

KRIGING IN TENSOR TRAIN DATA FORMAT

Sergey Dolgov¹, Alexander Litvinenko², and Dishi Liu³

¹ University of Bath
Claverton Down, Bath, BA2 7AY, United Kingdom
e-mail: s.dolgov@bath.ac.uk

² RWTH Aachen
Kackertstr. 9C, 52072, Aachen, Germany
e-mail: litvinenko@uq.rwth-aachen.de

³ Institute of Scientific Computing, Technische Universität Braunschweig
Mühlenpfordtstrasse 23, D-38106 Braunschweig, Germany.
e-mail: d.liu@tu-bs.de

Abstract. *Combination of low-tensor rank techniques and the Fast Fourier transform (FFT) based methods had turned out to be prominent in accelerating various statistical operations such as Kriging, computing conditional covariance, geostatistical optimal design, and others. However, the approximation of a full tensor by its low-rank format can be computationally formidable. In this work, we incorporate the robust Tensor Train (TT) approximation of covariance matrices and the efficient TT-Cross algorithm into the FFT-based Kriging. It is shown that here the computational complexity of Kriging is reduced to $\mathcal{O}(dr^3n)$, where n is the mode size of the estimation grid, d is the number of variables (the dimension), and r is the rank of the TT approximation of the covariance matrix. For many popular covariance functions the TT rank r remains stable for increasing n and d . The advantages of this approach against those using plain FFT are demonstrated in synthetic and real data examples.*

Keywords: low-rank tensor approximation; tensor train; geostatistical estimation; geostatistical optimal design, kriging, circulant, Toeplitz, FFT

This paper is dedicated to our wonderful colleague Prof. Hermann G. Matthies on the occasion of his 68th birth anniversary.

Contents

1	Introduction	2
1.1	State of the art for FFT-based Kriging	3
1.2	Goals, approach and contributions	4
1.3	Notation	5
2	Kriging and geostatistical optimal design	5
2.1	Matérn covariance	5
2.2	Computational tasks in Kriging and optimal sampling design	5
3	Interface from Kriging to FFT-based methods	6
3.1	Embedding Toeplitz covariance to circulant matrices	6
3.2	Injection, embedding and extraction of data tensors	7
3.3	Matrix-vector multiplication via FFT	7
4	FFT-based Kriging accelerated by low-rank tensor decomposition	8
4.1	TT decomposition	8
4.2	TT-cross approximation	9
4.3	TT representation of general and structured matrices	11
4.4	Kriging operations in TT format	12
4.4.1	Small number of scattered samples	12
4.4.2	Large number of structured samples	13
5	Numerical tests	13
5.1	Kriging of daily moisture data	13
5.2	High-dimensional field generation: computational benchmark	14
6	Discussion and Conclusions	16

1 Introduction

Kriging is an interpolation method that makes estimates of unmeasured quantities based on (sparse) scattered measurements. It is widely applied in the estimation of some spatially distributed quantities such as daily moisture, rainfall intensities, temperatures, contaminant concentrations or hydraulic conductivities, etc. [40, 22]. Kriging is also used as a surrogate of some complex physical models for the purpose of efficient uncertainty quantification (UQ), in which it estimates the model response under some random perturbation of the parameters. In the first case the estimation grids are usually in two or three dimensions [60, 9, 18] or four dimensions in a space-time Kriging [3, 34, 21], while in the latter the dimension number could be much larger (equals to the number of uncertain parameters). When considering finely resolved estimation grids (which is often the case for UQ jobs), Kriging can easily exceed the computational capacity of modern computers. In this case estimation variance of Kriging or solving the related geostatistical optimal design problems incurs even higher computational costs [41, 43, 55]. Kriging mainly involves three computational tasks. The first is solving a $N \times N$ system of equations to obtain the

Kriging weights, where N is the number of measurements. Despite its $\mathcal{O}(N^3)$ complexity this task is better manageable since N is usually much smaller than the number of estimates on a fine grid, $\bar{N} = \bar{n}^d$, d the dimensionality, especially when the measurement is expensive like for complex physical models. The second task is to compute the \bar{N} Kriging estimates by multiplying the weights vector to the $\bar{N} \times N$ cross-covariance matrix between measurements and unknowns. The third task is to evaluate the \bar{N} estimation variances as the diagonal of a $\bar{N} \times \bar{N}$ conditional covariance matrix. If we take the optimal design of sampling into account, there is an additional task to repeatedly evaluate the $\bar{N} \times \bar{N}$ conditional covariance matrix for the purpose of a high-dimensional non-linear optimization [32, 54, 51].

Remarkable progress had been made in speeding up Kriging computations by Fast Fourier transform (FFT) [11]. The low-rank tensor decomposition techniques brought a further possible reduction in the time cost, since d -dimensional FFT on a tensor in low-rank format can be made at the cost of a series of 1-dimensional FFT's, as exemplified in [59] by using canonical, Tucker and Tensor Train formats of tensors. The work in [44] brought a significant further reduction of computational cost for the second and third Kriging tasks as well as the task for the optimal design of sampling by applying a low-rank canonical tensor approximation to the vectors of interest.

In this paper, we enhance the methodology proposed in [44] by employing a more robust low-rank Tensor Train (TT) format instead of the canonical format. We apply the TT-cross algorithm for efficient approximation of tensors, which is a key improvement compared to the method introduced in [44] where the low-rank format of the covariance matrix was assumed to be given. We also consider a more broad Matérn class of covariance functions.

The current work improves the applicability of the use of low-rank techniques in the FFT-based Kriging. We achieve a reduction of the computational complexity of Kriging to the level of $\mathcal{O}(dr^3\bar{n})$, where r is the considered TT rank of the approximation, and \bar{n} is the number of grid points in *one* direction, such that $\bar{N} = \bar{n}^d$ is the total number of estimated points.

We assume second-order stationarity for the covariance function and simple Kriging on a rectangular, equispaced grid parallel to the axes.

We also discuss possible extensions to non-rectangular domains and to general (scattered) measurement points. In such cases, the tensor ranks may significantly increase, up to the full rank. For the cases when FFT technique is not applicable the authors of [52, 37, 35, 29] applied the hierarchical matrix technique (\mathcal{H} -matrices). A parallel implementation of Kriging was done in [50].

1.1 State of the art for FFT-based Kriging

Let us assume that the covariance function is second-order stationary and is discretized on a tensor (regular and equispaced grid) mesh with $\bar{N} = \bar{n}^d$ points. Then the $\bar{N} \times \bar{N}$ auto-covariance matrix of the unknowns has a symmetric (block-) Toeplitz structure (Section 3.1), which can be extended to a (block-) circulant matrix by a periodic embedding in which the number of rows and columns is enlarged, for example, from \bar{N} to $\tilde{N} = 2\bar{N} + 1$ [49, 22, 31]. It is known [11] that only the first column of the circulant matrix has to be stored. This reduces the computing cost from quadratic to log-linear [61] in \bar{N} . The key in the FFT-based Kriging is the fact that the multiplication of a circulant matrix and a vector is a discrete convolution which can be computed swiftly through FFT algorithm

so that the quadratic computational complexity is also reduced to a log-linear one [12].

If the measurements are given on a regular equispaced grid, the first Kriging task is solving a system also with a symmetric positive-definite Toeplitz matrix [11, 4]. Further development of methods handling measurements that are on a subset of a finer regular grid have been made in [49, 11].

The work in [44] combined the power of FFT and the low-rank canonical tensor decomposition. It was assumed that the covariance matrix and the vector of interest (of size \tilde{N}) are available in a low-rank canonical tensor format which is a sum of r Kronecker products of vectors of size \tilde{n} each, with $\tilde{n}^d = \tilde{N}$. Separable covariance functions (e.g. Gaussian, separate exponential) can be decomposed exactly with $r = 1$. For smooth non-separable covariance functions, a small r value can usually give a good approximation.

The canonical tensor representation can not only greatly reduce the memory storage size of the circulant matrix, but also speed up the Fourier transform since the d -dimensional FFT applied on the Kronecker product of matrices can be implemented by computing the 1-dimensional FFT on the first direction of each matrix. This reduces the complexity to $\mathcal{O}(dr\tilde{n}\log\tilde{n})$. For $r \ll \tilde{n}$ this is a significant reduction from the complexity of FFT on the full tensor, which is $\mathcal{O}(d\tilde{n}^d\log\tilde{n})$.

1.2 Goals, approach and contributions

However, converting a full tensor to a well approximating low-rank tensor format can be computationally formidable. Simply generating the full tensor itself might be beyond the memory capacity of a desktop computer. To make the low-rank FFT-based method practical, we need an efficient way to obtain a low-rank approximation directly from the multi-dimensional function that underlies the full tensor. It could be a challenging task though to approximate the first column of the Toeplitz (circulant) matrix in the canonical tensor format for $d \geq 3$. This is due to the fact that the class of rank- k canonical tensors is a nonclosed set in the corresponding tensor product space (pp 91-92 in [28]). The Tucker format tensor decomposition [27, 17, 15] adopted in [36] could be too costly to use for problems with $d \geq 3$.

