Tucker Tensor analysis of Matérn functions in spatial statistics

Alexander Litvinenko, David Keyes, Venera Khoromskaia,
Boris N. Khoromskij and Hermann G. Matthies

November 18, 2017

Abstract

In this work, we describe advanced numerical tools for working with multivariate functions and for the analysis of large data sets. These tools will drastically reduce the required computing time and the storage cost, and, therefore, will allow us to consider much larger data sets or finer meshes. Covariance matrices are crucial in spatio-temporal statistical tasks, but are often very expensive to compute and store, especially in 3D. Therefore, we approximate covariance functions by cheap surrogates in a low-rank tensor format. We apply the Tucker and canonical tensor decompositions to a family of Matérn- and Slater-type functions with varying parameters and demonstrate numerically that their approximations exhibit exponentially fast convergence. We prove the exponential convergence of the Tucker and canonical approximations in tensor rank parameters. Several statistical operations are performed in this low-rank tensor format, including evaluating the conditional covariance matrix, spatially averaged estimation variance, computing a quadratic form, determinant, trace, loglikelihood, inverse, and Cholesky decomposition of a large covariance matrix. Low-rank tensor approximations reduce the computing and storage costs essentially. For example, the storage cost is reduced from an exponential $\mathcal{O}(n^d)$ to a linear scaling $\mathcal{O}(drn)$, where d is the spatial dimension, n is the number of mesh points in one direction, and r is the tensor rank. Prerequisites for applicability of the proposed techniques are the assumptions that the data, locations, and measurements lie on a tensor (axes-parallel) grid and that the covariance function depends on a distance, $\|x - y\|$.

AMS subject classification: 60H15, 60H35, 65N25
Key words: *Fourier transform, low-rank tensor approximation, geostatistical optimal design, Kriging, Matérn covariance, Hilbert tensor, Kalman filter, Bayesian update, loglikelihood surrogate.*
Corresponding author: Dr. Alexander Litvinenko, alexander.litvinenko@kaust.edu.sa

1 Introduction

Nowadays it is very common to work with large spatial data sets [63, 14, 61, 43, 60, 51], for instance, with satellite data, collected over a very large area (e.g., the data collected by the National Center for Atmospheric Research, USA[1]). This data can also come from a computer simulator code as a solution of a certain multiparametric equation (e.g., Weather research and Forecasting model[2]), it could be also sensor data from multiple sources. Typical operations in spatial statistics, such as evaluating the spatially averaged estimation variance, computing quadratic forms of the conditional covariance matrix, or computing maximum of likelihood function [61] require high computing power and time. Our motivation for using low-rank tensor techniques is that operations on advanced matrices, such as hierarchical, low-rank and sparse matrices, are limited by their high computational costs, especially in 3D and for a large number of observations.

A tensor can be simply defined as a high-order matrix, where multi-indices are used instead of indices (see Section 3 and Eq. 3.1 for rigorous definition). One way to obtain a tensor from a vector or matrix

[1] https://www.earthsystemgrid.org/
[2] https://www.mmm.ucar.edu/weather-research-and-forecasting-model

is to reshape it. For example, we assume $\mathbf{v} \in \mathbb{R}^{10^6}$ is a vector. We reshape it and obtain a matrix of size $10^3 \times 10^3$, or a tensor of order 3 of size $10^2 \times 10^2 \times 10^2$ or a tensor of order 6 of size $10 \times ... \times 10$ (6 times). Each element of such six-dimensional hypercube is described by the multi-index $\alpha = (\alpha_1, ..., \alpha_6)$. The obtained tensors contain not only rows and columns, but also *slices* and *fibers* [39, 40, 9]. These slices and fibers can be analyzed for linear dependences, super symmetry, or sparsity and may result in a strong data compression. Another difference between tensors and matrices is that a matrix (obtained, for instance, after the discretization of a kernel $c(\mathbf{x}, \mathbf{y}) = c(|\mathbf{x} - \mathbf{y}|)$) separates a point $\mathbf{x} = (x_1, ..., x_d) \in \mathbb{R}^d$ from a point $\mathbf{y} = (y_1, ..., y_d) \in \mathbb{R}^d$, whereas the corresponding tensor (depending on tensor format) separates $x_1 - y_1$ from $x_2 - y_2$... and $x_d - y_d$. This implies that tensors may have not just one rank like a matrix, but many. Therefore, we speak about *tensor rank*, but not matrix rank. In this work, we consider two very common tensor formats: canonical (denoted as CP) and Tucker (see Section 3).

Low-rank tensor techniques cannot completely replace, e.g., low-rank or hierarchical techniques, but can complement them. For example, a 3D function can be approximated as the sum of the tensor products of one-dimensional function. Then the usual matrix techniques can be applied to those 1D functions.

To be more concrete, we consider a relatively wide class of Matérn covariance functions. We demonstrate how to approximate Matérn covariance matrices in a low-rank tensor format, then how to perform typical Kriging and spatial statistics operations in this tensor format. Matérn covariance matrices typically depend on three to five unknown hyper-parameters, such as smoothness, three covariance lengths (in a 3D anisotropic case), and variance. We study the dependences of tensor ranks and approximation errors on these parameters. Splitting the spatial variables via low-rank techniques reduces the computing cost for a matrix-vector product from $\mathcal{O}(N^2)$ to $\mathcal{O}(dr^2n^2)$ FLOPs, where d is the spatial dimension, r is the tensor rank, and n is the number of mesh points along the longest edge of the computational domain. For simplicity, we assume that $N = n^d$ (e.g., $d = 4$ for a time-space problem in 3D). Other motivating factors for applying low-rank tensor techniques include the following:

1. The storage cost is reduced from $\mathcal{O}(n^d)$ to $\mathcal{O}(drn)$ or, depending on the tensor format, to $\mathcal{O}(drn + r^d)$, where $d > 1$.

2. The low-rank tensor technique allows us to compute not only the matrix-vector product, but also the inverse $\mathbf{C}^{-1}$, square root $\mathbf{C}^{1/2}$, matrix exponent $\exp(\mathbf{C})$, $\text{trace}(\mathbf{C})$, $\det(\mathbf{C})$, and a likelihood function.

3. The low-rank tensor approximation is relatively new, but already a well-studied technique with free software libraries available.

4. The approximation accuracy is fully controlled by the tensor rank. The full rank gives an exact representation;

5. Low-rank techniques are either faster than a Fourier transform ($\mathcal{O}(drn)$ vs. $\mathcal{O}(n^d \log n^d)$) or can be efficiently combined with it [50, 12];

General limitations of the tensor technique are that 1) it could be time consuming to compute a low-rank tensor decomposition; 2) it requires axes-parallel mesh; 3) some theoretical estimations exist for functions depending on $|x - y|$ (although more general functions have a low-rank representation in practice).

During the last few years, there has been great interest in numerical methods for representing and approximating large covariance matrices [43, 53, 55, 50, 1, 2, 42]. Low-rank tensors were previously applied to accelerated Kriging and spatial design by orders of magnitude [50]. The covariance matrix under consideration was assumed to be circulant, and the first column had a low-rank decomposition. Therefore, d-dimensional Fourier was applied to and drastically reduce the storage and the computing cost.

The maximum likelihood estimator was computed for parameter-fitting given Gaussian observations with a Matérn covariance matrix [46]. The presented framework for unstructured observations in two

spatial dimensions allowed for an evaluation of the log-likelihood and its gradient with computational complexity $\mathcal{O}(n^{3/2})$. W. Hackbusch developed the $\mathcal{H}$-matrix technique [21] for approximating differential and integral operators. $\mathcal{H}$-matrices are very robust for approximating the covariance matrix [37, 55, 2, 25], [1, 4], its inverse [1], and its Cholesky decomposition [37, 43, 42], but can also be expensive, especially for large n in 3D. However, more efficient methods for fast and efficient matrix linear algebra operations are still needed.

The key idea is to compute a low-rank decomposition not of the covariance function (it could be hard), but of its analytically known spectral density (which could be a much easier object) and then apply the inverse Fourier to the obtained low-rank components. The Fourier transformation of the Matérn covariance function is known analytically as the Hilbert tensor. This Hilbert tensor can be decomposed numerically in a low-rank tensor format. Both the Fourier Transformation and its inverse have the canonical (CP) tensor rank-1. Therefore, the inverse Fourier does not change the tensor rank of the argument. By applying the inverse Fourier to the low-rank tensor, we obtain a low-rank approximation of the initial covariance matrix, which can be further used in the Kalman Filter update, Karhunen-Loève Expansion, Bayesian Update, and Kriging.

The structure of the paper is as follows. In Section 2, we list typical tasks from statistics that motivate us to use low-rank tensor techniques and define the Matérn covariance functions and their Fourier transformations. Section 3 is devoted to low-rank tensor decomposition. Sub-sections 3.4, 3.5, and 3.6 contain the main theoretical contribution of this work. We present low-rank tensor techniques and separate radial basis functions using the Laplace transform and the sinc quadrature, give estimations of the approximation error, convergence rate, and the tensor rank, and we also prove the existence of a low-rank approximation of a Matérn function. Section 4 contains another important contribution of this work, namely, the solutions to typical statistical tasks in the low-rank tensor format.

2 Motivation

2.1 Problem settings in spatial statistics

Below, we formulate **five tasks**. These computational tasks are very common and important in statistics. Fast and efficient solution of these tasks will help to solve many real-world problems, such as the weather prediction, moisture modeling, and optimal design in geostatistics.

Task 1. Approximate a Matérn covariance function in a low-rank tensor format. The covariance function $c(\mathbf{x}, \mathbf{y})$, $\mathbf{x} = (x_1, ..., x_d)$, $\mathbf{y} = (y_1, ..., y_d)$, is discretized on a tensor grid with N mesh points, $N = n^d$, $d \geq 1$ and $\varepsilon > 0$. The task is to find the following decomposition into one-dimensional functions, i.e., $\|c(\mathbf{x}, \mathbf{y}) - \sum_{i=1}^{r} \prod_{\mu=1}^{d} c_{i\mu}(x_\mu, y_\mu)\| \leq \varepsilon$, for some given $\varepsilon > 0$. Alternatively, we may look for factors $\mathbf{C}_{i\mu}$ such that $\|\mathbf{C} - \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}\| \leq \varepsilon$. Here, matrices $\mathbf{C}_{i\mu}$ correspond to the one-dimensional covariance functions $c_{i\mu}(x_\mu, y_\mu)$ in the direction μ.

Task 2. Computing of square root of C. The square root $\mathbf{C}^{1/2}$ of the covariance matrix $\mathbf{C}$ is needed in order to generate random fields and processes. It is also used in the Kalman filter update.

Task 3. Kriging. Spatial statistics and kriging [38] are used to model the distribution of ore grade, forecast of rainfall intensities, moisture, temperatures, or contaminant. The missing values are interpolated from the known measurements by kriging [45, 29]. When considering space-time kriging on fine meshes [65, 13, 26, 28], kriging may easily exceed the computational power of modern computers. Estimating the variance of kriging and geostatistical optimal design problems are especially numerically intensive [47, 49, 59].

The kriging can be defined as follows. Let $\hat{\mathbf{s}}$ be the $N \times 1$ vector of values to be estimated with zero expectation and covariance matrix $\mathbf{C}_{ss}$. Let $\mathbf{y}$ be the $m \times 1$, $m \ll N$, vector of measurements. The corresponding cross- and auto-covariance matrices are denoted by $\mathbf{C}_{sy}$ and $\mathbf{C}_{yy}$, and sized $N \times m$ and $m \times m$, respectively. If the measurements are subject to error, an error covariance matrix $\mathbf{R}$ is included in $\mathbf{C}_{yy}$. Using this notation, the kriging estimate $\hat{\mathbf{s}}$ is given by $\hat{\mathbf{s}} = \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{y}$.

Task 4. Geostatistical design. The goal of geostatistical optimal design is to optimize the sampling patterns from which the data values in $\mathbf{y}$ will be obtained. The objective function that will be minimized is typically a scalar measure of either the conditional covariance matrix or the estimation variance (Eq. 4.9). The two most common measures for geostatistical optimal design are φ_A and φ_C:

$$\varphi_A = N^{-1} \operatorname{trace}\left[\mathbf{C}_{ss|y}\right], \quad \text{and} \quad \varphi_C = \mathbf{z}^T(\mathbf{C}_{ss|y})\mathbf{z}, \tag{2.1}$$

where $\mathbf{C}_{ss|y} := \mathbf{C}_{ss} - \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys}$ [47, 49].

