(B_{X'}, B_{Y'}) + 2^{-\mathrm{level}(X')/d} \gtrsim \mathrm{dist}(B_X, B_Y).$$

By definition of the block-partitioning, for $(X,Y) \in \mathbb{T}_{\mathrm{far}}$ there holds that B_X, B_Y satisfy (5) and $B_{X'}, B_{Y'}$ do not satisfy (5). Altogether, this implies

$$\mathrm{dist}(B_X, B_Y) \lesssim \mathrm{dist}(B_{X'}, B_{Y'}) + 2^{-\mathrm{level}(X')/d} \lesssim \left(\frac{1}{\eta} + 1\right)2^{-\mathrm{level}(X')/d} \lesssim \max\{\mathrm{diam}(B_X), \mathrm{diam}(B_Y)\},$$

where we used $|\mathrm{level}(X') - \mathrm{level}(X)| \leq 1$. This concludes the proof.

The following lemma gives some basic facts about tensorial Chebychev-interpolation (see, e.g., [2, Section 4.4])

Lemma 15 *Let $f\colon B \to \mathbb{R}$ for an axis parallel box $B \subseteq \mathbb{R}^{2d}$ such that $\partial_j^k f \in L^\infty(B)$ for all $j = 1, \dots, d$ and all $0 \leq k \leq p$. Then, the tensorial Chebychev-interpolation operator of order p, $I_p\colon C(B) \to \mathcal{P}^p(B)$ satisfies*

$$\sup_{\boldsymbol{x} \in B} |I_p f(\boldsymbol{x}) - f(\boldsymbol{x})| \leq 2d \Lambda_p^{2d-1} 4 \frac{4^{-p}}{(p+1)!} \operatorname{diam}(B)^p \sum_{i=1}^{2d} \|\partial_{\boldsymbol{x}_i}^{p+1} f\|_{L^\infty(B)}, \tag{42}$$

where

$$\Lambda_p := \sup_{f \in C([-1,1])} \frac{\|I_p^{\boldsymbol{x}} f\|_{L^\infty([-1,1])}}{\|f\|_{L^\infty([-1,1])}} \leq \frac{2}{\pi} \log(p+1) + 1 \tag{43}$$

is the operator norm of the one dimensional Chebychev interpolation operator

Proof It is well-known that the one dimensional Chebychev interpolation operator $I_p^{\boldsymbol{x}}$ satisfies the error estimate for any $f \in C([-1,1])$

$$\|u - I_p^{\boldsymbol{x}} f\|_{L^\infty([-1,1])} \leq 4 \frac{2^{-p}}{(p+1)!} \|\partial^{(p+1)} f\|_{L^\infty([-1,1])}$$

with an operator norm given in (43). Consider $B := [-1,1]^{2d}$. Then, there holds with $I_p^{\boldsymbol{x}_i}$ denoting interpolation in the $\boldsymbol{x}_i$-variable $i \in \{1, \dots, 2d\}$

$$|f - I_p f| = |f - I_p^{\boldsymbol{x}_1} f + I_p^{\boldsymbol{x}_1} f - I_p^{\boldsymbol{x}_2} I_p^{\boldsymbol{x}_1} f + \dots - I_p f|$$

$$\leq \sum_{i=1}^{2d} 4 \frac{2^{-p}}{p!} \|\partial_{\boldsymbol{x}_i}^p I_p^{\boldsymbol{x}_1} (I_p^{\boldsymbol{x}_2} \dots I_p^{\boldsymbol{x}_{i-1}}) f\|_{L^\infty(B)} \leq \sum_{i=1}^{2d} \Lambda_p^{i-1} 4 \frac{2^{-p}}{p!} \|\partial_{\boldsymbol{x}_i}^{(p+1)} f\|_{L^\infty(B)}$$

$$\leq 2d \Lambda_p^{2d-1} 4 \frac{2^{-p}}{p!} \|\partial_{\boldsymbol{x}_i}^{(p+1)} f\|_{L^\infty(B)}.$$

Since, for any affine transformation $A\colon \mathbb{R}^{2d} \to \mathbb{R}^{2d}$, we have $I_p(f \circ A) = I_p(f) \circ A$, a standard scaling argument concludes the proof.