In this paper, we adopt an alternative tensor format, namely, the Tensor Train (TT) format [47, 17] (introduced in Section 4.1) which can be obtained from a full tensor in a stable direct way by a sequence of singular value decompositions of auxiliary matrices, or, more importantly, it can be computed iteratively by the *TT-cross* method [48] which has the complexity in the order of $\mathcal{O}(dr^3\bar{n})$, see Section 4.2 for more details. Often this is the most time-consuming stage of Kriging operations. Once the tensors are approximated in the TT format, the FFT can be carried out with a modest $\mathcal{O}(dr^2\bar{n}\log\bar{n})$ complexity. This makes the overall low-rank FFT-based Kriging practical for high dimensions. We test the efficiency of the method in terms of computational time and memory usage in Section 5.

Thus, our paper is novel in three aspects: (i) we approximate the covariance matrix in the low-rank TT tensor format using only the given covariance function as a black box (this part was missing in [44]), (ii) we extend the methodology to Matérn, exponential and spherical covariance functions (in addition to Gaussian functions), and (iii) we demonstrate that the low-rank approach enables high-dimensional Kriging.

1.3 Notation

We denote vectors by bold lower-case letters (e.g., \mathbf{c} , \mathbf{u} , $\boldsymbol{\xi}$) and matrices by bold upper-case letters (e.g., \mathbf{C}_{ss} , \mathbf{M} , \mathbf{H}). Letters decorated with an overbar represent the size of the tensor grid of estimates. Embedded matrices, vectors and their sizes are denoted by letters with a check accent (e.g., $\check{\mathbf{C}}$, $\check{\mathbf{c}}$, \check{n} , \check{n}_i). $\mathcal{F}^{[d]}$ stands for d -dimensional Fourier transform (FT), \mathcal{F}_i for one-dimensional FT along the i -th dimension. $\mathcal{F}^{[-d]}$ and \mathcal{F}_i^{-1} are their inverse operators.

2 Kriging and geostatistical optimal design

Like in [44], we work with the *function estimate form* [30, 31] of Kriging (introduced in Section 2.2). We take simple Kriging in which the estimates are assumed to have zero mean.

2.1 Matérn covariance

A low-rank approximation of the given function or a data set is a key component of the tasks formulated above. Among of the many covariance models available, the Matérn family [39] is widely used in spatial statistics and geostatistics.

The Matérn covariance function is defined as

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

Here $r := \|p_1 - p_2\|$ is the distance between two points p_1 and p_2 in \mathbb{R}^d ; $\nu > 0$ defines the smoothness. The larger is parameter ν , the smoother is the random field. The parameter $\ell > 0$ is called the covariance length and measures how quickly the correlation of the random field decays with distance. K_ν denotes the modified Bessel function of order ν . It is known that setting $\nu = 1/2$ we obtain the exponential covariance model. The value $\nu = \infty$ corresponds to a Gaussian covariance model.

In [36], the authors provided the analytic sinc-based proof of the existence of low-rank tensor approximations of Matérn functions. They investigated numerically the behavior of the Tucker and canonical ranks across a wide range of parameters specific to the family of Matérn kernels. It could be problematic to extend the results of this work to $d > 3$, since one of the terms in the Tucker decomposition storage cost $\mathcal{O}(drn + r^d)$ is growing exponentially with d .

2.2 Computational tasks in Kriging and optimal sampling design

The computation of a simple Kriging process and optimal sample design involve mainly these tasks:

Task-1. Let \mathbf{y} denote a N -size vector containing the sampled values, \mathbf{C}_{yy} denote the auto-covariance matrix. If the measurements are not exact and the covariance matrix \mathbf{R} of the random measurement error is available, \mathbf{R} is to be added to \mathbf{C}_{yy} . The first task is to solve the below system for the Kriging weights $\boldsymbol{\xi}$:

$$\mathbf{C}_{yy}\boldsymbol{\xi} = \mathbf{y} \quad (2)$$

Task-2. With the weights $\boldsymbol{\xi}$ we can obtain the Kriging estimates $\hat{\mathbf{s}}$ (sized $\bar{N} \times 1$) by a superposition of columns of the cross-covariance matrices \mathbf{C}_{sy} (sized $\bar{N} \times N$) weighted

by $\boldsymbol{\xi}$, i.e. the Kriging estimate $\hat{\mathbf{s}}$ is given by [31]:

$$\hat{\mathbf{s}} = \mathbf{C}_{sy}\boldsymbol{\xi}. \quad (3)$$

Task-3. The variance $\hat{\sigma}_{\mathbf{s}}^2$ of the estimates $\hat{\mathbf{s}}$ is to be obtained from the diagonal of the conditional covariance matrix $\mathbf{C}_{ss|y}$:

$$\begin{aligned} \hat{\sigma}_{\mathbf{s}}^2 = \text{diag}(\mathbf{C}_{ss|y}) &= \text{diag}(\mathbf{C}_{ss} - \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys}) \\ &= \text{diag}(\mathbf{C}_{ss}) - \sum_{i=1}^N (\mathbf{C}_{sy}\boldsymbol{\zeta}_i)^{\circ 2}, \end{aligned} \quad (4)$$

where $\boldsymbol{\zeta}_i$ is the i -th column of \mathbf{L}^{-T} with \mathbf{L} the lower triangular Cholesky factor matrix of \mathbf{C}_{yy} , and the superscript $\circ 2$ denotes Hadamard square.

Task-4. The goal of geostatistical design is to optimize sampling patterns (or locations) for \mathbf{y} . There two most common objective functions to be minimized, which are also called A - and C - criteria of geostatistical optimal design [41, 43, 5]:

$$\begin{aligned} \phi_A &= \bar{N}^{-1} \text{trace}[\mathbf{C}_{ss|y}] \\ \phi_C &= \mathbf{z}^T \mathbf{C}_{ss|y} \mathbf{z} = \mathbf{z}^T (\mathbf{C}_{ss} - \mathbf{C}_{sy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys}) \mathbf{z}, \end{aligned} \quad (5)$$

where \mathbf{z} is a data vector [43].

3 Interface from Kriging to FFT-based methods

In this section we give a brief introduction to the basics of FFT-based Kriging [11]. We assume that the measurement points are a subset of the estimate grid points. The simplest version of Kriging is a direct *injection*: the estimated values are set equal to the measurement values at the corresponding locations, and to zeros at all other points. Equivalently, we say that we inject a (small) tensor of measurements into a (larger) tensor of estimations.

For the FFT-based Kriging we use a regular, equispaced grid which leads to a (block) Toeplitz covariance matrix that can be augmented to a circulant one (Section 3.1). An embedding operation augments the injected tensor to the size that is compatible with the circulant covariance matrix. The (pseudo-)inverse of embedding is called extraction (Section 3.2).

3.1 Embedding Toeplitz covariance to circulant matrices

A Toeplitz matrix is constant along each descending diagonal (from left to right). A block Toeplitz matrix has identical sub-matrices in each descending diagonal block and each sub-matrix Toeplitz. If the covariance function is stationary and the estimates are made on a d -dimensional regular, equispaced grid, the covariance matrix \mathbf{C}_{ss} is symmetric level- d block Toeplitz [2]. Since submatrices are repeating along diagonals the required storage could be reduced from $\mathcal{O}(\bar{N}^2)$ to $\mathcal{O}(\bar{N})$ elements [61, 23].

A circulant matrix $\tilde{\mathbf{C}}$ is a Toeplitz matrix that has its first column $\tilde{\mathbf{c}}$ periodic. This type of matrices come from covariance functions that are periodic in the domain. A circulant matrix-vector product can be computed efficiently by FFT [57]. The eigenvalues of $\tilde{\mathbf{C}}$ can be computed as the Fourier transform of its first column $\tilde{\mathbf{c}}$ [58, 2, pp. 350-354]. These properties lead us to the fast FFT-based kriging methods.

A Toeplitz matrix \mathbf{C}_{ss} can always be augmented to a circulant matrix $\check{\mathbf{C}}$. This process is called *embedding*. Let $\mathbf{C}(:, 1)$ be the first column of \mathbf{C}_{ss} . Embedding is often done by appending the second through the last but one element of $\mathbf{C}(:, 1)$ to the end of $\mathbf{C}(:, 1)$ in reverse order, which makes a periodic vector $\check{\mathbf{c}}$. For the cases $d > 1$, this augmentation has to be done recursively in every level for the d -level Toeplitz covariance matrix. An equivalent way of doing this is to augment the domain (to be 2^d times larger) and extend the covariance function to be periodic on the domain, as illustrated in [33, 45]. In [42, 6, 45] the authors have addressed the issue of the minimum embedding size.