Task 5. Computing the joint Gaussian log-likelihood function. We assume that $\boldsymbol{z} \in \mathbb{R}^N$ is an available vector of measurements, and $\boldsymbol{\theta}$ is an unknown vector of the parameters of a covariance matrix $\mathbf{C}$. The task is to compute the maximum likelihood estimation (MLE), where the log-likelihood function is as follows

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log\det\{\mathbf{C}(\boldsymbol{\theta})\} - \frac{1}{2}(\mathbf{z}^T \cdot \mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{z}). \tag{2.2}$$

The difficulty here is that each iteration step of a maximization procedure requires the solution of a linear system $\mathbf{L}\mathbf{v} = \mathbf{z}$, the Cholesky decomposition, and the determinant.

In Section 4 we give detailed solutions. We give strict definition of tensors later in Section 3.

2.2 Matérn covariance and its Fourier transform

A low-rank approximation of the covariance function is a key component of the tasks formulated above. Among of the many covariance models available, the Matérn family [44, 24] is widely used in spatial statistics, geostatistics [7], machine learning [4], image analysis, weather forecast, moisture modeling, and as the correlation for temperature fields [48]. The work [24] introduced the Matérn form of spatial correlations into statistics as a flexible parametric class with one parameter determining the smoothness of the underlying spatial random field.

The main idea of this low-rank approximation is shown on the Diagram 1 and explained in details in Section 3.3. Diagram 1 demonstrates two possible ways to find a low-rank tensor approximation of the Matérn covariance function. The first way (marked with '?') is not so trivial and the second via the Fast Fourier Transform (FFT), low-rank and the inverse FFT (IFFT) is more trivial. We use here the fact that the FT of the Matérn covariance is analytically known and has a known low-rank approximation. The IFFT can be computed numerically and does not change the tensor ranks.

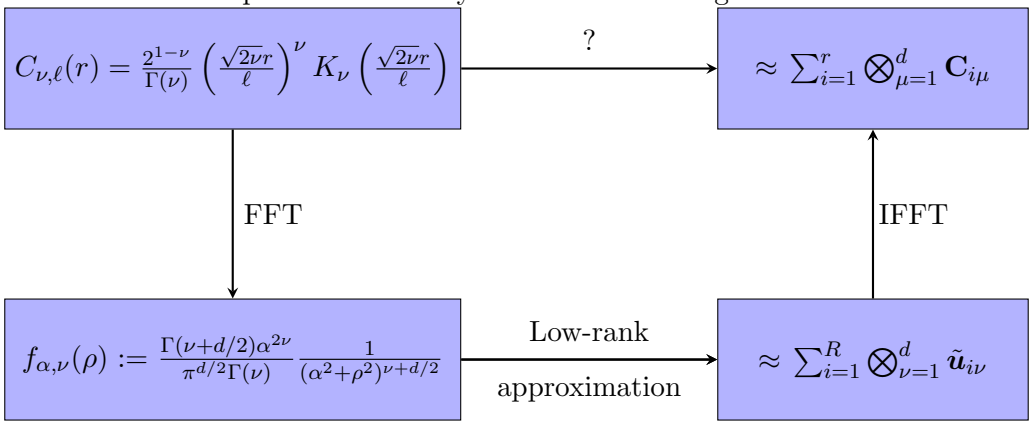

Diagram 1: Two possible ways to find a low rank tensor approximation of the Matérn covariance matrix $C_{\nu,\ell}(r)$.

The Matérn covariance function is defined as

$$C_{\nu,\ell}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}r}{\ell}\right), \tag{2.3}$$

where distance $r := \|x-y\|$, x, y two points in $\mathbb{R}^d$, $\nu > 0$ defines the smoothness of the random field. The larger ν, the smoother random field. The parameter $\ell > 0$ is a spatial range parameter that measures how quickly the correlation of the random field decays with distance, with larger ℓ corresponding to a slower decay (keeping ν fixed). $\mathcal{K}_\nu$ denotes the modified Bessel function of order ν. When $\nu = 1/2$ [54], the Matérn covariance function reduces to the exponential covariance model and describes a rough field. The value $\nu = \infty$ corresponds to a Gaussian covariance model, which describes a very smooth field, that is infinitely differentiable. Random fields with a Matérn covariance function are $\lfloor \nu - 1 \rfloor$ times mean square differentiable.

Thus, the hyperparameter ν controls the degree of smoothness.

The d-dimensional Fourier transform $\boldsymbol{F}^d(C(r,\nu)$ of the Matérn kernel, defined in Eq. (2.3), in $\mathbb{R}^d$ is given by [44]

$$\boldsymbol{U}(\xi) := \boldsymbol{F}^d(C(r,\nu) = \beta \cdot \left(1 + \frac{\ell^2}{2\nu}|\xi|^2\right)^{-\nu - d/2}, \tag{2.4}$$

where $\beta = \beta(\nu, \ell, n)$ is a constant and $|\xi|$ is the Euclidean distance in $\mathbb{R}^d$.

The following tensor approach also applies to the case of anisotropic distance, where $r^2 = \sqrt{\langle A(x-y), (x-y)\rangle}$, and A is a positive diagonal $d \times d$ matrix.

3 Low-rank tensor decompositions

In this section, we review the definitions of the CP and Tucker tensor formats. Then, we provide the analytic sinc-based proof of the existence of low-rank tensor approximations of Matérn functions. We investigate numerically the behavior of the Tucker and CP ranks across a wide range of parameters specific to the family of Matérn kernels in Eq. 2.4. The Tucker tensor format is used in this work for additional rank compression of the CP factors. There are no reliable algorithms to compute CP decomposition, which can be difficult to compute, but there are such algorithms for Tucker decomposition. The Tucker decomposition is only limited w.r.t. the available memory storage, since the term r^d in $\mathcal{O}(drn + r^d)$ grows exponentially with d.

3.1 General definitions

CP and Tucker rank-structured tensor formats have been applied for the quantitative analysis of correlation in multi-dimensional experimental data for a long time in chemometrics and signal processing [58, 8]. The Tucker tensor format was introduced in 1966 for tensor decomposition of multidimensional arrays in chemometrics [64]. Though the canonical representation of multivariate functions was introduced as early as in 1927 [27], only the Tucker tensor format provides a stable algorithm for decomposition of full-size tensors. A mathematical approval of the Tucker decomposition algorithm was presented in papers on higher order singular value decomposition (HOSVD) and the Tucker ALS algorithm for orthogonal Tucker approximation of higher order tensors [9]. For higher dimensions, the so-called Matrix Product States (MPS) (see the survey paper [56]) or the Tensor Train (TT) [52] decompositions can be applied. However, for 3D applications, the Tucker and CP tensor formats remain the best choices. The fast convergence of the Tucker decomposition was proved and demonstrated numerically for higher order tensors that arise from the discretization of linear operators and functions in $\mathbb{R}^d$ for a class of function-related tensors and Green's kernels in particular, it was found that the approximation error of the Tucker decomposition decayed exponentially in the Tucker rank [33, 35].

These results inspired the canonical-to-Tucker (C2T) and Tucker-to-canonical (T2C) decompositions for function-related tensors in the case of large input ranks, as well as the multigrid Tucker approximation [36].

A tensor of order d in a full format is defined as a multidimensional array over a d-tuple index set:

$$\mathbf{A} = [a_{i_1,\ldots,i_d}] \equiv [a(i_1,\ldots,i_d)] \in \mathbb{R}^{n_1 \times \ldots \times n_d} \text{ with } \quad i_\ell \in I_\ell := \{1,\ldots,n_\ell\}. \tag{3.1}$$

Here, $\mathbf{A}$ is an element of the linear space

$$\mathbb{V}_n = \bigotimes_{\ell=1}^{d} \mathbb{V}_\ell, \quad \mathbb{V}_\ell = \mathbb{R}^{I_\ell}$$

equipped with the Euclidean scalar product $\langle \cdot, \cdot \rangle : \mathbb{V}_n \times \mathbb{V}_n \to \mathbb{R}$, defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle := \sum_{(i_1 \ldots i_d) \in \mathcal{I}} a_{i_1 \ldots i_d} b_{i_1 \ldots i_d}, \quad \text{for } \mathbf{A},\, \mathbf{B} \in \mathbb{V}_{\mathbf{n}}.$$

Tensors with all dimensions having equal size $n_\ell = n$, $\ell = 1, \ldots d$, are called $n^{\otimes d}$ tensors. The required storage size scales exponentially with the dimension, n^d, which results in the so-called "curse of dimensionality".

To avoid exponential scaling in the dimension, the rank-structured separable representations (approximations) of the multidimensional tensors can be used. The simplest separable element is given by the rank-1 tensor,

$$\mathbf{U} = \mathbf{u}^{(1)} \otimes \ldots \otimes \mathbf{u}^{(d)} \in \mathbb{R}^{n_1 \times \ldots \times n_d},$$

with entries $u_{i_1,\ldots,i_d} = u^{(1)}_{i_1} \cdots u^{(d)}_{i_d}$, which requires only $n_1 + \ldots + n_d$ numbers for storage.

The rank-1 canonical tensor is a discrete counterpart of the separable d-variate function, which can be represented as the product of univariate functions,

$$f(x_1, x_2, \ldots, x_d) = f_1(x_1) f_2(x_2) \ldots f_d(x_d).$$

An example of the separable d-variate function is $f(x_1, x_2, x_3) = e^{(x_1+x_2+x_3)}$. Then, by discretization of this multivariate function on a tensor grid in a computational box, we obtain a canonical rank-1 tensor.

A tensor in the R-term canonical format is defined by a finite sum of rank-1 tensors (Fig. 3.1, left),

$$\mathbf{A}_c = \sum\nolimits_{k=1}^{R} \xi_k \mathbf{u}^{(1)}_k \otimes \ldots \otimes \mathbf{u}^{(d)}_k, \quad \xi_k \in \mathbb{R}, \tag{3.2}$$

where $\mathbf{u}^{(\ell)}_k \in \mathbb{R}^{n_\ell}$ are normalized vectors, and R is the canonical rank. The storage cost of this parametrization is bounded by dRn. An element $a(i_1, ..., i_d)$ of tensor $\mathbf{A} = \sum_{i=1}^{R} \bigotimes_{\nu=1}^{d} u_{i\nu}$ can be computed as

$$a(i_1, ..., i_d) = \sum_{\alpha=1}^{R} u_1(i_1, \alpha) u_2(i_2, \alpha) ... u_d(i_d, \alpha).$$

An alternative (contracted product) notation is used in computer science community:

$$\mathbf{A} = \mathbf{C} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \cdots \times_d U^{(d)}, \tag{3.3}$$

where $\mathbf{C} = diag\{c_1, ..., c_d\} \in \mathbb{R}^{R^{\otimes d}}$, and $U^{(\ell)} = [\mathbf{u}^{(\ell)}_1 \ldots \mathbf{u}^{(\ell)}_R] \in \mathbb{R}^{n_\ell \times R}$. An analogous multivariate function can be represented by a sum of univariate functions:

$$f(x_1, x_2, \ldots, x_d) = \sum_{k=1}^{R} f_{1,k}(x_1) f_{2,k}(x_2) \ldots f_{d,k}(x_2).$$

For $d \geq 3$, there are no stable algorithms to compute the canonical rank of a tensor $\mathbf{A}$, that is the minimal number R in representation (3.2), and the respective decomposition with the polynomial cost in d, i.e., the computation of the canonical decomposition is an N-P hard problem [18].