Proof (Proof of Proposition 1)
We start by proving that $\lambda_{\min}(\boldsymbol{C}_p) > 0$ if p satisfies (7). To that end, note

$$\lambda_{\min}(\boldsymbol{C}_p) = \min_{\boldsymbol{z} \in \mathbb{R}^N \setminus \{0\}} \frac{(\boldsymbol{C}_p \boldsymbol{z})^T \boldsymbol{z}}{|\boldsymbol{z}|} \geq \min_{\boldsymbol{z} \in \mathbb{R}^N \setminus \{0\}} \frac{(\boldsymbol{C} \boldsymbol{z})^T \boldsymbol{z}}{|\boldsymbol{z}|} - \sup_{\boldsymbol{z} \in \mathbb{R}^N \setminus \{0\}} \frac{((\boldsymbol{C}_p - \boldsymbol{C}) \boldsymbol{z})^T \boldsymbol{z}}{|\boldsymbol{z}|}$$

$$\geq \lambda_{\min}(\boldsymbol{C}) - \|\boldsymbol{C} - \boldsymbol{C}_p\|_2 \geq \lambda_{\min}(\boldsymbol{C}) - \|\boldsymbol{C} - \boldsymbol{C}_p\|_F,$$

since the Frobenius norm is an upper bound for the spectral norm. By use of (6) (which is proved below) and (7), we conclude $\lambda_{\min}(\boldsymbol{C}_p) > 0$.

To see (6), we first estimate the maximal depth of the tree $\mathbb{T}_{\mathrm{cl}}$. With (39b)–(39c), we obtain $C_{\mathrm{leaf}} \leq |X| \lesssim 2^{-\mathrm{level}(X)}$ for all $X \in \mathbb{T}_{\mathrm{cl}}$ with $\mathrm{sons}(X) \neq \emptyset$. Thus, there holds

$$\max_{X \in \mathbb{T}_{\mathrm{cl}}} \mathrm{level}(X) \lesssim \log(|\mathcal{N}|).$$

Second, we bound the so-called sparsity constant

$$C_{\mathrm{sparse}} := \max_{X \in \mathbb{T}_{\mathrm{cl}}} \Big(|\{Y \in \mathbb{T}_{\mathrm{cl}} : (X,Y) \in \mathbb{T}_{\mathrm{near}} \cup \mathbb{T}_{\mathrm{far}}\}|$$

$$+ |\{Y \in \mathbb{T}_{\mathrm{cl}} : (Y,X) \in \mathbb{T}_{\mathrm{near}} \cup \mathbb{T}_{\mathrm{far}}\}| \Big).$$

The H-matrix case can be found in [9, Lemma 4.5]. For the H^2-matrix case, the combination of (40) and (41) (from Lemma 14) shows that $(X,Y) \in \mathbb{T}_{\mathrm{far}}$ only if B_Y touches the (hyper-) annulus with center B_X and radii $C^{-1}\operatorname{diam}(B_X)$ and $C\operatorname{diam}(B_X)$. By comparing the volumes of this annulus and of B_Y and using the fact that all the bounding boxes are disjoint, we see that the number of Y such that $(X,Y) \in \mathbb{T}_{\mathrm{far}}$ is bounded in terms of C and the constants in (39).

For $Y \in \mathbb{T}_{\mathrm{cl}}$ such that $(X,Y) \in \mathbb{T}_{\mathrm{near}}$, we have with (39)–(40)

$$\operatorname{diam}(B_X) \simeq \max\{\operatorname{diam}(B_X), \operatorname{diam}(B_Y)\} > \eta \operatorname{dist}(B_X, B_Y).$$

Again, comparing the volumes of the ball with radius $\operatorname{diam}(B_X)$ and of B_Y, we see that the number of Y such that $(X,Y) \in \mathbb{T}_{\mathrm{near}}$ is bounded in terms of the constants in (39). Altogether, we bound C_{sparse} uniformly in terms of the constants of Lemma 13. Now, [2, Lemma 3.38] proves the estimate for storage requirements and [2, Theorem 3.42] proves the estimate for matrix-vector multiplication.

It remains to prove the error estimate (see also [2, Section 4.6] for the integral operator case). To that end, note that since the near field $\mathbb{T}_{\text{near}}$ is stored exactly, there holds

$$\|\boldsymbol{C} - \boldsymbol{C}_p\|_F^2 = \sum_{(X,Y)\in\mathbb{T}_{\text{far}}} \|\boldsymbol{C}|_{I(X)\times I(Y)} - V^X M^{XY}(W^Y)^T\|_F^2.$$