3.2 Injection, embedding and extraction of data tensors

Suppose we obtained the Kriging weights ξ for the measurements by solving (2). The injection of ξ means to insert it in a larger all-zero tensor that has the same size of the estimate tensor, i.e. the *injected* tensor has non-zero entries only at the measurement sites.

Suppose we have N measurements indexed by $j = 1, \dots, N$, each associated with a weight ξ_j and a site index vector ℓ_j , then the injection of ξ results in a tensor $\bar{\xi} \in \mathbb{R}^{\bar{n}_1 \times \bar{n}_2 \times \dots \times \bar{n}_d}$ with entries:

$$\bar{\xi}(i_1, i_2, \dots, i_d) = \begin{cases} \xi_j & \text{if } \mathbf{i} = \ell_j, \forall j \in [1, \dots, N] \\ 0 & \text{otherwise} \end{cases} . \quad (6)$$

We denote the injection operation by $\mathcal{H} : \xi \rightarrow \bar{\xi}$.

Embedding an injected weight tensor enhances its mode size from \bar{n} to $\check{n} = 2\bar{n}$ by padding zeros to the extra entries so that the tensor is of 2^d times the original size. The embedded weight tensor $\check{\xi} \in \mathbb{R}^{\check{n}_1 \times \check{n}_2 \times \dots \times \check{n}_d}$ has entries:

$$\check{\xi}(i_1, i_2, \dots, i_d) = \begin{cases} \bar{\xi}(i_1, i_2, \dots, i_d) & \text{if } i_\ell \leq \bar{n}_\ell, 1 \leq \ell \leq d \\ 0 & \text{otherwise} \end{cases} . \quad (7)$$

We denote the embedding operation by $\mathcal{M} : \bar{\xi} \rightarrow \check{\xi}$.

The extraction is the inverse operation of embedding, we denoted it by \mathcal{M}^\dagger . By $\mathcal{M}^\dagger(\eta)$ we take only the first half of η in every dimension, which results in a new tensor of only $\frac{1}{2^d}$ of the size of η .

3.3 Matrix-vector multiplication via FFT

With the circulant covariance matrix $\check{\mathbf{C}}$ obtained as explained in Section 3.1, the Task-2 in (3) becomes a discrete convolution which can be computed by using FFT[57], this is written as (e.g., Fritz, Nowak and Neuweiler, [11]):

$$\begin{aligned} \mathbf{C}_{sy}\xi &= \mathbf{C}_{ss}\mathcal{H}(\xi) = \mathcal{M}^\dagger\check{\mathbf{C}}\mathcal{M}(\mathcal{H}(\xi)) \\ &= \mathcal{M}^\dagger\mathcal{F}^{[-d]}(\mathcal{F}^{[d]}(\check{\mathbf{c}}) \circ \mathcal{F}^{[d]}(\check{\xi})) . \end{aligned} \quad (8)$$

where the operation $\mathcal{M}(\mathcal{H}(\cdot))$ injects and embeds ξ into $\check{\xi}$. The $\mathcal{F}^{[d]}$ is evaluated by the Fast Fourier Transformation (FFT) [10]. Without using tensor approximations the computational complexity for Kriging is reduced to $\mathcal{O}(\check{N} \log \check{N})$, and the storage size reduced to $\mathcal{O}(\check{N})$.

For the variance estimation (Task-3) in (4) the FFT method also applies. We first need to do a Cholesky decomposition $\mathbf{C}_{yy} = \mathbf{L}\mathbf{L}^T$, and inject and embed each column ζ_i of

\mathbf{L}^{-T} to get the corresponding $\check{\boldsymbol{\zeta}}_i$. Then (4) can be computed as

$$\hat{\boldsymbol{\sigma}}_s^2 = \sigma_s^2 \mathbf{1}_{\bar{N}} - \sum_{i=1}^N [\mathcal{M}^\dagger \mathcal{F}^{[-d]} (\mathcal{F}^{[d]} (\check{\mathbf{c}}) \circ \mathcal{F}^{[d]} (\check{\boldsymbol{\zeta}}_i))]^{\circ 2}, \quad (9)$$

where σ_s^2 is the prior variance, $\mathbf{1}_{\bar{N}}$ is a \bar{N} -length vector of all ones.

4 FFT-based Kriging accelerated by low-rank tensor decomposition

In addition to the efficient FFT-based method enabled by the Teopltz structure of covariance matrices, the Kriging process can be further sped up by low-rank representations of the embedded covariance matrices. Since the covariance functions are usually smooth, large covariance matrices could be well approximated by a low-rank tensor format. A literature survey of low-rank tensor approximation techniques is available in [27, 15].

In this section, we approximate the first column of the circulant covariance matrix in tensor train (TT) format and then rewrite 8 also in the TT format. We start with a brief reviewing of the TT technique.

4.1 TT decomposition

We assume that the data vectors (\mathbf{c} , $\boldsymbol{\xi}$, etc.) can be associated to a function discretised on a structured grid in d dimensions, for example, if $\xi(x, y, z)$ is sampled on a Cartesian 3-dimensional grid,

$$\boldsymbol{\xi} = \{\xi(x_{i_1}, y_{i_2}, z_{i_3})\}_{i_1, i_2, i_3=1}^{n_1, n_2, n_3}. \quad (10)$$

Then we can enumerate the entries of the vector via sub-indices i_1, i_2, \dots, i_d , thereby seeing it as a *tensor* with elements $\boldsymbol{\xi}(i_1, \dots, i_d)$. We approximate such tensors, and, consequently, associated data vectors, in the Tensor Train (TT) decomposition [47],

$$\boldsymbol{\xi}(i_1, i_2, \dots, i_d) \approx \tilde{\boldsymbol{\xi}}(i_1, i_2, \dots, i_d) := \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} \xi_{\alpha_0, \alpha_1}^{(1)}(i_1) \xi_{\alpha_1, \alpha_2}^{(2)}(i_2) \cdots \xi_{\alpha_{d-1}, \alpha_d}^{(d)}(i_d). \quad (11)$$

Here $\xi^{(k)}$, $k = 1, \dots, d$, are called *TT blocks*. Each TT block $\xi^{(k)}$ is a three-dimensional tensor of size $r_{k-1} \times n_k \times r_k$, $r_0 = r_d = 1$. The efficiency of this representation relies on the *TT ranks* r_0, \dots, r_d being bounded by a moderate constant r . For simplicity we can also introduce an upper bound of the univariate grid sizes $n_k \leq n$. Then we can notice that the TT format (11) contains at most dnr^2 elements. This is much smaller than the number of entries in the original tensor which grows exponentially in d . Using Kronecker products, one can rewrite (11) as follows,

$$\tilde{\boldsymbol{\xi}} = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} \xi_{\alpha_0, \alpha_1}^{(1)} \otimes \xi_{\alpha_1, \alpha_2}^{(2)} \otimes \cdots \otimes \xi_{\alpha_{d-1}, \alpha_d}^{(d)},$$

i.e. we see each TT block as a set of vectors of length n_k .

Of course, one can think of any other scheme of sampling a function, e.g. at random points, but the TT decomposition requires independence of sub-indices i_1, \dots, i_d , and therefore the Cartesian product discretisation. The rationale behind using this, on the first glance excessive, scheme, is the fast convergence of the approximation error ε with the TT ranks. If $\xi(x, y, z)$ is analytic, the TT ranks often depend logarithmically on ε [56, 26, 53]. Combining the TT approximation with collocation on the Chebyshev grid,

which allows to take $n = \mathcal{O}(|\log \varepsilon|)$ for analytic functions, one arrives at $\mathcal{O}(d|\log \varepsilon|^3)$ overall cost of interpolation or integration using the TT format. This can be significantly cheaper than the $\mathcal{O}(\varepsilon^{-2})$ cost of Monte Carlo quadrature or Radial Basis function interpolation. Moreover, TT ranks depend usually very mildly on the particular univariate discretisation scheme, provided that it can resolve the function. We can use any univariate grid in each variable instead of the Chebyshev rule. For example, a uniform grid yields Toeplitz or circulant covariance matrices, which are amenable to fast FFT-based multiplication/diagonalisation.

However, it is difficult to obtain sharp bounds for the TT ranks theoretically. Therefore, we resort to robust numerical algorithms to compute a TT approximation of given data.

4.2 TT-cross approximation

A full tensor can be compressed into a TT format quasi-optimally for the desired tolerance via the truncated singular value decomposition (SVD) [47]. However, the full tensor might even be impossible to store. In this section we recall the practical TT-cross method [48] that computes the representation (11) using *only a few* entries from ξ . It is based on the skeleton decomposition of a matrix [14], which represents an $n \times m$ matrix A of rank r as the *cross* (in Matlab-like notation)

$$A = A(:, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, :) \quad (12)$$

of r columns and rows, where \mathcal{I} and \mathcal{J} are two index sets of cardinality r such that $A(\mathcal{I}, \mathcal{J})$ (the intersection matrix) is invertible. If $r \ll n, m$, the right-hand side requires only $(n + m - r)r \ll nm$ elements of the original matrix.