The Tucker tensor format (Fig. 3.1, right) is suitable for stable numerical decompositions with a fixed truncation threshold. We say that the tensor $\mathbf{A}$ is represented in the rank-$\mathbf{r}$ orthogonal Tucker format with the rank parameter $\mathbf{r} = (r_1, \ldots, r_d)$ if

$$\mathbf{A} = \sum_{\nu_1=1}^{r_1} \ldots \sum_{\nu_d=1}^{r_d} \beta_{\nu_1,\ldots,\nu_d} \mathbf{v}^{(1)}_{\nu_1} \otimes \ldots \otimes \mathbf{v}^{(\ell)}_{\nu_\ell} \ldots \otimes \mathbf{v}^{(d)}_{\nu_d}, \quad \ell = 1, \ldots, d, \tag{3.4}$$

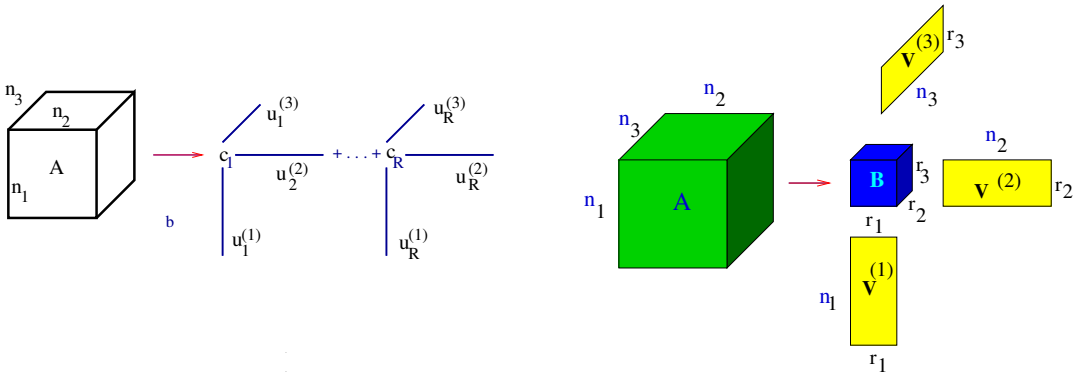

Figure 3.1: Canonical (left) and Tucker (right) decompositions of 3D tensors.

where $\{\mathbf{v}_{\nu_\ell}^{(\ell)}\}_{\nu_\ell=1}^{r_\ell} \in \mathbb{R}^{n_\ell}$ represents a set of orthonormal vectors for $\ell = 1, \ldots, d$, and $\mathbf{B} = [\mathbf{B}_{\nu_1,\ldots,\nu_d}] \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is the Tucker core tensor. The storage cost for the Tucker tensor is bounded by $drn + r^d$, with $r = |\mathbf{r}| := \max_\ell r_\ell$. Using the orthogonal side matrices $V^{(\ell)} = [v_1^{(\ell)} \ldots v_{r_\ell}^{(\ell)}]$ and contracted products, the Tucker tensor decomposition may be presented in the alternative notation,

$$\mathbf{A}_{(\mathbf{r})} = \mathbf{B} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 \ldots \times_d V^{(d)}.$$

In the case $d = 2$, the orthogonal Tucker decomposition is equivalent to the singular value decomposition (SVD) of a rectangular matrix.

3.2 Tucker decomposition of full format tensors

We use the following algorithm to compute the Tucker decomposition of the full format tensor. The most time-consuming part of the Tucker algorithm is higher order singular value decomposition (HOSVD), the computation of the initial guess for matrices $V^{(\ell)}$ using the SVD of the matrix unfolding $A_{(\ell)}$, $\ell = 1,\ 2,\ 3$ (Fig. 3.2), of the original tensor along each mode of a tensor [9]. Figure 3.2 illustrates the matrix unfolding of the full format tensor A (3.1) along the index set $I_1 = \{1, \ldots, n_1\}$.

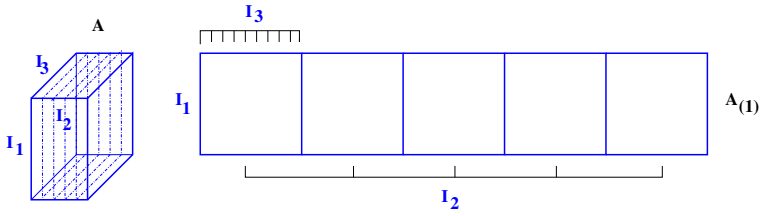

Figure 3.2: Unfolding of a 3D tensor along the mode I_ℓ with $\ell = 1$.

The second part of the algorithm is the ALS procedure. For every tensor mode, a "single-hole" tensor of reduced size is constructed by the mapping all of the modes of the original tensor except one into the subspaces $V^{(\ell)}$. Then, the subspace $V^{(\ell)}$ for the current mode is updated by SVD of the unfolding of the "single hole" tensor for this mode. This alternates over all modes of the tensor, which are updated at the current iteration of ALS. The final step of the algorithm is computation of the core tensor by using the ultimate mapping matrices from ALS.

The numerical cost of Tucker decomposition for full size tensors is dominated by the initial guess, which is estimated as $O(n^{d+1})$ when all $n_\ell = n$ are equal , or $O(n^4)$ for our 3D case. This step restricts the available size of the tensor to be decomposed since, for conventional computers, the 3D case $n_\ell > 10^2$ is the limiting case for SVD.

The multigrid Tucker algorithm for full size tensors allows the computational complexity to be linear in the full size of the tensor, $O(n^d)$, i.e. $O(n^3)$ for 3D tensors [36]. It is computed on a sequence of

diadically refined grids and is based on implementing the HOSVD only at the coarsest grid level, $n_0 \ll n$. The initial guess for the ALS procedure is computed at each refined level by the interpolation of the dominating Tucker subspaces obtained from the previous coarser grid. In this way, at fine 3D Cartesian grids, we need only $O(n^3)$ storage (to represent the initial full format tensor) to contract with the Tucker side matrices, obtained by the Tucker approximation via ALS on the previous grids.

3.3 Illustration of the idea

We describe the idea shown on Diagram 1. Let $\boldsymbol{F}^d = \bigotimes_{\nu=1}^d \boldsymbol{F}_\nu$ be the d-dimensional Fourier transform, and $\boldsymbol{F}^{-d} = \bigotimes_{\nu=1}^d \boldsymbol{F}_\nu^{-1}$ be its inverse, where $\bigotimes$ denotes the Kronecker product. We are using the fact that the Fourier and inverse Fourier Transformations do not change the Kronecker tensor rank of the argument [50]. Let us assume $\boldsymbol{U}(\xi) = \boldsymbol{F}^d(C(r,\nu))$ is known analytically and has a low-rank tensor approximation $\mathbf{U} = \sum_{j=1}^r \bigotimes_{\nu=1}^d \mathbf{u}_{j\nu}$, then by applying the inverse Fourier, we obtain a low-rank representation of the covariance function:

$$\boldsymbol{F}^{-d}(\mathbf{U}) = \left(\bigotimes_{\nu=1}^d \boldsymbol{F}_\nu^{-1}\right) \sum_{i=1}^r \left(\bigotimes_{\nu=1}^d \mathbf{u}_{\nu i}\right) = \sum_{i=1}^r \bigotimes_{\nu=1}^d \left(\boldsymbol{F}_\nu^{-1}(\mathbf{u}_{\nu i})\right) = \sum_{i=1}^r \bigotimes_{\nu=1}^d \tilde{\mathbf{u}}_{\nu i} =: C(r,\nu). \tag{3.5}$$

3.4 sinc approximation of the Matérn function

Methods of separable approximation to the 3D Newton kernel and many other spherically symmetric functions by using the Gaussian sums have been addressed in the chemical and mathematical literature since [5] and [62, 6, 22, 16], respectively. The approach to tensor decomposition for a class of lattice structured interaction potentials was presented in [31]. In this section, we recall the grid-based method for the low-rank canonical representation of a spherically symmetric kernel function $q(\|x\|)$, $x \in \mathbb{R}^d$ for $d = 2, 3, \ldots$, by its projection onto the set of piecewise constant basis functions, see [3] for the case of Newton and Yukawa kernels for $x \in \mathbb{R}^3$.

Let $d = 3$, following the standard schemes we introduce the uniform $n \times n \times n$ rectangular Cartesian grid Ω_n with mesh size $h = 2b/n$ (n even) in the computational domain $\Omega = [-b, b]^3$.

Let $\{\psi_{\mathbf{i}}\}$ be a set of tensor-product piecewise constant basis functions, $\psi_{\mathbf{i}}(\mathbf{x}) = \prod_{\ell=1}^3 \psi_{i_\ell}^{(\ell)}(x_\ell)$, for the 3-tuple index $\mathbf{i} = (i_1, i_2, i_3) \in \mathcal{I}$, $\mathcal{I} = I_1 \times I_2 \times I_3$ with $i_\ell \in I_\ell = \{1, ..., n\}$, $\ell = 1, 2, 3$. The generating kernel $q(\|x\|)$ is discretized by its projection onto the basis set $\{\psi_{\mathbf{i}}\}$ in the form of a third order tensor of size $n \times n \times n$, defined entry-wise as

$$\mathbf{Q} := [q_{\mathbf{i}}] \in \mathbb{R}^{n\times n\times n}, \quad q_{\mathbf{i}} = \int_{\mathbb{R}^3} \psi_{\mathbf{i}}(x) q(\|x\|)\, \mathrm{d}x. \tag{3.6}$$

The low-rank canonical decomposition of the 3rd order tensor $\mathbf{Q}$ is based on using exponentially convergent sinc-quadratures applied to the integral representation of the function $q(p)$, $p \in \mathbb{R}$ in the form

$$q(p) = \int_{\mathbb{R}} a_1(t) e^{-p^2 a_2(t)} dt,$$

specified by a certain weights $a_1(t), a_2(t) > 0$. Diagram 2 illustrates a scheme of the proof of existence of the canonical low-rank tensor approximation. It could be easier to apply the Laplace transform to the Fourier transform of a Matérn covariance matrix. To approximate the resulting Laplace integral we apply the sinc quadrature. The number $(2M + 1)$ of terms in the approximate sum (Eq. 3.7) is the canonical tensor rank.

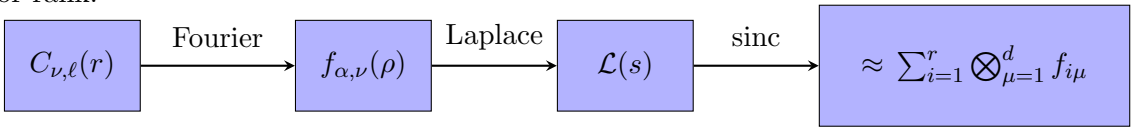

Diagram 2: Scheme of the proof of existence of low-rank tensor approximation, $r = 2M + 1$.

In particular, the sinc-quadrature for the Laplace-Gauss transform

$$q(p) = \int_{\mathbb{R}_+} a(t) e^{-t^2 p^2} \, \mathrm{d}t \approx \sum_{k=-M}^{M} a_k e^{-t_k^2 p^2} \quad \text{for} \quad |p| > 0, \quad p \in \mathbb{R}, \tag{3.7}$$

can be applied, where the quadrature points t_k and weights a_k are given by

$$t_k = k\mathfrak{h}_M, \quad a_k = a(t_k)\mathfrak{h}_M, \quad \mathfrak{h}_M = C_0 \log(M)/M, \quad C_0 > 0. \tag{3.8}$$

Under the assumption $0 < a_0 \le |p| < \infty$, this quadrature can be proven to provide an exponential convergence rate in M (uniformly in p) for a class of functions $a(z)$ analytic in a certain strip $|z| \le D$ in the complex plane, such that on the real axis the functions $a_1(t)e^{-p^2 a_2(t)}$ decay polynomially or exponentially. We refer to basic results in [62, 6, 22], where the exponential convergence of the sinc-approximation in the number of terms (i.e., the canonical rank $R = 2M + 1$) was analyzed.