Given, $(i,j) \in I(X) \times I(Y)$, we have with the interpolation operator I_p from Lemma 15 and (1)

$$|\boldsymbol{C}_{ij} - (\boldsymbol{C}_p)_{ij}| = \Big|\varrho(\boldsymbol{x}_i,\boldsymbol{x}_j) - \sum_{n,m=1}^{p^d} \varrho(q_n^X, q_m^Y) L_n^X(\boldsymbol{x}_i) L_m^Y(\boldsymbol{x}_j)\Big| = |\varrho(\boldsymbol{x}_i,\boldsymbol{x}_j) - (I_p c)(\boldsymbol{x}_i,\boldsymbol{x}_j)|$$

$$\lesssim (\log(p)+1)^{2d-1} \frac{4^{-p}}{(p+1)!} \text{diam}(B_X \times B_Y)^p \sum_{i=1}^d \left(\|\partial_{\boldsymbol{x}_i}^{(p+1)} c\|_{L^\infty(B)} + \|\partial_{\boldsymbol{y}_i}^{(p+1)} c\|_{L^\infty(B_X\times B_Y)}\right)$$

$$\lesssim (\log(p)+1)^{2d-1} \frac{4^{-p}}{(p+1)!} \text{diam}(B_X \times B_Y)^p (c_2\text{dist}(B_X,B_Y))^{-p} p!.$$

With the admissibility condition (5), we get

$$\text{diam}(B_X \times B_Y) \lesssim \max\{\text{diam}(B_X),\text{diam}(B_Y)\} \le \eta\,\text{dist}(B_X,B_Y)$$

and hence

$$|\boldsymbol{C}_{ij} - (\boldsymbol{C}_p)_{ij}| \lesssim (\log(p)+1)^{2d-1}\big(\frac{\eta}{4c_2}\big)^p.$$

The combination of the above estimates concludes the proof.

References

1. I. Babuška, B. Andersson, P. J. Smith, and K. Levin. Damage analysis of fiber composites. I. Statistical analysis on fiber scale. *Comput. Methods Appl. Mech. Engrg.*, 172(1-4):27–77, 1999.
2. Steffen Börm. *Efficient numerical methods for non-local operators*, volume 14 of *EMS Tracts in Mathematics*. European Mathematical Society (EMS), Zürich, 2010.
3. Grace Chan and Andrew T.A. Wood. Algorithm as 312: An algorithm for simulating stationary gaussian random fields. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(1):171–181, 1997.
4. C. R. Dietrich and G. N. Newsam. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. *SIAM J. Sci. Comput.*, 18(4):1088–1107, 1997.
5. J. Dölz, H. Harbrecht, and Ch. Schwab. Covariance regularity and h-matrix approximation for rough random fields. *Numerische Mathematik*, pages 1–27, 2016.
6. I. Elishakoff, editor. *Whys and hows in uncertainty modelling*, volume 388 of *CISM Courses and Lectures*. Springer-Verlag, Vienna, 1999. Probability, fuzziness and anti-optimization.
7. Andreas Frommer. Monotone convergence of the Lanczos approximations to matrix functions of Hermitian matrices. *Electron. Trans. Numer. Anal.*, 35:118–128, 2009.
8. I.G. Graham, F.Y. Kuo, D. Nuyens, R. Scheichl, and I.H. Sloan. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *Journal of Computational Physics*, 230(10):3668 – 3694, 2011.
9. Lars Grasedyck and Wolfgang Hackbusch. Construction and arithmetics of H-matrices. *Computing*, 70(4):295–334, 2003.
10. Wolfgang Hackbusch. *Hierarchical matrices: algorithms and analysis*, volume 49 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2015.
11. Helmut Harbrecht, Michael Peters, and Markus Siebenmorgen. Efficient approximation of random fields for numerical applications. *Numer. Linear Algebra Appl.*, 22(4):596–617, 2015.
12. D. Higdon, J. Swall, and J. Kern. Non-stationary spatial modeling.
13. Nicholas J. Higham. Computing real square roots of a real matrix. *Linear Algebra Appl.*, 88/89:405–430, 1987.
14. Nicholas J. Higham. Stable iterations for the matrix square root. *Numer. Algorithms*, 15(2):227–242, 1997.
15. Charles Kenney and Alan J. Laub. Rational iterative methods for the matrix sign function. *SIAM J. Matrix Anal. Appl.*, 12(2):273–291, 1991.
16. B. N. Khoromskij, A. Litvinenko, and H. G. Matthies. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*, 84(1-2):49–67, 2009.
17. Igor Moret. Rational Lanczos approximations to the matrix square root and related functions. *Numer. Linear Algebra Appl.*, 16(6):431–445, 2009.
18. Bernhard A. Schmitt. Perturbation bounds for matrix square roots and pythagorean sums. *Linear Algebra and its Applications*, 174:215 – 227, 1992.