In order to describe the TT-cross method, we introduce the so-called *unfolding* matrices $\Xi_k = [\xi(i_1, \dots, i_k; i_{k+1}, \dots, i_d)]$, that have the first k indices grouped together to index rows, and the remaining indices grouped to index columns. Let us now consider Ξ_1 and apply the idea of the matrix cross (12). Assume that there exists a set of r_1 index tuples, $\mathcal{I}_{>1} = \{i_2^{\alpha_1}, \dots, i_d^{\alpha_1}\}_{\alpha_1=1}^{r_1}$, such that the $\mathcal{I}_{>1}$ -“columns” of the original tensor $\xi(:, \mathcal{I}_{>1})$ form a “good” basis for all columns of Ξ_1 . The reduction (12) may be formed for r_1 rows at positions $\mathcal{I}_{<2} = \{i_1^{\alpha_1}\}_{\alpha_1=1}^{r_1}$, which are now *optimized* by choosing the $r_1 \times r_1$ submatrix $\xi(\mathcal{I}_{<2}, \mathcal{I}_{>1})$ such that its *volume* (modulus of determinant) is maximal. This can be done by the *maxvol* algorithm [13] in $\mathcal{O}(nr_1^2)$ operations. Now we construct the first TT block $\xi^{(1)}$ as the $n \times r_1$ matrix $\xi(:, \mathcal{I}_{>1})\xi(\mathcal{I}_{<2}, \mathcal{I}_{>1})^{-1}$. In a practical algorithm, the inversion is performed via the QR-decomposition for numerical stability. Next, we reduce the tensor onto $\mathcal{I}_{<2}$ in the first variable, and apply TT-cross inductively to $[\Xi_{>1}(\alpha_1, i_2, \dots, i_d)] = [\xi(i_1^{\alpha_1}, i_2, \dots, i_d)]$.

In the k -th step, assume that we are given the reduction $\Xi_{>k-1}(\alpha_{k-1}, i_k, \dots, i_d)$, a “left” index set $\mathcal{I}_{<k} = \{i_1^{\alpha_{k-1}}, \dots, i_{k-1}^{\alpha_{k-1}}\}_{\alpha_{k-1}=1}^{r_{k-1}}$, and a “right” set $\mathcal{I}_{>k} = \{i_{k+1}^{\alpha_k}, \dots, i_d^{\alpha_k}\}_{\alpha_k=1}^{r_k}$. The $r_{k-1}n \times r_k$ reduced unfolding matrix $[\Xi_{>k-1}(\alpha_{k-1}, i_k; \mathcal{I}_{>k})]$ is again feasible for the *maxvol* algorithm, which produces a set of row positions $\ell_k = \{\alpha_{k-1}^{\alpha_k}, i_k^{\alpha_k}\}_{\alpha_k=1}^{r_k}$. The next left set $\mathcal{I}_{<k+1}$ is constructed from ℓ_k by replacing α_{k-1} with the corresponding indexes $i_1^{\alpha_{k-1}}, \dots, i_{k-1}^{\alpha_{k-1}}$ from $\mathcal{I}_{<k}$. Continuing this process until the last variable, where we just copy $\xi^{(d)} = \Xi_{>d-1}$, we complete the induction.

This process can be also organized in a form of a binary tree, which gives rise to the so-called hierarchical Tucker cross algorithm [1]. In total, we need $\mathcal{O}(dnr^2)$ evaluations of ξ and $\mathcal{O}(dnr^3)$ additional operations in computations of the maximum volume matrices.

Algorithm 1 TT cross algorithm with rank adaptation.

Require: Initial index sets $\mathcal{I}_{>k}$, rank increasing parameter $\rho \geq 0$, stopping tolerance $\delta > 0$ and/or maximum number of iterations iter_{\max} .

Ensure: TT blocks of an approximation (11) to ξ .

```

1: while  $\text{iter} < \text{iter}_{\max}$  and  $\|\tilde{\xi}_{\text{iter}} - \tilde{\xi}_{\text{iter}-1}\| > \delta \|\tilde{\xi}_{\text{iter}}\|$  do
2:   for  $k = 1, 2, \dots, d$  do ▷ Forward iteration
3:     (Optionally) prepare an auxiliary enrichment set  $\mathcal{I}_{>k}^{\text{aux}}$ .
4:     Compute the  $r_{k-1}n \times r_k$  unfolding matrix  $\xi(\mathcal{I}_{<k}, i_k; \mathcal{I}_{>k})$ .
5:     Compute  $\mathcal{I}_{<k+1}$  by the maxvol algorithm and (optionally) truncate.
6:   end for
7:   for  $k = d, d-1, \dots, 1$  do ▷ Backward iteration
8:     (Optionally) prepare an auxiliary enrichment set  $\mathcal{I}_{<k}^{\text{aux}}$ .
9:     Compute the  $r_{k-1} \times nr_k$  unfolding matrix  $\xi(\mathcal{I}_{<k}; i_k, \mathcal{I}_{>k})$ .
10:    Compute  $\mathcal{I}_{>k-1}$  by the maxvol algorithm and (optionally) truncate.
11:   end for
12: end while

```

The TT-cross method requires some starting index sets $\mathcal{I}_{>k}$. Without any prior knowledge, it seems reasonable to initialize $\mathcal{I}_{>k}$ with independent realizations of any easy to sample reference distribution (e.g. uniform or Gaussian). If the target tensor ξ admits an *exact* TT decomposition with TT ranks not greater than r_1, \dots, r_{d-1} , and all unfolding matrices have ranks not smaller than the TT ranks of ξ , the cross iteration outlined above reconstructs ξ *exactly* [48]. However, practical tensors can usually only be *approximated* by a TT decomposition with low ranks. Nevertheless a slight *overestimation* of the ranks can deliver a good approximation, if a tensor was produced from a regular enough function [1, 7].

However, it might be necessary to refine the sets $\mathcal{I}_{<k}, \mathcal{I}_{>k}$ by conducting *several* TT cross iterations, going back and forth over the TT blocks and optimizing the sets by the maxvol algorithm. For example, after computing $\xi^{(d)} = \Xi_{>d-1}$, we “reverse” the algorithm and apply the maxvol method to the *columns* of a $r_{d-1} \times n$ matrix $\xi^{(d)}$. This gives a *refined* set of points $\mathcal{I}_{>d-1} = \{i_d^{\alpha_{d-1}}\}$. The recursion continues from $k = d$ to $k = 1$, optimizing the right sets $\mathcal{I}_{>k}$, while taking the left sets $\mathcal{I}_{<k}$ from the previous (forward) iteration. After several iterations, both $\mathcal{I}_{<k}$ and $\mathcal{I}_{>k}$ can be optimized to the particular target function, even if the starting sets were inaccurate.

This adaptation of points can be combined with the *adaptation of ranks*. If the initial ranks r_1, \dots, r_{d-1} were too large, they can be reduced to quasi-optimal values for the desired accuracy via SVD. However, we can also *increase* the ranks by computing the unfolding matrix $[\xi(\mathcal{I}_{<k}, i_k; i_{k+1}^{\alpha_k}, \dots, i_d^{\alpha_k})]$ on an *enriched* index set: we take $\{i_{k+1}^{\alpha_k}, \dots, i_d^{\alpha_k}\}$ from $\mathcal{I}_{>k}$ for $\alpha_k = 1, \dots, r_k$, and also from an *auxiliary* set $\mathcal{I}_{>k}^{\text{aux}}$ for $\alpha_k = r_k + 1, \dots, r_k + \rho$. This increases the k -th TT rank from r_k to $r_k + \rho$. The auxiliary set can be chosen at random [46] or using a surrogate for the error [8]. The pseudocode of the entire TT cross method is listed in Algorithm 1, where we let $\mathcal{I}_{<1} = \mathcal{I}_{>d} = \emptyset$ for uniformity. Empowered with the enrichment scheme, we are not limited to just truncating ranks from above. Instead, we can start with a low-rank initial guess and increase the ranks until the desired accuracy is met.