Assume that the representation like (3.7) exists, for any fixed $x = (x_1, x_2, x_3) \in \mathbb{R}^3$, such that $\|x\| > a_0 > 0$, we apply the sinc-quadrature approximation (3.7), (3.8) to obtain the separable expansion

$$q(\|x\|) = \int_{\mathbb{R}_+} a(t) e^{-t^2 \|x\|^2} \, \mathrm{d}t \approx \sum_{k=-M}^{M} a_k e^{-t_k^2 \|x\|^2} = \sum_{k=-M}^{M} a_k \prod_{\ell=1}^{3} e^{-t_k^2 x_\ell^2}, \tag{3.9}$$

providing an exponential convergence rate in M,

$$\left| q(\|x\|) - \sum_{k=-M}^{M} a_k e^{-t_k^2 \|x\|^2} \right| \le \frac{C}{a} \, e^{-\beta\sqrt{M}}, \quad \text{with some } C, \beta > 0. \tag{3.10}$$

Combining (3.6) and (3.9), and taking into account the separability of the Gaussian basis functions, we arrive at the low-rank approximation to each entry of the tensor $\mathbf{Q}$,

$$q_{\mathbf{i}} \approx \sum_{k=-M}^{M} a_k \int_{\mathbb{R}^3} \psi_{\mathbf{i}}(x) e^{-t_k^2 \|x\|^2} \mathrm{d}x = \sum_{k=-M}^{M} a_k \prod_{\ell=1}^{3} \int_{\mathbb{R}} \psi_{i_\ell}^{(\ell)}(x_\ell) e^{-t_k^2 x_\ell^2} \mathrm{d}x_\ell.$$

Define the vector (recall that $a_k > 0$)

$$\mathbf{q}_k^{(\ell)} = a_k^{1/3} \left[b_{i_\ell}^{(\ell)}(t_k) \right]_{i_\ell=1}^{n_\ell} \in \mathbb{R}^{n_\ell} \quad \text{with} \quad b_{i_\ell}^{(\ell)}(t_k) = \int_{\mathbb{R}} \psi_{i_\ell}^{(\ell)}(x_\ell) e^{-t_k^2 x_\ell^2} \mathrm{d}x_\ell,$$

then the 3rd order tensor $\mathbf{Q}$ can be approximated by the R-term ($R = 2M + 1$) canonical representation

$$\mathbf{Q} \approx \mathbf{Q}_R = \sum_{k=-M}^{M} a_k \bigotimes_{\ell=1}^{3} \mathbf{b}^{(\ell)}(t_k) = \sum_{k=-M}^{M} \mathbf{q}_k^{(1)} \otimes \mathbf{q}_k^{(2)} \otimes \mathbf{q}_k^{(3)} \in \mathbb{R}^{n \times n \times n}, \tag{3.11}$$

where $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^n$. Given a threshold $\varepsilon > 0$, M can be chosen as the minimal number such that in the max-norm

$$\|\mathbf{Q} - \mathbf{Q}_R\| \le \varepsilon \|\mathbf{Q}\|.$$

The skeleton vectors can be re-numerated by $k \mapsto k' = k + M + 1$, $\mathbf{q}_k^{(\ell)} \mapsto \mathbf{q}_{k'}^{(\ell)}$, $(k' = 1, ..., R = 2M + 1)$, $\ell = 1, 2, 3$. The symmetric canonical tensor $\mathbf{Q}_R \in \mathbb{R}^{n \times n \times n}$ in (3.11) approximates the 3D symmetric kernel function $q(\|x\|)$ $(x \in \Omega)$, centered at the origin, such that $\mathbf{q}_{k'}^{(1)} = \mathbf{q}_{k'}^{(2)} = \mathbf{q}_{k'}^{(3)}$ $(k' = 1, ..., R)$.

In some applications the tensor can be given in the canonical tensor format, but with large rank R and discretized on large grids $n \times n \times \ldots \times n$, so that the computation of the initial of guess in the Tucker-ALS decomposition algorithm becomes intractable. In particular, this situation may arise in the

tensor approximation of complicated kernel functions obtained by composition of simple radial functions which can be represented in the low-rank CP format.

For such cases the canonical-to-Tucker decomposition algorithm was introduced in [36]. It is also based on the minimization ALS procedure, as the Tucker algorithm for full size tensor described in Section 3.2, but the initial guess is computed just by the singular value decomposition of the side matrices $U^{(\ell)} = [\mathbf{u}_1^{(\ell)} \dots \mathbf{u}_R^{(\ell)}] \in \mathbb{R}^{n_\ell \times R}$, $\ell = 1,\ 2,\ 3$, see (3.3). This is the so-called Reduced HOSVD (RHOSVD) which does not need the construction of the unfolding of the full tensor.

An efficient way of rank-structured representation of the multidimensional tensors is the so-called mixed tensor format introduced in [30], which combines the canonical to Tucker decomposition with the Tucker-to-canonical decomposition. It may also combine a standard Tucker decomposition with the canonical-to-Tucker decomposition to produce a canonical tensor from a full size tensor.

3.5 Laplace transform of the covariance matrix

The integral representations like (3.7) can be derived by using the Laplace transform applied either directly to the Matérn covariance function or to its spectral density (2.4).

For example, in the case of the Newton kernel we have $q(p) = 1/p$, the Laplace-Gauss transform representation takes the form

$$\frac{1}{p} = \frac{2}{\sqrt{\pi}} \int_{\mathbb{R}_+} e^{-p^2 t^2} dt, \quad \text{where} \quad p = \|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}. \tag{3.12}$$

In this case the term $\mathbf{q}_k^{(\ell)}$ equals $\mathbf{q}_{-k}^{(\ell)}$, and the sum (3.11) reduces to $k = 0, 1, ..., M$, implying $R = M + 1$. The Laplace transform representation for the Slater function $q(p) = e^{-2\sqrt{\alpha p}}$ (i.e., exponential covariance) with $p = \|x\|^2$ reads

$$q(p) = e^{-2\sqrt{\alpha p}} = \frac{\sqrt{\alpha}}{\sqrt{\pi}} \int_{\mathbb{R}_+} t^{-3/2} exp(-\alpha/t - pt) dt. \tag{3.13}$$

In the case of Matérn spectral density in (2.4) with an even dimension parameter $d = 2d_1$, $d_1 = 1, 2, ...$ and $\nu = 0, 1, 2, ...$ the Laplace transform

$$\frac{\eta!}{(p+a)^{\eta+1}} = \int_{\mathbb{R}_+} t^\eta e^{-at} e^{-pt} dt \tag{3.14}$$

can be applied after the substitution $p = \|\xi\|^2$, $q(p) = \beta \left(1 + \frac{\ell^2}{2\nu} p\right)^\eta$, $\eta = -\nu - d_1$.

If $-\eta = \nu + d_1 = 1/2, 3/2, 5/2, ..., \frac{2k+1}{2}, ...$ the Laplace transform is

$$\frac{(2\eta)!\sqrt{\pi}}{\eta! 4^\eta} \frac{1}{(p+a)^\eta \sqrt{p+a}} = \int_{\mathbb{R}_+} \frac{t^\eta}{\sqrt{t}} e^{-at} e^{-pt} dt, \quad \eta \in \mathbb{N}. \tag{3.15}$$

3.6 Covariance matrix in rank-structured tensor format

In what follows, we consider the CP approximation of the radial function $q(r)$ in the positive sector, i.e., on the domain $[0, b]^3$ (By symmetry, the canonical tensor can be extended to the whole computational domain $[-b, b]^3$.) Let the covariance function $q(r) = C_{\nu,\ell}(r)$ in (2.3) be represented on $n \times n \times n$ tensor grid $\Omega_n \subset \Omega = [0, b]^3$ by the rank-R symmetric CP tensor as described in the previous sections,

$$q(r) \mapsto \mathbb{Q} \approx \mathbb{Q}_R = \sum_{k=1}^{R} \mathbf{q}_k^{(1)} \otimes \mathbf{q}_k^{(2)} \otimes \mathbf{q}_k^{(3)} \in \mathbb{R}^{n \times n \times n}, \tag{3.16}$$

with the same skeleton vectors $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^n$ for $\ell = 1, 2, 3$.

Define the covariance matrix $\mathbf{C} = [c_{\mathbf{i},\mathbf{j}}] \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ associated with the tensor grid Ω_n entry-wise by

$$c_{\mathbf{i},\mathbf{j}} = C_{\nu,\ell}(\|x_{\mathbf{i}} - y_{\mathbf{j}}\|), \quad \mathbf{i},\mathbf{j} \in \mathcal{I}. \tag{3.17}$$

Using the tensor representation (3.16) we represent the large $n^3 \times n^3$ matrix $\mathbf{C}$ in the rank-R Kronecker (tensor) format as follows

$$\mathbf{C} \approx \mathbf{C}_R = \sum_{k=1}^{R} \mathbf{Q}_k^{(1)} \otimes \mathbf{Q}_k^{(2)} \otimes \mathbf{Q}_k^{(3)}, \tag{3.18}$$

where the symmetric Toeplitz matrix $\mathbf{Q}_k^{(\ell)} = \mathrm{Toepl}[\mathbf{q}_k^{(\ell)}] \in \mathbb{R}^{n\times n}$, $\ell = 1,2,3$, is defined by its first column which is specified by the skeleton vectors $\mathbf{q}_k^{(1)} = \mathbf{q}_k^{(2)} = \mathbf{q}_k^{(3)}$ in the decomposition (3.16).

Figure 3.3 illustrates eight selected Canonical generating vectors from $\mathbf{q}_k^{(1)}$ for $k = 1,\ldots,R$, $R = 34$, on the grid of size $n = 2049$ for the Slater function $e^{-\|x\|^p}$, which define the corresponding Toeplitz matrices.

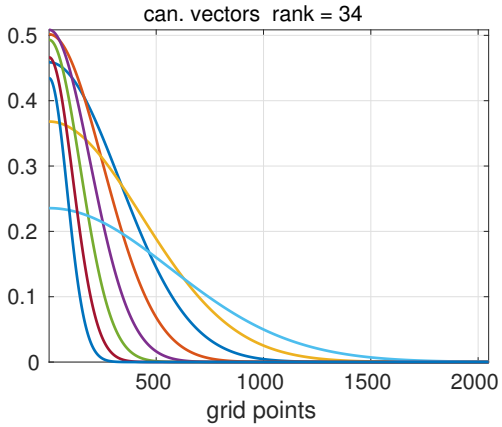

Figure 3.3: Selected eight canonical vectors from the full set $\mathbf{q}_k^{(1)}$, $k = 1,\ldots,R$, see (3.16).

Notice that a Toeplitz matrix can be multiplied by a vector in $O(n \log n)$ operations via complementing it to the circulant matrix. In general, the inverse of a Toeplitz matrix could not be calculated in the closed form as it is the case for circulant matrices which can be diagonalized by the Fourier transform.

In the rest of this section, we introduce the numerical scheme which is capable for rank-structured calculations of the analytic matrix functions $\mathcal{F}(\mathbf{C}_R)$. This scheme is based on certain specific properties of the skeleton matrices in the symmetric rank-R decomposition (3.18). We discuss the most interesting examples of the functions $\mathcal{F}_1(\mathbf{C}) = \mathbf{C}^{-1}$ and $\mathcal{F}_2(\mathbf{C}) = \mathbf{C}^{1/2}$, where we abbreviate $\mathbf{C} = \mathbf{C}_R$.

Given an s.p.d. positive definite matrix s.t. $\|\mathbf{A}\| = q < 1$, let the matrix valued function be given as exponentially fast converging series

$$\mathcal{F}(\mathbf{A}) = \mathbf{E} + a_1\mathbf{A} + a_2\mathbf{A}^2 + \ldots, \tag{3.19}$$

where the matrix $\mathbf{A}$, acting in the multi-dimensional index set, allows the low-rank Kronecker tensor decomposition. Then the computation of the low-rank tensor approximation to $\mathcal{F}(\mathbf{A})$ can be performed by the "add-and-compress" scheme, where the addition of each term the series above should be accomplished by the rank truncation algorithm in the corresponding format.

To reduce the rank-structured evaluation of $\mathcal{F}_1(\mathbf{C})$ to the described framework, we propose the special rank-structured additive splitting of the covariance matrix $\mathbf{C}$ with the easily invertible dominating part. To that end, we construct the diagonal matrix $\mathbf{Q}_0^{(1)}$ by assembling all diagonal sub-matrices in $\mathbf{Q}_k^{(1)}$, $k = 1,\ldots,R$, (in the following we skip the upper index (1)),

$$\mathbf{Q}_0 := \sum_{k=1}^{R} \mathrm{diag}(\mathbf{Q}_k), \tag{3.20}$$

and modify each Toeplitz matrix $\mathbf{Q}_k$ by subtraction of its diagonal part,

$$\mathbf{Q}_k \mapsto \widehat{\mathbf{Q}}_k := \mathbf{Q}_k - \mathrm{diag}(\mathbf{Q}_k), \quad k = 1, \ldots, R. \tag{3.21}$$

Using above defined matrices we introduce the rank $R+1$ additive splitting of $\mathbf{C}$ defined by the skeleton matrices $\mathbf{Q}_0, \widehat{\mathbf{Q}}_k$ (due to the Kronecker symmetry of $\mathbf{C}$)

$$\mathbf{C} = \mathbf{Q}_0 \otimes \mathbf{Q}_0 \otimes \mathbf{Q}_0 + \sum_{k=1}^{R} \widehat{\mathbf{Q}}_k^{(1)} \otimes \widehat{\mathbf{Q}}_k^{(2)} \otimes \widehat{\mathbf{Q}}_k^{(3)}. \tag{3.22}$$

Hence we have

$$\mathbf{C}^{-1} = \mathbf{Q}_0^{-1} \otimes \mathbf{Q}_0^{-1} \otimes \mathbf{Q}_0^{-1} (\mathbf{E} + \sum_{k=1}^{R} \mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(1)} \otimes \mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(2)} \otimes \mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(3)})^{-1}. \tag{3.23}$$

Likewise, since $\mathbf{Q}_0 \otimes \mathbf{Q}_0 \otimes \mathbf{Q}_0$ is a scaled identity, we obtain

$$\mathbf{C}^{1/2} = \mathbf{Q}_0^{1/2} \otimes \mathbf{Q}_0^{1/2} \otimes \mathbf{Q}_0^{1/2} (\mathbf{E} + \sum_{k=1}^{R} \mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(1)} \otimes \mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(2)} \otimes \mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(3)})^{1/2}. \tag{3.24}$$

Assume that $\|\mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(1)}\| < 1$ for $k = 1, \ldots, R$ in some norm, then we are in a position to apply the "add-and-compress" scheme described above.