4.3 TT representation of general and structured matrices

Let us now consider how the TT format (11) can be generalised to matrices $\mathbf{C} \in \mathbb{R}^{n^d \times n^d}$, such as the \mathbf{C}_{ss} matrix from (4). Using sub-indices i_1, \dots, i_d , we can think of a matrix as a $2d$ -dimensional tensor with elements $\mathbf{C}(i_1, \dots, i_d; j_1, \dots, j_d)$. However, most matrices in our applications have full ranks, and a straightforward $2d$ -dimensional TT decomposition would be inefficient. Instead, we consider a permuted, or *matrix* TT decomposition [47]:

$$\mathbf{C}(i_1, \dots, i_d; j_1, \dots, j_d) = \sum_{\beta_0, \dots, \beta_d=1}^{R_0, \dots, R_d} C_{\beta_0, \beta_1}^{(1)}(i_1, j_1) C_{\beta_1, \beta_2}^{(2)}(i_2, j_2) \cdots C_{\beta_{d-1}, \beta_d}^{(d)}(i_d, j_d), \quad (13)$$

or in the Kronecker form,

$$\mathbf{C} = \sum_{\beta_0, \dots, \beta_d=1}^{R_0, \dots, R_d} C_{\beta_0, \beta_1}^{(1)} \otimes C_{\beta_1, \beta_2}^{(2)} \otimes \cdots \otimes C_{\beta_{d-1}, \beta_d}^{(d)}. \quad (14)$$

The identity matrix can be trivially represented in matrix TT format $I_{n^d} = I_n \otimes \cdots \otimes I_d$ with $R_0 = \cdots = R_d = 1$. Furthermore, we can quickly assemble block Toeplitz and circulant matrices if their first column/row is given in the TT format [24]. Let us introduce the operation $\mathcal{T} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{n \times n}$ which assembles a Toeplitz matrix from a vector of its first column and row stacked together, and the operation $\mathcal{C} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ which assembles a circulant matrix from its first column. Assume that a vector \mathbf{c} of size $(2n)^d$ or a vector $\check{\mathbf{c}}$ of size n^d are given in the TT format (11),

$$\mathbf{c} = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} c_{\alpha_0, \alpha_1}^{(1)} \otimes \cdots \otimes c_{\alpha_{d-1}, \alpha_d}^{(d)}, \quad \check{\mathbf{c}} = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} \check{c}_{\alpha_0, \alpha_1}^{(1)} \otimes \cdots \otimes \check{c}_{\alpha_{d-1}, \alpha_d}^{(d)} \quad (15)$$

Then the block Toeplitz or circulant matrix, respectively

$$\mathbf{C} = \left(\bigotimes_{k=1}^d \mathcal{T} \right) \mathbf{c}, \quad \check{\mathbf{C}} = \left(\bigotimes_{k=1}^d \mathcal{C} \right) \check{\mathbf{c}},$$

can be written in the matrix TT formats (13) with the same TT ranks,

$$\mathbf{C} = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} (\mathcal{T} c_{\alpha_0, \alpha_1}^{(1)}) \otimes \cdots \otimes (\mathcal{T} c_{\alpha_{d-1}, \alpha_d}^{(d)}), \quad \check{\mathbf{C}} = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} (\mathcal{C} \check{c}_{\alpha_0, \alpha_1}^{(1)}) \otimes \cdots \otimes (\mathcal{C} \check{c}_{\alpha_{d-1}, \alpha_d}^{(d)}).$$

Similarly we can apply the multivariate Fourier transform without changing TT ranks:

$$\left(\bigotimes_{k=1}^d \mathcal{F} \right) \mathbf{c} = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} (\mathcal{F} c_{\alpha_0, \alpha_1}^{(1)}) \otimes \cdots \otimes (\mathcal{F} c_{\alpha_{d-1}, \alpha_d}^{(d)}), \quad (16)$$

where $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the univariate FFT. This reduces the complexity of FFT from $\mathcal{O}(N \log N) = \mathcal{O}(dn^d \log n)$ to $\mathcal{O}(dr^2n \log n)$.

In general, the TT format allows to represent the product of any matrix given in (13) and a compatible vector given in (11) in another TT format [47] with multiplied ranks,

$$\mathbf{C}\boldsymbol{\xi} = \sum_{\gamma_0, \dots, \gamma_d=1}^{(r_0 R_0), \dots, (r_d R_d)} \left(C_{\beta_0, \beta_1}^{(1)} \xi_{\alpha_0, \alpha_1}^{(1)} \right)_{\gamma_0, \gamma_1} \otimes \cdots \otimes \left(C_{\beta_{d-1}, \beta_d}^{(d)} \xi_{\alpha_{d-1}, \alpha_d}^{(d)} \right)_{\gamma_{d-1}, \gamma_d}, \quad (17)$$

where $\gamma_k = \alpha_k + (\beta_k - 1)r_k$, $k = 0, \dots, d$.

4.4 Kriging operations in TT format

To rewrite the Kriging estimation (8) in low rank format, we first find a TT approximation (15) of \mathbf{c} by using the TT-cross algorithm introduced in Section 4.2. With the rest of the operations we can proceed in two ways.

4.4.1 Small number of scattered samples

If we assume N to be small, the Task-1 of computing Kriging weights, $\mathbf{C}_{yy}\boldsymbol{\xi} = \mathbf{y}$, can be computed directly at low cost. Now we inject the scattered values into a TT tensor of desired size as introduced in (6). Suppose $\ell_j \in \mathbb{N}^d$ is the position of the j th sample, $j = 1, \dots, N$, we can define

$$\mathbf{H}_j = \bigotimes_{k=1}^d \mathbf{e}_j^{(k)}, \quad \text{where} \quad \mathbf{e}_j^{(k)}(i_k) = \begin{cases} 1, & i_k = \ell_j(k) \\ 0, & \text{otherwise,} \end{cases}$$

i.e. the injection operation (6) per sample. Now the injected tensor is written in the CP format as

$$\bar{\boldsymbol{\xi}} = \sum_{j=1}^N \xi_j \mathbf{H}_j, \quad (18)$$

which can be converted to TT format directly by the formula in [16, pp. 380] or using the Alternating Least Squares (ALS) [19] approximation.

Similarly, we can use the direct truncation or the ALS method for summing columns of \mathbf{C}_{sy} with the weights $\boldsymbol{\xi}_i$ in (4), as well as the summation of different vectors $(\mathbf{C}_{sy}\boldsymbol{\xi}_i)^{\circ 2}$.

Embedding operation (7) is simpler and more efficient: we just need to pad every TT block with zeros. Assuming we are given a vector $\boldsymbol{\xi}$ in the form (11), we construct the following new TT blocks of a vector $\tilde{\boldsymbol{\xi}}$:

$$\tilde{\xi}_{\alpha_{k-1}, \alpha_k}^{(k)}(i_k) = \begin{cases} \xi_{\alpha_{k-1}, \alpha_k}^{(k)}(i_k), & i_k = 1, \dots, \bar{n}_k, \\ 0, & i_k = \bar{n}_k + 1, \dots, n_k, \end{cases} \quad k = 1, \dots, d. \quad (19)$$

Similarly, Extraction operation is performed by truncating the range of i_k in each TT block from n_k back to \bar{n}_k . Most importantly, embedding and extraction can be performed very efficiently without changing the TT ranks, similarly to FFT (16).

Finally, we need to compute the Hadamard products of TT tensors, e.g. $\mathcal{F}^{[d]}(\check{\mathbf{c}}) \circ \mathcal{F}^{[d]}(\check{\boldsymbol{\xi}})$ in (8). The Hadamard product can be constructed exactly via (17) by noticing that

$$\mathbf{s} := \mathbf{c} \circ \boldsymbol{\xi} = \mathbf{C}\boldsymbol{\xi}, \quad \text{for} \quad \mathbf{C} = \text{diag}(\mathbf{c}),$$

or approximately by applying the TT-Cross algorithm to a tensor given elementwise by the formula $\mathbf{s}(i_1, \dots, i_d) = \mathbf{c}(i_1, \dots, i_d)\boldsymbol{\xi}(i_1, \dots, i_d)$. The direct multiplication requires $\mathcal{O}(dnR^2r^2)$ operations, and the truncation afterwards has an even higher cost $\mathcal{O}(dnR^3r^3)$. In contrast, the TT-Cross approach needs computing $\mathcal{O}(dnr^2)$ samples of the target tensor \mathbf{s} , which means taking samples of the TT decompositions for \mathbf{c} and $\boldsymbol{\xi}$ and multiplying them. Sampling another TT tensor requires in total $\mathcal{O}(dnR^2r)$ operations, which, assuming that the ranks are comparable, $R \sim r$, results in a total of $\mathcal{O}(dnr^3)$ operations in the TT-Cross computation of Hadamard products, which is thus preferred in this paper.

For geostatistical optimal design (Task-4) we need to compute the trace of $\mathbf{C}_{ss|y}$. Since in the Task-3 we obtain already the diagonal of $\mathbf{C}_{ss|y}$ in the TT format, the trace can be evaluated swiftly by computing a dot product with the all-ones tensor.

4.4.2 Large number of structured samples

When N is large, the summation (18) can be a difficult operation in the TT format, potentially leading also to the TT ranks being in the order of N . However, a large number of samples usually means that these samples are distributed fairly uniformly in the domain of interest. In this case, we switch to the TT computations even before Task-1 in equation (2). First, we interpolate the given samples onto a uniform Cartesian grid with the mesh interval being in the order of the average distance between the original samples. In the remaining operations, we assume that \mathbf{y} is structured in this way, i.e. it can be seen as a tensor $\mathbf{y}(i_1, \dots, i_d)$, $i_k = 1, \dots, \bar{m}_k$, $k = 1, \dots, d$. Thus, we can approximate \mathbf{y} in the TT format.

The solution for weights (2) becomes a rather difficult operation for a large N . However, given the TT decompositions for \mathbf{y} and \mathbf{C}_{yy} , the linear system can be solved more efficiently by employing ALS and similar tensor algorithms [19, 8]. Similarly, we can compute $\mathbf{C}_{yy}^{-1}\mathbf{C}_{ys}$ for (4) by treating \mathbf{C}_{ys} as the right hand side, and expanding \mathbf{C}_{yy} accordingly.

If we interpolate \mathbf{y} onto a periodic uniform Cartesian grid, the matrix \mathbf{C}_{yy} becomes circulant, similarly to $\tilde{\mathbf{C}}$. In this case we can approximate only its first column in the TT format, perform the Fourier transform to obtain the eigenvalues, and apply again the TT-Cross method to approximate the pointwise division $\mathcal{F}^{[d]}(\mathbf{y})(i_1, \dots, i_d) / \mathcal{F}^{[d]}(\mathbf{c})(i_1, \dots, i_d)$.

5 Numerical tests

We used the Matlab package *TT-Toolbox* (<https://github.com/oseledets/TT-Toolbox>) for Tensor Train algorithms. The codes used for numerical experiments are available at <https://github.com/dolgov/TT-FFT-COV>. All computations are done on a MacBook Pro produced in 2013, equipped with 16GB RAM and an 2.7 GHz Intel Core i7 CPU.

We consider three test cases: 1) a 2-dimensional problem with $N = \prod_{i=1}^2 n_i = 600^2$ (it is easy to visualize); 2) a 3-dimensional problem with $N = 10^{15}$ and 3) 10-dimensional problem with $N = \prod_{i=1}^{10} n_i = 100^{10}$. One of these parameters could be, for example, time. The daily soil moisture data set, used below, is taken from [20, 37, 38], where only one replicate, sampled at N locations, is used.

5.1 Kriging of daily moisture data

Numerical models play important role in climate studies. These numerical models are complicated and high-dimensional, including such variables as pressure, temperature, speed, and direction of the wind, level of precipitation, humidity, and moisture. Many parameters are uncertain or even unknown. Accurate modeling of soil moisture finds applications in the agriculture, weather prediction, early warnings of flood and in some others. Since the underlined geographical areas are usually large and high spatial resolutions are required, the involved data sets are huge. This could make the computational process in dense matrix format unfeasible or very expensive. By involving efficient low-rank tensor calculus, we can increase the spatial and time resolution and consider more parameters. It is clear that utilization of the rank k tensor approximation introduces an additional numerical error in quantities of interest (QoIs). By increasing tensor ranks we reduce this approximation error.

We consider high-resolution soil moisture data from January 1, 2014, measured in the topsoil layer of the Mississippi River basin, U.S.A (Fig. 1).

Figure 2 shows an example of daily moisture data. On the left picture we used 2000

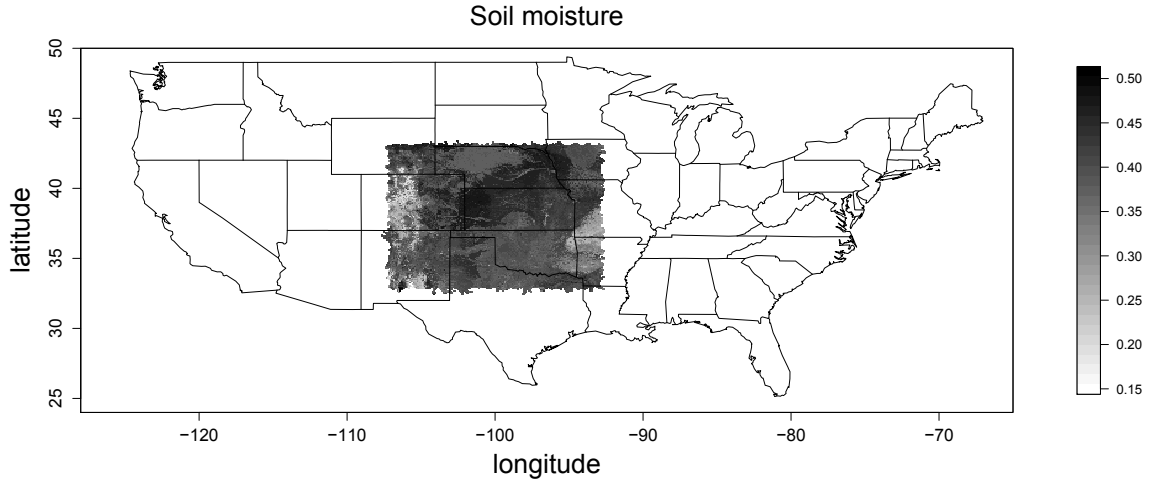


Figure 1: The area where the daily soil moisture data were measured, Mississippi River basin, U.S.A.

points $(x, y, v)_{i=1}^N$, $N = 2000$ for interpolation, and on the right 4000 points. The third picture shows two set of locations: one with 2000 points, marked with the blue symbol + and with 4000 points, marked with red dot.

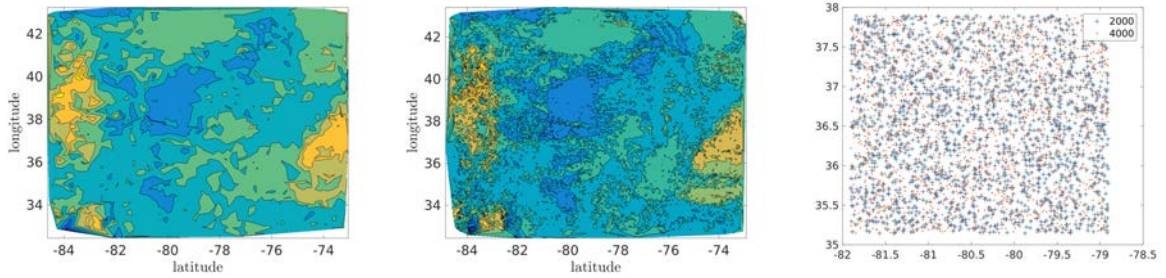


Figure 2: Daily moisture data. Interpolated from (left) 2000 and (center) 4000 measurement points. (right) Two sets of sampling points, 2000 and 4000.

The spatial resolution is 0.0083 degrees, and the distance of one-degree difference in this region is approximately 87.5 km. The grid consists of $1830 \times 1329 = 2.432.070$ locations with 2.000.000 observations and 432.070 missing values. Therefore, the available spatial data are not on a regular grid.

The tensor product Kriging is performed as described in Sec. 4.4.2. First, we interpolate the given measurements (Fig. 3, left) onto a (coarse) Cartesian grid with the mesh interval being approximately equal to the average distance between the measurements. Specifically, we ended up with a 65×65 grid (Fig. 3, center). Then the tensor of values on this coarse grid is approximated into a TT decomposition. Finally, the Kriging estimate (2)–(3) on a fine grid with 257×257 points (Fig. 3, right) is computed in the TT format using FFT and TT-Cross algorithms.

5.2 High-dimensional field generation: computational benchmark

To generate the following 2D, 3D and 10D random fields we used the Matlab script `test_generate_y_tt.m` in <https://github.com/dolgov/TT-FFT-COV>.

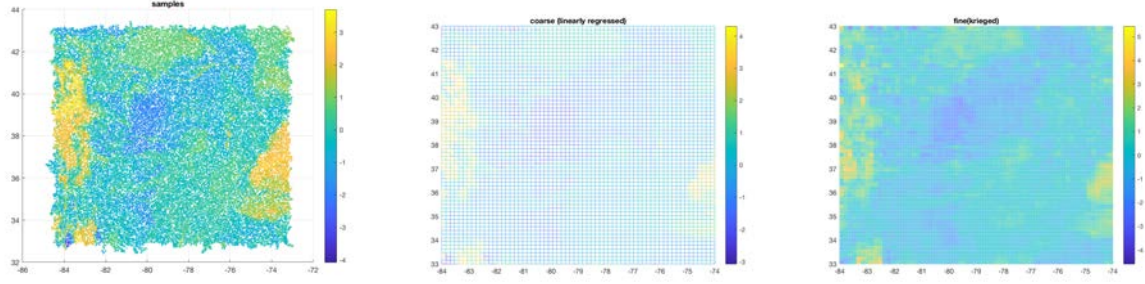


Figure 3: (left) 64000 measurements of the moisture; (center) regression on a coarse 65×65 Cartesian mesh; (right) TT-Kriging approximation on a fine mesh.