In the following example, we illustrate the presented computational scheme. We consider the covariance matrix $\mathbf{C}_R$ obtained by the rank-R sinc approximation to the Slater function $e^{-\|x\|^p}$ with $R = 40$ on the grid with $n = 1025$ sampling points. Figure 3.4 demonstrates the decay in both the matrix norms $\mathbf{Q}_k^{(1)}$ (left), and in the scaled preconditioned modified matrices $\mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(1)}$, $k = 1, \ldots, R$ (right). The scaling factor is chosen as $1/n$. The right figure indicates that the analytic matrix functions $\mathcal{F}(\mathbf{C}_R)$ can be evaluated by using exponentially fast convergent power series supported via add-and-compress strategy for the control of tensor rank.

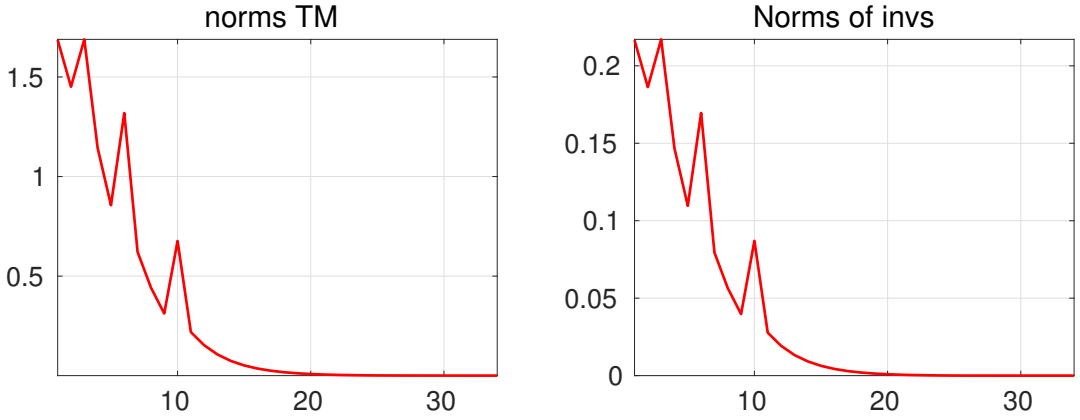

Figure 3.4: Scaled norms $\|\mathbf{Q}_k^{(1)}\|$ (left) and $\|\mathbf{Q}_0^{-1}\widehat{\mathbf{Q}}_k^{(1)}\|$ (right) vs. $k = 1, \ldots, R$.

Notice that in the Kriging calculations (Task 3 above) the low-rank tensor structure in the covariance matrix $\mathbf{C}_R$ can be directly utilized if the sampling points in the Kriging algorithm form the smaller $m_1 \times m_2 \times m_3$ tensor sub-grid of the initial $n \times n \times n$ tensor grid Ω_n with $m_\ell < n$. The same argument applies to the evaluation of conditional covariance. In the general case of "non-tensor" location of sampling points some mixed tensor factorizations could be applied.

3.7 Numerical illustrations

In what follows, we check the low-rank Tucker tensor approximation to p-Slater function $C(x) = e^{-\|x\|^p}$, and for some particular examples of the Matérn kernels by using full grid tensor representation. We demonstrate the fast exponential convergence of the tensor approximation in the Tucker rank. The functions were sampled on the $n_1 \times n_2 \times n_3$ 3D Cartesian grid with $n_\ell = 100$, $\ell = 1, 2, 3$.

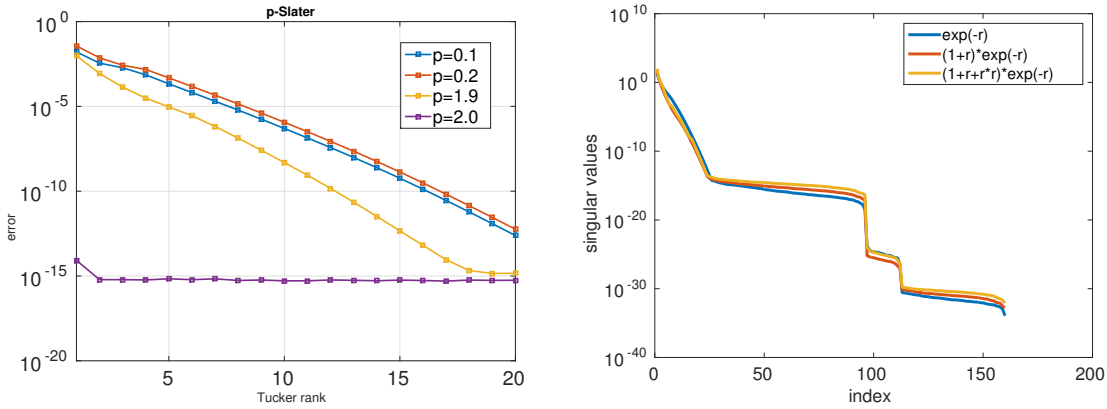

Figure 3.5: Convergence in the Frobenius error (3.26) w.r.t. the Tucker rank for the function (3.27) with $p = 0.1,\ 0.2,\ 1.9,\ 2.0$ (left); Decay of singular values of the weighted Slater function (right).

For a given continuous function $q : \Omega \to \mathbb{R}$, $\Omega := \prod_{\ell=1}^{d}[-b_\ell, b_\ell] \subset \mathbb{R}^d$, $0 < b_\ell < \infty$, we introduce the collocation-type function related tensor of order d by

$$\mathbf{Q} \equiv \mathbf{Q}(q) := [q_{i_1 \ldots i_d}] \in \mathbb{R}^{I_1 \times \ldots \times I_d} \text{ with } q_{i_1 \ldots i_d} := q(x_{i_1}^{(1)}, \ldots, x_{i_d}^{(d)}),$$

where $(x_{i_1}^{(1)}, \ldots, x_{i_d}^{(d)}) \in \mathbb{R}^d$ are grid collocation points, indexed by $\mathcal{I} = I_1 \times \ldots \times I_d$,

$$x_{i_\ell}^{(\ell)} = -b_\ell + (i_\ell - 1)h_\ell, \quad i_\ell = 1, 2, \ldots, n_\ell,\ \ell = 1, \ldots, d, \tag{3.25}$$

which are the nodes of equally spaced subintervals with the mesh size $h_\ell = 2b_\ell/(n_\ell - 1)$.

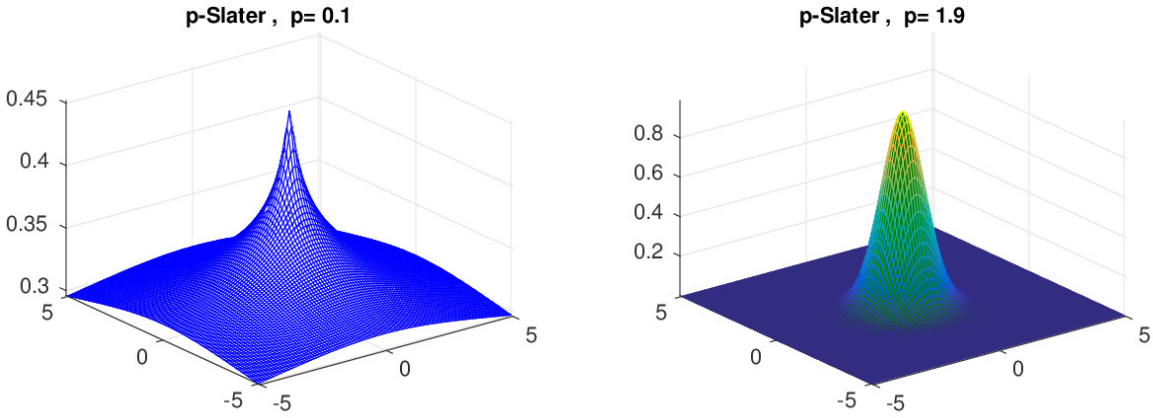

Figure 3.6: Cross section of the 3D radial function (3.27) with $p = 0.1$ (left) and $p = 1.9$ (right) at level z=0.

For the p-Slater discretized on the 3D Cartesian grid we tested the convergence of the error in relative Frobenius norm with respect to the Tucker rank for p-Slater functions with $p = 0.1,\ 0.2,\ 1.9,\ 2.0$. The Frobenius norm is computed as

$$E_{FN} = \frac{\|\mathbf{Q} - \mathbf{Q}_{(r)}\|}{\|\mathbf{Q}\|}, \tag{3.26}$$

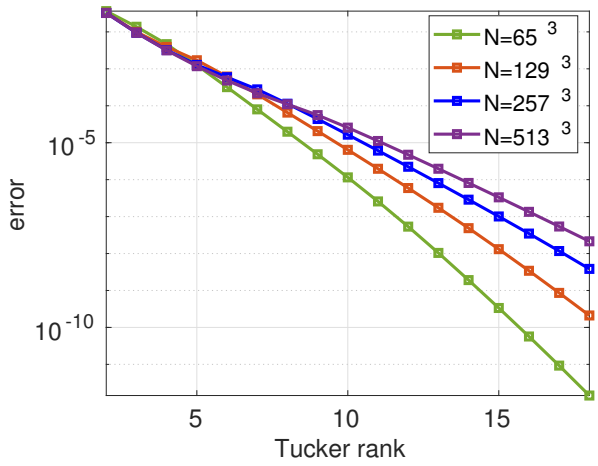

Figure 3.7: Multigrid Tucker: convergence w.r.t. Tucker ranks of a Slater function with $p = 1$ on a sequence of grids.

where $\mathbf{Q}_{(r)}$ is the tensor reconstructed from the Tucker rank-r decomposition of $\mathbf{Q}$.

Figure 3.5 shows convergence of the Frobenius error with respect to the Tucker rank for the "p-Slater" function

$$C(x, y) = e^{-\|x-y\|^p} \tag{3.27}$$

for different values of the parameter p. Figure 3.6 visualizes these functions for $p = 0.1$ and $p = 1.9$, presenting function values on the plane at the level $z = 0$. Figure 3.7 shows for a Slater function with $p = 1$ the dependence of the Tucker decomposition error in Frobenius norm (3.26) on the Tucker rank for increasing grid parameter n.

Figure 3.8 shows the convergence with respect to the Tucker rank for spectral density of Matérn covariance,

$$f_{\alpha,\nu}(\rho) := \frac{C}{(\alpha^2 + \rho^2)^{\nu+d/2}}, \tag{3.28}$$

where $\alpha \in (0.1, 100)$ and $d = 1, 2, 3$. Figures demonstrate a strong dependence of the Tucker decomposition rank on the parameter α and rather weak dependence on the parameter ν. Figure 3.9 visualizes the 3D Matérn functions with the parameters $\nu = 0.4$, $\alpha = 0.1$ (left) and $\nu = 0.4$, $\alpha = 100$ (right), showing the function on the plane at $z = 0$.