2D example. In this example we generated a high-resolution 2-dimensional Matérn random field in $[0, 2000]^2$. One realization is presented in Fig. 4. The smoothness of the Matérn field is $\nu = 0.4$, covariance lengths in x and y directions (1, 1) and the variance 10. This realization is computed by the following formula in the TT format

$$\mathbf{u}' = \mathbf{C}^{1/2} \boldsymbol{\xi} = \sqrt{\frac{1}{n}} \mathbf{F}^\top \boldsymbol{\Lambda}^{1/2} \boldsymbol{\xi} = \sqrt{\frac{1}{n}} \mathcal{F}^{-1}(\boldsymbol{\lambda}^{1/2} \circ \boldsymbol{\xi}), \quad (20)$$

where the inverse Fourier \mathcal{F}^{-1} , the square root of eigenvalues $\boldsymbol{\lambda}^{1/2}$, and tensor product $\boldsymbol{\xi}$ of two Gaussian random vectors are approximated in the TT format. Particularly, $\boldsymbol{\xi} = \boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2$ is a tensor product of two Gaussian vectors. The size of the first column $\tilde{\mathbf{c}}$ of $\tilde{\mathbf{C}}$ is 3200×3600 and the computing time was 1 sec. With TT procedures one can create very fine resolved random fields in large domains. For instance, generation of a random field in the domain $[0, 1.000.000]^2$ with $1.600.000 \times 1.800.000$ locations takes less than 1 minute.

3D example. This example is very similar to the previous 2D example. The difference is only that the domain is $[0, 100.000]^3$ and the size of the first column of \mathbf{C} is $160.000 \times 180.000 \times 160.000 = 4.608 \cdot 10^{15}$. The computing time was 3 minutes.

10D example. In this example, we generated a 10-dimensional Matérn random field. One of the dimensions could be time, for example. Table 1 contains all model parameters and the number of unknowns in (hypothetical) full tensor and in the TT decomposition of the final field $\hat{\mathbf{s}}$. In this example we computed TT approximation of the first column of the multilevel circulant covariance matrix (cf. [24, 25]). Then we diagonalized this circulant matrix via FFT and computed square root of diagonal elements. After that we generated a random field by multiplying the square root with a random vector of the following structure $\boldsymbol{\xi} := \bigotimes_{\nu=1}^{10} \boldsymbol{\xi}_\nu$, where $\boldsymbol{\xi}_\nu$ is a normal vector. We note that we never store the whole vector $\boldsymbol{\xi}$ explicitly, but only its tensor components $\boldsymbol{\xi}_\nu$. Also, note that $\boldsymbol{\xi}$ is not Gaussian.

The TT approximation tolerance is set to 10^{-4} . In the 10-dimensional case above the maximal rank was 143, and the total computing time 118 sec. In the similar 8-dimensional case the maximal rank was 138, and the total computing time 96 sec. Of course, one should observe tensor ranks not only of $\hat{\mathbf{s}}$, but of other steps such as the TT approximation of the measurement vector and of the first column of the covariance matrix. These TT ranks were smaller than the TT ranks of the final solution though.

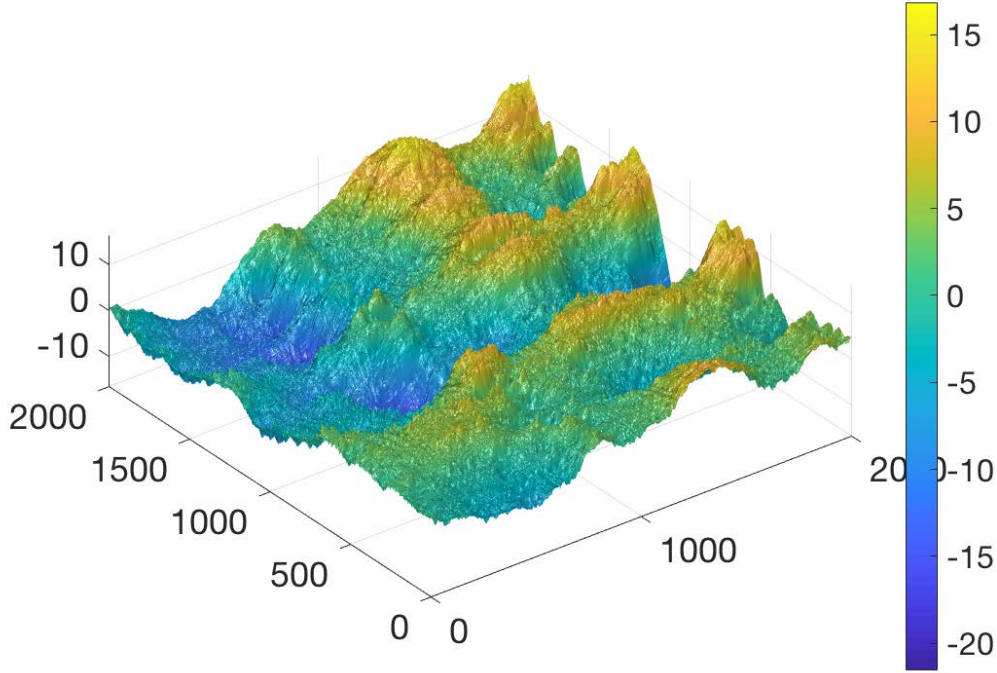


Figure 4: High-resolution realization of 2D Matérn random field, computed with TT tensor format in $[0, 2000]^2$.

Table 1: Parameters of the 10-dimensional problem.

parameter	value
variance of model	10
vector of correlation length in x_1, \dots, x_{10} -direction	[1, 5, 10, 15, 20, 25, 30, 35, 40, 45]
length of domain in x_1, \dots, x_{10} -direction	[10, 50, 100, 150, 200, 250, 300, 350, 400, 450]
number of elements in x_1, \dots, x_{10} -direction	[100, 100, 100, 100, 100, 100, 100, 100, 100, 100]
number of elements in original tensor	$100^{10} = 10^{20}$
number of elements in TT tensor	10^7

6 Discussion and Conclusions

In this paper, we proposed an FFT-based Kriging that utilizes a low-rank Tensor Train (TT) approximation of the covariance matrix. We apply the TT-Cross algorithm to generate a low-rank decomposition avoiding full tensors which could be well beyond the memory capacity of a desktop PC.

The low-rank format reduces the storage of the embedded circulant covariance matrix from exponential to linear in the number of variables. The circulant matrix can be diagonalized by FFT. Furthermore, due to the linearity of the Fourier transform, the TT format allows to implement the d -dimensional FFT at the cost of $\mathcal{O}(dr^2)$ one-dimensional FFT operations.

We then use the same technique to generate large Matérn random fields since the diagonalized covariance matrix gives eigen pairs for the spectral expansion of the underlying random field. We show in numerical examples that this method can generate very large

random fields with a commonly affordable computational resource.

We demonstrated how to utilize the TT tensor format to speed up such geostatistical tasks as the generation of large random fields, computing kriging coefficients, kriging estimates, conditional covariance, and geostatistical optimal design. We used the fact that after discretization on a tensor grid the obtained matrix could be extended to a circulant one. Then, much expensive linear algebra operation could be done via d -dimensional FFT. From the definition, one can see that FFT has tensor rank 1. After approximating the first column of the circulant matrix in the TT format (we assumed that such approximation exists) we were able to apply efficient TT tensor arithmetics and speedup expensive calculations even more. Utilizing TT format in FFT calculus allowed us to decrease computational cost and storage from $\mathcal{O}(\bar{N} \log \bar{N})$ to $\mathcal{O}(dr^3\bar{n})$, where $r \geq 1$ is the tensor rank, d the dimensionality of the problem and \bar{n} is the number of points along the single longest edge of the estimation grid.

The presented numerical techniques have memory requirements as low as $\mathcal{O}(d\bar{n}r^2)$. Thus, we achieved log-complexity in the total number of lattice points. The resulting methods allow much better spatial resolution and significantly reduce the computing time.

The fundamental assumptions are: the covariance matrix is separable or has a TT-rank $r \ll n$, the interpolation grid is a rectangular tensor grid, and the measurements also lie in the tensor grid. The random vector used to generate the random field is a Kronecker product of smaller random vectors.

Acknowledgments

The research reported in this publication was supported by funding from the Alexander von Humboldt Foundation. We also would like to thank Wolfgang Nowak for sharing his Matlab code.

REFERENCES

- [1] J. Ballani and L. Grasedyck. Hierarchical tensor approximation of output quantities of parameter-dependent PDEs. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):852–872, 2015.
- [2] S. Barnett. *Matrices Methods and Applications*. Oxford Applied Mathematics and Computing Science Series. Clarendon Press, Oxford, 1990.
- [3] P. Bogaert. Comparison of kriging techniques in a space-time context. *Mathematical Geology*, 28(1):73–86, 1996.
- [4] R. H. Chan and M. K. Ng. Conjugate gradient methods for Toeplitz systems. *SIAM Review*, 38(3):427–482, 1996.
- [5] O. A. Cirpka and W. Nowak. First-order variance of travel time in non-stationary formations. *Water Resour. Res.*, 40(3):W03507, 2004.
- [6] 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.

- [7] S. Dolgov and R. Scheichl. A hybrid Alternating Least Squares – TT Cross algorithm for parametric PDEs. arXiv preprint 1707.04562, 2017.
- [8] S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM J. Sci. Comput.*, 36(5):A2248–A2271, 2014.
- [9] P. A. Finke, D. J. Brus, M. F. P. Bierkens, T. Hoogland, M. Knotters, and F. De Vries. Mapping groundwater dynamics using multiple sources of exhaustive high resolution data. *Geoderma*, 123(1):23–39, 2004.
- [10] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proc. ICASSP*, volume 3, pages 1381–1384, IEEE, Seattle, WA, 1998. <http://www.fftw.org>.
- [11] J. Fritz, W. Nowak, and I. Neuweiler. Application of FFT-based algorithms for large-scale universal Kriging problems. *Math. Geosci.*, 41(5):509–533, 2009.
- [12] M. G. Genton. Separable approximations of space-time covariance matrices. *Environmetrics*, 18:681–695, 2007.
- [13] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. How to find a good submatrix. In V. Olshevsky and E. Tyrtyshnikov, editors, *Matrix Methods: Theory, Algorithms, Applications*, pages 247–256. World Scientific, Hackensack, NY, 2010.
- [14] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.
- [15] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitt.*, 36(1):53–78, 2013.
- [16] W. Hackbusch. *Elliptic differential equations*, volume 18 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1992. Theory and numerical treatment, Translated from the author’s revision of the 1986 German original by Regine Fadiman and Patrick D. F. Ion.
- [17] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer Series in Computational Mathematics. Springer Verlag, 2012.
- [18] M. R. Haylock, N. Hofstra A. M. G. Klein 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:D20119, 2008.
- [19] S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM J. Sci. Comput.*, 34(2):A683–A713, 2012.
- [20] H. Huang and Ying S. Hierarchical low rank approximation of likelihoods for large spatial datasets. *Journal of Computational and Graphical Statistics*, 27(1):110–118, 2018.

- [21] S. De Iaco, S. Maggio, M. Palma, and D. Posa. Toward an automatic procedure for modeling multivariate space-time data. *Computers & Geosciences*, 41:1–11, 2011.
- [22] A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, New York, 1978.
- [23] T. Kailath and A. H. Sayed. Displacement structure: Theory and applications. *SIAM Review*, 37(3):297–386, 1995.
- [24] V. Kazeev, B. Khoromskij, and E. Tyrtysnikov. Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity. *SIAM J. Sci. Comput.*, 35(3):A1511–A1536, 2013.
- [25] V. Khoromskaia and B. N. Khoromskij. Block circulant and toeplitz structures in the linearized hartree–fock equation on finite lattices: Tensor approach. *Computational Methods in Applied Mathematics*, 17(3):431–455, 02 2017.
- [26] B. N. Khoromskij. Structured rank- (r_1, \dots, r_d) decomposition of function-related operators in \mathbb{R}^d . *Comput. Methods Appl. Math*, 6(2):194–220, 2006.
- [27] B. N. Khoromskij. Tensor-structured numerical methods in scientific computing: Survey on recent advances. *Chemom. Intell. Lab. Syst.*, 110(1):1–19, 2012.
- [28] B. N. Khoromskij. *Tensor numerical methods in scientific computing*. Walter de Gruyter GmbH & Co KG, 2018.
- [29] B. N. Khoromskij and A. Litvinenko. Data sparse computation of the karhunen-loève expansion. *AIP Conference Proceedings*, 1048(1):311–314, 2008.
- [30] P. K. Kitanidis. Analytical expressions of conditional mean, covariance, and sample functions in geostatistics. *Stoch. Hydrol. Hydraul.*, 12:279–294, 1996.
- [31] P. K. Kitanidis. *Introduction to Geostatistics*. Cambridge University Press, Cambridge, 1997.
- [32] 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(5):828 – 845, 2008.
- [33] B. Kozintsev. *Computations with Gaussian random fields*. PhD thesis, Institute for Systems Research, University of Maryland, 1999.
- [34] P. Kyriakidis and A. Journel. Geostatistical space–time models: A review. *Mathematical Geology*, 31:651–684, 1999. doi:10.1023/A:1007528426688.
- [35] A. Litvinenko. HLIBCov: Parallel Hierarchical Matrix Approximation of Large Covariance Matrices and Likelihoods with Applications in Parameter Identification. *arXiv 1709.08625*, Sep 2017.
- [36] A. Litvinenko, D. Keyes, V. Khoromskaia, B. N. Khoromskij, and H. G. Matthies. Tucker tensor analysis of matérn functions in spatial statistics. *Computational Methods in Applied Mathematics*, 19(1):101–122, 2019.

- [37] A. Litvinenko, Y. Sun, M. G. Genton, and D. E. Keyes. Likelihood approximation with hierarchical matrices for large spatial datasets. *Computational Statistics & Data Analysis*, 137:115 – 132, 2019.
- [38] A. Litvinenko, Y. Sung, H. Huang, M. G. Genton, and D. E. Keyes. Github repository: daily moisture data, <https://github.com/litvinen/HLIBCov.git>, 2017.
- [39] B. Matérn. *Spatial variation*. Springer, Berlin, Germany, 1986.
- [40] G. Matheron. *The Theory of Regionalized Variables and Its Applications*. Ecole de Mines, Fontainebleau, France, 1971.
- [41] W. G. Müller. *Collecting spatial data. Optimum design of experiments for random fields*. Springer, Berlin, Germany, 3 edition, 2007.
- [42] G. N. Newsam and C. R. Dietrich. Bounds on the size of nonnegative definite circulant embeddings of positive definite Toeplitz matrices. *IEEE Transactions on Information Theory*, 40(4):1218–1220, 1994.
- [43] W. Nowak. Measures of parameter uncertainty in geostatistical estimation and geostatistical optimal design. *Math. Geosciences*, 42(2):199–221, 2010.
- [44] 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(4):411–435, May 2013.
- [45] W. Nowak, S. Tenkleve, and O. A. Cirpka. Efficient computation of linearized cross-covariance and auto-covariance matrices of interdependent quantities. *Math. Geol.*, 35(1):53–66, 2003.
- [46] I. V. Oseledets. DMRG approach to fast linear algebra in the TT-format. *Comput. Meth. Appl. Math.*, 11(3):382–393, 2011.
- [47] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.
- [48] I. V. Oseledets and E. E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.*, 432(1):70–88, 2010.
- [49] G. G. S. Pegram. Spatial interpolation and mapping of rainfall (SIMAR) Vol.3: Data merging for rainfall map production. *Water Research Commission Report*, (1153/1/04), 2004.
- [50] L. Pesquer, A. Cortés, and X. Pons. Parallel ordinary kriging interpolation incorporating automatic variogram fitting. *Computers & Geosciences*, 37(4):464–473, 2011.
- [51] P. Reed, B. Minsker, and A. J. Valocchi. Cost-effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation. *Water Resour. Res.*, 36(12):3731–3741, 2000.
- [52] A. K. Saibaba and P. K. Kitanidis. Efficient methods for large-scale linear inversion using a geostatistical approach. *Water Resour. Res.*, 48(W05522), 2012.

- [53] R. Schneider and A. Uschmajew. Approximation rates for the hierarchical tensor format in periodic sobolev spaces. *Journal of Complexity*, 2013.
- [54] 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(3):466 – 479, 2011.
- [55] G. Spöck and J. Pilz. Spatial sampling design and covariance-robust minimax prediction based on convex design ideas. *Stochastic Environmental Research and Risk Assessment*, 24:463–482, 2010.
- [56] E. E. Tyrtysnikov. Tensor approximations of matrices generated by asymptotically smooth functions. *Sbornik: Mathematics*, 194(6):941–954, 2003.
- [57] C. F. van Loan. *Computational Frameworks for the Fast Fourier Transform*. SIAM Publications, Philadelphia, PA, 1992.
- [58] R. S. Varga. Eigenvalues of circulant matrices. *Pacific J. Math.*, 4:151–160, 1954.
- [59] J. Vondřejc, D. Liu, M. Ladecký, and H. G. Matthies. FFT-based homogenisation accelerated by low-rank approximations. *arXiv e-prints*, page arXiv:1902.07455, Feb 2019.
- [60] S. M. Wesson and G. G. S. Pegram. Radar rainfall image repair techniques. *Hydrological and Earth Systems Sciences*, 8(2):8220–8234, 2004.
- [61] D. L. Zimmerman. Computationally exploitable structure of covariance matrices and generalized covariance matrices in spatial models. *J. Stat. Comput. Sim.*, 32(1/2):1–15, 1989.