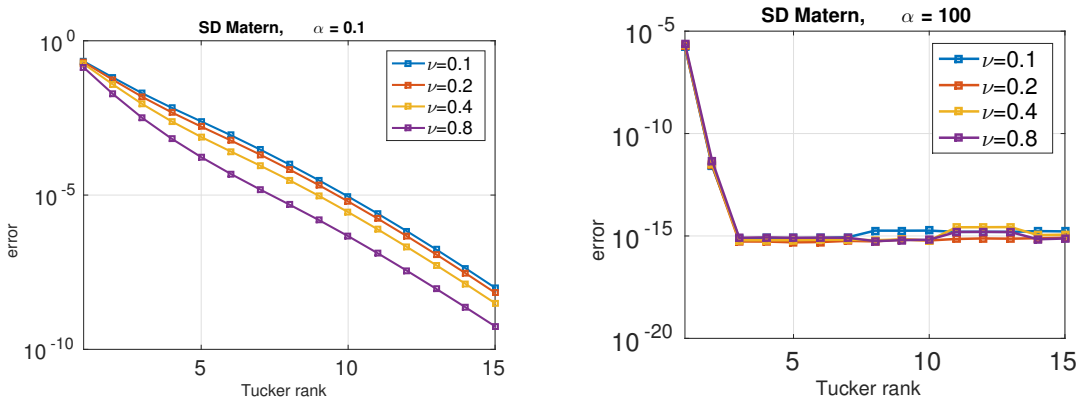

Figure 3.8: Convergence w.r.t the Tucker rank of 3D spectral density of Matérn covariance (3.28) with $\alpha = 0.1$ (left) and $\alpha = 100$ (right).

These numerical experiments demonstrate good algebraic separability of the typical multidimensional functions used in the spatial statistics, thus motivating the application of the low-rank tensor decomposition methods in multidimensional problems of statistical data analysis.

Figure 3.9 illustrates the shape of 3D spectral density of Matérn covariance (3.28) with $\alpha = 0.1$ (left) and $\alpha = 100$ (right).

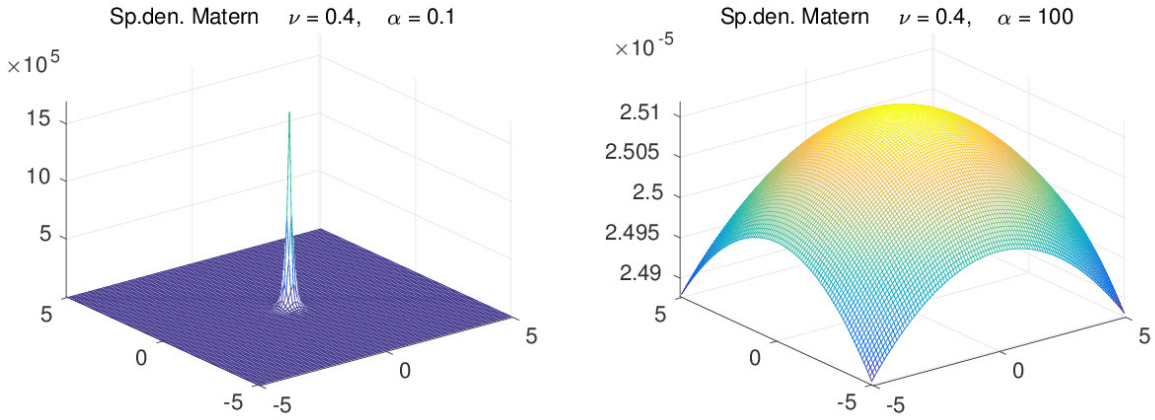

Figure 3.9: The shape of 3D spectral density of Matérn covariance (3.28) with $\alpha = 0.1$ (left) and $\alpha = 100$ (right).

4 Solution of typical tasks in low-rank tensor format

This section contains the second main contribution of this work. We provide solutions to the typical statistical questions listed above in the motivation. After each task, we add a lemma, which states the new computing and storage costs. Let $N = n^d$, measurement vector $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{C}_{ss} \in \mathbb{R}^{N\times N}$, $\mathbf{C}_{zz} \in \mathbb{R}^{m\times m}$, $\mathbf{C}_{sz} \in \mathbb{R}^{N\times m}$. We also introduce restriction operator P, which consists of only ones and zeros and pick up sub-indices $\{i_1, ..., i_m\}$ from the whole index set $\{i_1, ..., i_N\}$. This operator has tensor-1 structure, i.e., $P = \bigotimes_{\nu=1}^{d} P_\nu$. An application of this restriction tensor does not change tensor ranks.

Computing matrix-vector product: Let $\mathbf{C} = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\nu}$. If $\mathbf{z}$ is separable, i.e., $\|\mathbf{z} - \sum_{j=1}^{r_b} \bigotimes_{\nu=1}^{d} \mathbf{z}_{j\nu}\| \le \varepsilon$, then

$$\mathbf{C}\mathbf{z} = \sum_{i=1}^{r} \sum_{j=1}^{r_z} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}\mathbf{z}_{j\mu}. \tag{4.1}$$

If $\mathbf{z}$ is non-separable, then one can not employ low-rank tensor properties and should try to apply either the FFT idea [50] or the hierarchical matrix technique [37, 21, 19].

Lemma 4.1 *The computing cost of the product* $\mathbf{C}\mathbf{z}$ *is reduced from* $\mathcal{O}(N^2)$ *to* $\mathcal{O}(rr_z dn^2)$*, where* $N = n^d$*,* $d \ge 1$*.*

Trace, diagonal and determinant of C:

Lemma 4.2 *Let* $\mathbf{C} \approx \tilde{\mathbf{C}} = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}$*, then*

$$\operatorname{diag}(\tilde{\mathbf{C}}) = \operatorname{diag}\left(\sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}\right) = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \operatorname{diag}\left(\mathbf{C}_{i\mu}\right), \tag{4.2}$$

$$\operatorname{trace}(\tilde{\mathbf{C}}) = \operatorname{trace}\left(\sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}\right) = \sum_{i=1}^{r} \prod_{\mu=1}^{d} \operatorname{trace}(\mathbf{C}_{i\mu}), \tag{4.3}$$

and for the determinant it holds only for $r = 1$

$$\det(\tilde{\mathbf{C}}) = \det\left(\bigotimes_{\mu=1}^{d} \mathbf{C}_\mu\right) = \prod_{\mu=1}^{d} \det(\mathbf{C}_\mu). \tag{4.4}$$

Proof: follows from properties of the Kronecker tensors. The cost of computing the right-hand sides in Eq. 4.2-Eq. 4.4 is rdn, where $\mathbf{C}_{i\mu} \in \mathbb{R}^{n\times n}$. For simplicity, we assume that $n_1 = n_2 = ... = n_d = n$ and $\sum_{i=1}^{d} n_i = dn$.

Lemma 4.3 *The computing cost of* $\mathrm{diag}(\mathbf{C})$ *and* $\mathrm{trace}(\mathbf{C})$ *is reduced from* $\mathcal{O}(N)$ *to* $\mathcal{O}(rdn)$. *The cost of* $\det(\mathbf{C})$ *is reduced from* $\mathcal{O}(N^3)$ *to* $\mathcal{O}(dn^3)$.

Example 4.1 *A simple Matlab test for computing* $\mathrm{trace}(\mathbf{C})$ *on a working station with 128GB shows the computing times (in sec.) as in Table 4.1:*

	$n = 100$	$n = 500$	$n = 1000$
$d = 1000$	3.7	67	491

Table 4.1: Computing time (in sec.) to set up and to compute the trace of $\tilde{\mathbf{C}} = \sum_{j=1}^{r} \bigotimes_{\nu=1}^{d} \mathbf{C}_{j\nu}$, $r = 10$, $\tilde{\mathbf{C}} \in \mathbb{R}^{N\times N}$, where $N = n^d$, $d = 1000$ and $n = \{100, 500, 1000\}$. A modern desktop computer with 128 GB RAM was used.

In what follows, we discuss the computation of trace and diag in the case of Tucker representation of the generating Matérn function on $(2k+1)^{\otimes d}$ grid and present the corresponding numerical example for $d = 3$. We notice that due to Toeplitz structure of the skeleton matrices of size $(2k+1)\times(2k+1)$, the diagonal of the covariance matrix $\mathbf{C}$ is the weighted Tucker sum of the Kronecker products of scaled identity matrices in $\mathbb{R}^{2k+1}$. The scaling factor is determined by the value of generating Tucker vector $\mathbf{v}_{\nu_\ell}^{(\ell)}(k+1)$ corresponding the origin of the computational box $[-b, b]^d$. Let the matrix $\mathbf{C}$ be composed by using the Tucker tensor $\mathbf{A}$ approximating the the generating Matérn function. Then, as a straightforward consequence of the above remark, we derive the simple representations

$$\mathrm{diag}(\mathbf{C}) = \mathbf{A}(x_0)\mathbf{E}^{(d)}, \quad \mathrm{trace}(\mathbf{C}) = \mathbf{A}(x_0)(2k+1)^d,$$

where x_0 corresponds to the origin $x = 0$ in $[-b, b]^d$ and $\mathbf{E}^{(d)}$ is the identity matrix in the full tensor space. The grid coordinate of x_0 is determined by the multi-index $(k+1, ..., k+1)$.

Next table represents the values of $\mathbf{A}(x_0)$ computed by the Tucker approximation to the 3D Slater function $e^{-\|x\|}$ on the $(2k+1)^{\otimes 3}$ grids with $n = 2k+1 = 129, 257, 513$, and for different Tucker rank parameters $r = 1, 2, ..., 10$.

Tucker rank r	1	2	3	4	5	6	7	8	9	10
n=129	0.386	0.20	0.12	0.07	0.04	0.017	0.002	1.2e-4	8.4e-6	7.5e-6
n=257	0.386	0.20	0.12	0.073	0.046	0.029	0.017	0.007	8.0e-4	1.4e-5
n=513	0.386	0.20	0.12	0.073	0.047	0.031	0.020	0.0138	0.008	0.0035

Table 4.2: The error of the Tucker approximation to the value $\mathbf{A}$ at the origin (the exact value is equal to 1) versus the Tucker rank and the grid size $n = 2k+1$.

Given the rank-$\mathbf{r}$ Tucker tensor $\mathbf{A}$, the complexity for calculation of $\mathbf{A}(x_0)$ is estimated by $O(r^d)$.

Computing square root $\mathbf{C}^{1/2}$: $\mathbf{C}^{1/2}$ can be computed as in (3.24). An iterative method for computing $\mathbf{C}^{1/2}$ is presented in [15].

Linear solvers in a low-rank tensor format: Likely, there is already a good theory for solving linear systems $\mathbf{Cw} = \mathbf{z}$ with symmetric and positive definite matrix $\mathbf{C}$, in a tensor format. We refer to the overview works [32, 34, 17, 20]. Some particular linear solvers are developed in [12, 2, 23, 35, 11, 10, 35]. We also recommend to use the Tensor Toolbox [40], which contains routines for CP and Tucker tensor formats.

4.1 Computing $\mathbf{z}^T\mathbf{C}^{-1}\mathbf{z}$

Lemma 4.4 *Let $\|\mathbf{z} - \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{z}_{i\mu}\| \le \varepsilon$. Assume there is an iterative method that can be used to solve the linear system $\mathbf{Cw} = \mathbf{z}$ in a low-rank tensor format and to find the solution in a form $\mathbf{w} = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{w}_{i\mu}$. Then, the quadratic form $\mathbf{z}^T\mathbf{C}^{-1}\mathbf{z}$ is the following scalar products:*

$$\mathbf{z}^T\mathbf{C}^{-1}\mathbf{z} = \sum_{i=1}^{r}\sum_{j=1}^{r_z}\prod_{\mu=1}^{d}(\mathbf{w}_{i\mu}, \mathbf{z}_{j\mu}) \tag{4.5}$$

Proof: Follows from definition and properties of tensor and scalar products.
If $\mathbf{z}$ is non-separable, then one can not employ low-rank tensor properties and should try to apply either the FFT idea [50] or the Hierarchical matrix technique [37] (for $d \le 3$).

Lemma 4.5 *The computing cost of the quadratic form $\mathbf{z}^T\mathbf{C}^{-1}\mathbf{z}$ is a product of the number of required iterations #iters and the cost of one iteration, which is $\mathcal{O}(rr_zdm^2)$ (assuming that the iterative method required only matrix-vector products).*

Proof: Follows from definitions and properties of the tensor product and the scalar product.

4.2 The interpolation by simple Kriging

The three computationally most demanding tasks in Kriging are : (1) solving an $M \times M$ system of equations to obtain the Kriging weights, (2) obtaining the $N \times 1$ Kriging estimate from superposition of the Kriging weights with the $N \times M$ cross-covariance matrix between measurements and unknowns, and (3) evaluating the $N \times 1$ estimation variance as the diagonal of an $N \times N$ conditional covariance matrix [50]. Here, M refers to the number of measured data values, and N to the number of estimation points. For the optimal design of sampling patterns, the challenge is to evaluate scalar measures of the $N \times N$ conditional covariance matrix (see φ_A and φ_C in Eq. 2.1 and Task 4), repeatedly within a high-dimensional and non-linear optimization procedure (e.g., [41, 57]).

The following kriging formula is well-known [50]

$$\hat{\mathbf{s}} = \mathbf{C}_{sz}\mathbf{C}_{zz}^{-1}\mathbf{z}. \tag{4.6}$$

Let $N = n^d$ and $M = m^d$.

Lemma 4.6 *Let the settings in Lemma 4.4 hold. If $\|\mathbf{C}_{sz} - \sum_{i=1}^{r_C} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}\| \le \varepsilon$, for some small $\varepsilon \ge 0$, and Lemma 4.4 holds, then*

$$\mathbf{C}_{sz}\mathbf{C}_{zz}^{-1}\mathbf{z} \approx \sum_{i=1}^{r_z}\sum_{j=1}^{r_C}\bigotimes_{\nu=1}^{d}\mathbf{C}_{i\nu}\mathbf{w}_{j\nu}. \tag{4.7}$$

Proof: Follows from definition and properties of tensor product and the scalar product.

Lemma 4.7 *The computing cost of solving the linear system to compute $\mathbf{C}_{zz}^{-1}\mathbf{z}$ is $\mathcal{O}(\#iters \cdot r_zrdm^2)$. To compute the Kriging coefficients, Eq. 4.7, is $\mathcal{O}(r_zr_Cdnm) + \mathcal{O}(\#iters \cdot r_zrdm^2)$.*

If $\mathbf{z}$ is non-separable, then one can not employ low-rank tensor properties and should try to apply either the FFT idea [50] or the Hierarchical matrix technique [37].

4.3 Computing conditional covariance

Let $\mathbf{y} \in \mathbb{R}^m$ be the vector of measurements. The conditional covariance matrix is

$$\mathbf{C}_{ss|y} = \mathbf{C}_{ss} - \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys}. \tag{4.8}$$

The associated estimation variance $\hat{\boldsymbol{\sigma}}$ is the diagonal of the $N \times N$ conditional covariance matrix $\mathbf{C}_{ss|y}$

$$\hat{\boldsymbol{\sigma}}_{\mathbf{s}} = \mathrm{diag}(\mathbf{C}_{ss|y}) = \mathrm{diag}\left(\mathbf{C}_{ss} - \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys}\right). \tag{4.9}$$

We assume that the measurements are taken in locations, which are forming a subset of the total set of nodes $\mathcal{I} = \{0, ..., N-1\}$, i.e., $\mathcal{I}_{\mathcal{M}} = \{i_1, ..., i_m\} \subset \mathcal{I}$. We also assume that $\mathcal{I}_{\mathcal{M}}$ belong to a tensor mesh. If $\mathcal{I} = \bigotimes_{\nu=1}^{d} I_\nu$, then $\mathcal{I}_{\mathcal{M}} = \bigotimes_{\nu=1}^{d} \hat{I}_\nu$, $\hat{I}_\nu \subseteq \mathcal{I}$.

Let $\mathbf{C}_{yy} = \sum_{k=1}^{r} \bigotimes_{\mu=1}^{d} \mathbf{C}_{k\mu}$. We can again use low-rank tensor solvers to solve the matrix system $\mathbf{C}_{yy}\mathbf{W} = \mathbf{C}_{ys}$. We obtain the solution $\mathbf{W} = \mathbf{C}_{yy}^{-1}\mathbf{C}_{ys} = \sum_{j=1}^{r_w} \bigotimes_{\mu=1}^{d} \mathbf{C}_{j\mu}$. Then, assuming again $\mathbf{C}_{sy} \approx \sum_{i=1}^{r_C} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}$, obtain

$$\mathbf{C}_{sy}\mathbf{W} = \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys} = \sum_{i=1}^{r_C} \bigotimes_{\nu=1}^{d} \mathbf{C}_{i\nu} \sum_{j=1}^{r_w} \bigotimes_{\mu=1}^{d} \mathbf{C}_{j\mu} = \sum_{i=1}^{r_C}\sum_{j=1}^{r_w} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}\mathbf{C}_{j\mu} \approx \sum_{j=1}^{r_0} \bigotimes_{\mu=1}^{d} \tilde{\mathbf{C}}_{j\mu}, \tag{4.10}$$

where $r_0 > 0$ is a new rank after a rank truncation procedure and $\tilde{\mathbf{C}}_{j\mu}$ new factors. Let $\mathbf{C}_{ss} = \sum_{i=1}^{r_s} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu}$. The conditional covariance will be:

$$\mathbf{C}_{ss|y} = \sum_{i=1}^{r_s} \bigotimes_{\mu=1}^{d} \mathbf{C}_{i\mu} - \sum_{i=1}^{r_0} \bigotimes_{\mu=1}^{d} \tilde{\mathbf{C}}_{i\mu} = \sum_{i=1}^{r_s+r_0} \bigotimes_{\mu=1}^{d} \hat{\mathbf{C}}_{i\mu}, \tag{4.11}$$

where $\hat{\mathbf{C}}_{i\mu} = \mathbf{C}_{i\mu}$ for $1 \le i \le r_s$ and $\hat{\mathbf{C}}_{i\mu} = -\tilde{\mathbf{C}}_{i\nu}$ for $r_s < i \le r_0 + r_s$.

4.4 Example. Separable covariance matrices

Let $\mathrm{cov}(\mathbf{x}, \mathbf{y}) = \exp^{-|\mathbf{x}-\mathbf{y}|^2}$ be the Gaussian covariance function, where $\mathbf{x} = (x_1, .., x_d)$, $\mathbf{y} = (y_1, ..., y_d) \in \mathcal{D} \subset \mathbb{R}^d$. $\mathrm{cov}(\mathbf{x}, \mathbf{y})$ can be written as a tensor product of 1D functions

$$\mathrm{cov}(\mathbf{x}, \mathbf{y}) = \exp^{-|x_1-y_1|^2} \otimes \ldots \otimes \exp^{-|x_d-y_d|^2}.$$

After discretization $\mathrm{cov}(\mathbf{x}, \mathbf{y})$, obtain $\mathbf{C}$ as a rank-1 Kronecker product of 1D-dimensional covariance matrices, i.e.,

$$\mathbf{C} = \mathbf{C}_1 \otimes ... \otimes \mathbf{C}_d. \tag{4.12}$$

Note, that one may have arbitrary discretization in each directions (anisotropy).

Lemma 4.8 *If d Cholesky decompositions exist, i.e, $\mathbf{C}_i = \mathbf{L}_i \cdot \mathbf{L}_i^T$, $i = 1..d$. Then*

$$\mathbf{C}_1 \otimes ... \otimes \mathbf{C}_d = (\mathbf{L}_1\mathbf{L}_1^T) \otimes ... \otimes (\mathbf{L}_d\mathbf{L}_d^T) = (\mathbf{L}_1 \otimes ... \otimes \mathbf{L}_d) \cdot (\mathbf{L}_1^T \otimes ... \otimes \mathbf{L}_d^T) =: \mathbf{L} \cdot \mathbf{L}^T, \tag{4.13}$$

where $\mathbf{L} := \mathbf{L}_1 \otimes ... \otimes \mathbf{L}_d$, $\mathbf{L}^T := \mathbf{L}_1^T \otimes ... \otimes \mathbf{L}_d^T$ are also low- and upper-triangular matrices.

Lemma 4.8 shows that the Gaussian covariance function in dimensions $d > 1$ can be a) written as a tensor sum of one-dimensional covariance functions and b) its Cholesky factor can be computed via Cholesky factors, computed from one-dimensional covariances. The computational complexity drops from $\mathcal{O}(N \log N)$, $N = n^d$, to $\mathcal{O}(dn \log n)$, where n is number of mesh points in one-dimensional problem. Further research is required for non-Gaussian covariance functions.

Lemma 4.9 *Let* $\mathbf{C} = \mathbf{C}_1 \otimes ... \otimes \mathbf{C}_d$. *If the inverse matrices* $\mathbf{C}_i^{-1}$, $i = 1..d$, *exist, then*

$$(\mathbf{C}_1 \otimes ... \otimes \mathbf{C}_d)^{-1} = \mathbf{C}_1^{-1} \otimes ... \otimes \mathbf{C}_d^{-1}. \tag{4.14}$$

The computational complexity drops from $\mathcal{O}(N \log N)$, $N = n^d$, to $\mathcal{O}(dn \log n)$, where n is the number of mesh points in an one-dimensional problem.

Remark 4.10 *We assumed here that we have an efficient method to invert* $\mathbf{C}$ *(e.g., FFT or hierarchical matrices) with the cost* $\mathcal{O}(N \log N)$. *If not, then the complexity cost drops from* $\mathcal{O}(N^3)$ *to* $\mathcal{O}(dn^3)$.

Lemma 4.11 *Let* $\mathbf{C}_i$, $i = 1..d$, *are covariance matrices. Then*

$$\log\det(\mathbf{C}_1 \otimes \ldots \otimes \mathbf{C}_d) = \sum_{j=1}^{d} \log\det \mathbf{C}_j \prod_{i=1, i\neq j}^{d} n_i. \tag{4.15}$$

Proof: We check for $d = 2$: $\det(\mathbf{C}_1 \otimes \mathbf{C}_2) = \det(\mathbf{C}_1)^{n_2} \cdot \det(\mathbf{C}_2)^{n_1}$ and then apply mathematical induction.

Lemma 4.11 tells us how to compute $\log\det \mathbf{C}$, where $\mathbf{C}$ is a separable rank-1 d-dimensional covariance function. The cost drops again from $\mathcal{O}(N \log N)$, $N = n^d$, to $\mathcal{O}(dn \log n)$. A similar assumption as in Remark 4.10 for computing $\det(\mathbf{C})$ also holds here.

Example 4.2 *Let* $n = 6000$, $d = 3$, $N = 6000^3$. *The Matlab time to setup matrices* $\mathbf{C}_1$,$\mathbf{C}_2$ *and* $\mathbf{C}_3$ *takes 11 second and to compute* $\mathbf{L}_1$, $\mathbf{L}_2$ *and* $\mathbf{L}_3$ *takes 4 seconds on usual MacBookPro with 16 GB RAM. Large matrices* $\mathbf{C}$ *and* $\mathbf{L}$ *are never constructed (i.e., the Kronecker product is never calculated).*

Example 4.3 *(combining the hierarchical matrix technique and the Kronecker tensor product) In [42] we showed how to use the hierarchical matrix technique to approximate* $\mathbf{C}_\nu$ *and its Cholesky factor* $\mathbf{L}_\nu$ *for* $n = 2 \cdot 10^6$ *in 2 minutes. Assuming* $\mathbf{C} = \mathbf{C}_1 \otimes \ldots \otimes \mathbf{C}_d$, *we can approximate* $\mathbf{C}$ *for* $n = (2 \cdot 10^6)^d$ *also in few minutes. Such fine resolution could be useful for describing multi-scale phenomena.*

Lemma 4.12 *Let* $\mathbf{C}$ *be as in Eq. 4.12. Let Lemma 4.4 hold. Then, applying the property in Eq. 4.15, we obtain a tensor approximation of the log-likelihood:*

$$\mathcal{L} \approx \tilde{\mathcal{L}} = -\frac{\prod_{\nu=1}^{d} n_\nu}{\log(2\pi)} - \sum_{j=1}^{d} \log\det \mathbf{C}_j \prod_{i=1, i\neq j}^{d} n_i - \sum_{i=1}^{r}\sum_{j=1}^{r} \prod_{\nu=1}^{d} (\mathbf{u}_{i,\nu}^T, \mathbf{u}_{j,\nu}). \tag{4.16}$$

Equation 4.16 shows a disadvantage of the Gaussian log-likelihood function in high dimensions. Namely, the log-likelihood grows exponentially with d as n^d.

5 Conclusion

In this work, we demonstrate that the basic functions and operators used in spatial statistics may be represented using rank-structured tensor formats and that the error of this representation exhibits the exponential decay with respect to the tensor rank. We applied the Tucker and canonical tensor decompositions to a family of Matérn- and Slater-type functions with varying parameters and demonstrated numerically that their approximations exhibit exponentially fast convergence. A low-rank tensor approximation of the Matérn covariance function and its Fourier transform is considered. We separated the radial basis functions using Laplace transforms to prove the existence of such a low-rank approximation, and applied the sinc quadrature to estimate the tensor ranks and accuracy.

We also demonstrated how to compute $\mathrm{diag}(\mathbf{C})$, $\mathrm{trace}(\mathbf{C})$, the matrix-vector product, Kriging operations, and the geostatistical optimal design in a low-rank tensor format with at a linear cost. Operations

such as computing the Cholesky factorization, inverse, and determinant can only be rewritten for rank-1 tensors (e.g., the Gaussian covariance has tensor rank-1). These formulas could be useful for developing successive rank-1 updates in greedy algorithms. Further investigations are needed for representation these quantities with ranks higher than one.

Additionally, in Section 3.7 we studies the influence of the parameters of the Matérn covariance function on the tensor ranks (Figures 3.5 and 3.7). We observed (see Fig. 3.8) that the dependence of the parameters of the Matérn covariance function on the tensor ranks is very weak, and the ranks grew slowly. In this paper, we also highlighted that big data statistical problems can be effectively treated by using special low-rank tensor techniques.

References

[1] S. Ambikasaran, J. Y. Li, P. K. Kitanidis, and E. Darve, *Large-scale stochastic linear inversion using hierarchical matrices*, Computational Geosciences, 17 (2013), pp. 913–927.

[2] J. Ballani and D. Kressner, *Sparse inverse covariance estimation with hierarchical matrices*, http://sma.epfl.ch/∼anchpcommon/publications/quic_ballani_kressner_2014.pdf, (2015).

[3] C. Bertoglio and B. N. Khoromskij, *Low-rank quadrature-based tensor approximation of the Galerkin projected Newton/Yukawa kernels*, Comp. Phys. Comm., 183 (2012), pp. 904–912.

[4] S. Börm and J. Garcke, *Approximating gaussian processes with H^2-matrices*, in Proceedings of 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. ECML 2007, J. N. Kok, J. Koronacki, R. L. de Mantaras, S. Matwin, D. Mladen, and A. Skowron, eds., vol. 4701, 2007, pp. 42–53.

[5] S. F. Boys, G. B. Cook, C. M. Reeves, and I. Shavitt, *Automatic fundamental calculations of molecular structure*, Nature, 178 (1956), pp. 1207–1209.

[6] D. Braess, *Nonlinear approximation theory*, Springer-Verlag, 1986.

[7] J.-P. Chiles and P. Delfiner, *Geostatistics: modeling spatial uncertainty*, vol. 497, John Wiley & Sons, 2009.

[8] A. Cichocki, , and S. Sh. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, Wiley, 2002.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[10] S. Dolgov, B. N. Khoromskij, A. Litvinenko, and H. G. Matthies, *Computation of the response surface in the tensor train data format*, arXiv preprint arXiv:1406.2816, (2014).

[11] S. Dolgov, B. N. Khoromskij, A. Litvinenko, and H. G. Matthies, *Polynomial chaos expansion of random coefficients and the solution of stochastic partial differential equations in the tensor train format*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 1109–1135.

[12] S. V. Dolgov, B. N. Khoromskij, and D. V. Savostyanov, *Superfast Fourier transform using QTT approximation*, J. Chem. Phys., 18 (2012), pp. 915–953.

[13] P. A. Finke, D. J. Brus, M. F. P. Bierkens, T. Hoogland, M. Knotters, and F. D. Vries, *Mapping groundwater dynamics using multiple sources of exhaustive high resolution data*, Geoderma, 123 (2004), pp. 23–39.

[14] R. Furrer and M. G. Genton, *Aggregation-cokriging for highly-multivariate spatial data*, Biometrika, 98 (2011), pp. 615–631.

[15] I. Gavrilyuk, W. Hackbusch, and B. Khoromskij, *Data-sparse approximation to a class of operator-valued functions*, Mathematics of computation, 74 (2005), pp. 681–708.

[16] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, *Hierarchical tensor-product approximation to the inverse and related operators in high-dimensional elliptic problems*, Computing, 74 (2005), pp. 131–157.

[17] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen, 36 (2013), pp. 53–78.

[18] J. Håstad, *Tensor rank is np complete*, J. Algorithms, 11 (1990), pp. 644–654.

[19] W. Hackbusch, *A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices*, Computing, 62 (1999), pp. 89–108.

[20] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer Series in Computational Mathematics, Springer Verlag, 2012.

[21] W. Hackbusch, *Hierarchical matrices: Algorithms and Analysis*, vol. 49 of Springer Series in Comp. Math., Springer, 2015.

[22] W. Hackbusch and B. Khoromskij, *Low-rank Kronecker product approximation to multi-dimensional nonlocal operators. Part I. Separable approximation of multi-variate functions*, Computing, 76 (2006), pp. 177–202.

[23] W. Hackbusch and B. N. Khoromskij, *Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. II. HKT representation of certain operators*, Computing, 76 (2006), pp. 203–225.

[24] M. S. Handcock and M. L. Stein, *A Bayesian analysis of kriging*, Technometrics, 35 (1993), pp. 403–410.

[25] H. Harbrecht, M. Peters, and M. Siebenmorgen, *Efficient approximation of random fields for numerical applications*, Numerical Linear Algebra with Applications, (2015).

[26] M. R. Haylock, N. Hofstra, A. M. K. Tank, E. J. Klok, P. D. Jones, and M. New, *A european daily high-resolution gridded data set of surface temperature and precipitation for 1950–2006*, J. Geophys. Res, 113 (2008), p. D20119.

[27] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Physics, 6 (1927), pp. 164–189.

[28] S. D. Iaco, S. Maggio, M. Palma, and D. Posa, *Toward an automatic procedure for modeling multivariate space-time data*, Computers & Geosciences, 41 (2011), pp. 1–11, doi:10.1016/j.cageo.2011.08.008.

[29] A. G. Journel and C. J. Huijbregts, *Mining Geostatistics*, Academic Press, New York, 1978.

[30] V. Khoromskaia, *Numerical Solution of the Hartree-Fock Equation by Multilevel Tensor-structured methods*, PhD thesis, Technical University of Berlin, 2010.

[31] V. Khoromskaia and B. N. Khoromskij, *Fast tensor method for summation of long-range potentials on 3D lattices with defects*, Numer. Lin. Alg. Appl., 23 (2016), pp. 249–271.

[32] B. Khoromskij, *Tensors-structured numerical methods in scientific computing: Survey on recent advances*, Chemometrics and Intelligent Laboratory Systems, 110 (2011), pp. 1–19.

[33] B. N. Khoromskij, *Structured rank-$(r_1, \dots, r_d)$ decomposition of function-related operators in $\mathbb{R}^d$*, Comp. Meth. Appl. Math, 6 (2006), pp. 194–220.

[34] ———, *Tensor Numerical Methods for Multidimensional PDEs: Basic Theory and Initial Applications*, ESAIM: Proceedings and Surveys, N. Champagnat, T. Leliévre, A. Nouy, eds, 48 (January 2015), pp. 1–28.

[35] B. N. Khoromskij and V. Khoromskaia, *Low rank Tucker-type tensor approximation to classical potentials*, Cent. Eur. J. Math., 5 (2007), pp. 523–550 (electronic).

[36] ———, *Multigrid accelerated tensor approximation of function related multidimensional arrays*, SIAM J. Sci. Comp., 31 (2009), pp. 3002–3026.

[37] B. N. Khoromskij, A. Litvinenko, and H. G. Matthies, *Application of hierarchical matrices for computing the Karhunen–Loève expansion*, Computing, 84 (2009), pp. 49–67.

[38] P. K. Kitanidis, *Introduction to Geostatistics*, Cambridge University Press, Cambridge, 1997.

[39] T. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.

[40] T. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

[41] J. B. Kollat, P. M. Reed, and J. R. Kasprzyk, *A new epsilon-dominance hierarchical bayesian optimization algorithm for large multiobjective monitoring network design problems*, Adv. in Water Res., 31 (2008), pp. 828 – 845.

[42] A. Litvinenko, *HLIBCov: Parallel Hierarchical Matrix Approximation of Large Covariance Matrices and Likelihoods with Applications in Parameter Identification*, ArXiv 1709.08625, (2017).

[43] A. Litvinenko, Y. Sun, M. G. Genton, and D. Keyes, *Likelihood Approximation With Hierarchical Matrices For Large Spatial Datasets*, ArXiv 1709.04419, (2017).

[44] B. Matérn, *Spatial Variation*, vol. 36 of Lecture Notes in Statistics, Springer-Verlag, Berlin; New York, second edition ed., 1986.

[45] G. Matheron, *The Theory of Regionalized Variables and Its Applications*, Ecole de Mines, Fontainebleau, France, 1971.

[46] V. Minden, A. Damle, K. L. Ho, and L. Ying, *Fast spatial gaussian process maximum likelihood estimation via skeletonization factorizations*, arXiv:1603.08057v1, (2016).

[47] W. G. Müller, *Collecting spatial data. Optimum design of experiments for random fields*, Springer, Berlin, Germany, 3rd ed., 2007.

[48] G. R. North, J. Wang, and M. G. Genton, *Correlation models for temperature fields*, Journal of Climate, 24 (2011), pp. 5850–5862.

[49] W. Nowak, *Measures of parameter uncertainty in geostatistical estimation and geostatistical optimal design*, Math. Geosciences, 42 (2010), pp. 199–221. doi:10.1007/s11004-009-9245-1.

[50] W. Nowak and A. Litvinenko, *Kriging and spatial design accelerated by orders of magnitude: combining low-rank covariance approximations with fft-techniques*, Mathematical Geosciences, 45 (2013), pp. 411–435.

[51] D. Nychka, S. Bandyopadhyay, D. Hammerling, F. Lindgren, and S. Sain, *A multiresolution gaussian process model for the analysis of large spatial datasets*, Journal of Computational and Graphical Statistics, 24 (2015), pp. 579–599.

[52] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comp., 33 (2011), pp. 2295–2317.

[53] J. Quinonero-candela, C. E. Rasmussen, and R. Herbrich, *A unifying view of sparse approximate gaussian process regression*, Journal of Machine Learning Research, 6 (2005), p. 2005.

[54] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning: www.gaussianprocess.org/gpml/chapters/*, vol. 497, MIT Press, 2006.

[55] A. K. Saibaba, S. Ambikasaran, J. Yue Li, P. Kitanidis, and E. Darve, *Application of hierarchical matrices to linear inverse problems in geostatistics*, Oil & Gas Science and Technology–Rev. IFP Energies nouvelles, 67 (2012), pp. 857–875.

[56] U. Schollwöck, *The density-matrix renormalization group in the age of matrix product states*, Ann.Phys., 51 (2011), pp. 96–192.

[57] R. Shah and P. M. Reed, *Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems*, European Journal of Operational Research, 211 (2011), pp. 466 – 479.

[58] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis with applications in the chemical sciences*, Wiley, 2004.

[59] G. Spöck and J. Pilz, *Spatial sampling design and covariance-robust minimax prediction based on convex design ideas*, Stochastic Environmental Research and Risk Assessment, 24 (2010), pp. 463–482. doi:10.1007/s00477-009-0334-y.

[60] M. L. Stein, J. Chen, and M. Anitescu, *Difference filter preconditioning for large covariance matrices*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 52–72.

[61] M. L. Stein, Z. Chi, and L. J. Welty, *Approximating likelihoods for large spatial data sets*, Journal of the Royal Statistical Society, Series B, 66 (2004), pp. 275–296.

[62] F. Stenger, *Numerical methods based on sinc and analytic functions*, vol. 20 of Springer Series in Computational Mathematics, Springer-Verlag, New York, 1993.

[63] Y. Sun and M. L. Stein, *Statistically and computationally efficient estimating equations for large spatial datasets*, Journal of Computational and Graphical Statistics, 25 (2016), pp. 187–208.

[64] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[65] S. M. Wesson and G. G. S. Pegram, *Radar rainfall image repair techniques*, Hydrological and Earth Systems Sciences, 8 (2004), pp. 8220–8234.