論文 / 著書情報
Article / Book Information

Title	Multi-level restricted maximum likelihood covariance estimation and kriging for large non-gridded spatial datasets
Authors	Julio Castrillon-Candas, Marc Genton, Rio Yokota
Citation	Spatial Statistics, Vol. 18, , pp. 105--124
Pub. date	2015, 11
DOI	http://dx.doi.org/10.1016/j.spasta.2015.10.006
Creative Commons	See next page.
Note	このファイルは著者（最終）版です。 This file is author (final) version.

License

Multi-Level Restricted Maximum Likelihood Covariance Estimation and Kriging for Large Non-Gridded Spatial Datasets

Julio E. Castrillón-Candás[1], Marc G. Genton[2], and Rio Yokota[3]

August 18, 2015

Abstract

We develop a multi-level restricted Gaussian maximum likelihood method for estimating the covariance function parameters and computing the best unbiased predictor. Our approach produces a new set of multi-level contrasts where the deterministic parameters of the model are filtered out thus enabling the estimation of the covariance parameters to be decoupled from the deterministic component. Moreover, the multi-level covariance matrix of the contrasts exhibit fast decay that is dependent on the smoothness of the covariance function. Due to the fast decay of the multi-level covariance matrix coefficients only a small set is computed with a level dependent criterion. We demonstrate our approach on problems of up to 512,000 observations with a Matérn covariance function and highly irregular placements of the observations. In addition, these problems are numerically unstable and hard to solve with traditional methods.

KEY WORDS: Fast Multipole Method; Hierarchical Basis; High Performance Computing; Sparsification of Covariance Matrices

Short title: Multi-Level Restricted Maximum Likelihood and Kriging

SRI Center for Uncertainty Quantification in Computational Science and Engineering[1]; Computer, Electrical and Mathematical Sciences and Engineering[2], King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia; Tokyo Institute of Technology Global Scientific and Computing Center[3], 2-12-1 i7-2 O-okayama Meguro-ku, 152-8550, Tokyo, Japan. E-mails: uvel@alum.mit.edu, marc.genton@kaust.edu.sa, rioyokota@gsic.titech.ac.jp

1 Introduction

Consider the following model for a Gaussian spatial random field Z:

$$Z(\mathbf{s}) = \mathbf{m}(\mathbf{s})^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon(\mathbf{s}), \qquad \mathbf{s} \in \mathbb{R}^d, \tag{1}$$

where $\mathbf{m} \in \mathbb{R}^p$ is a known function of the spatial location $\mathbf{s}$, $\boldsymbol{\beta} \in \mathbb{R}^p$ is an unknown vector of coefficients, and ε is a stationary mean zero Gaussian random field with parametric covariance function $C(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}) = \mathrm{cov}\{\varepsilon(\mathbf{s}), \varepsilon(\mathbf{s}')\}$ having an unknown vector $\boldsymbol{\theta} \in \mathbb{R}^w$ of parameters. We observe the data vector $\mathbf{Z} = (Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n))^{\mathrm{T}}$ at locations $\mathbb{S} := \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, where $\mathbf{s}_1 \neq \mathbf{s}_2 \neq \mathbf{s}_3 \neq \cdots \neq \mathbf{s}_{n-1} \neq \mathbf{s}_n$, and wish to: 1) estimate the unknown vectors $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$; and 2) predict $Z(\mathbf{s}_0)$, where $\mathbf{s}_0$ is a new spatial location. These two tasks are particularly challenging when the sample size n is large.

To address the estimation part, let $\mathbf{C}(\boldsymbol{\theta}) = \mathrm{cov}(\mathbf{Z}, \mathbf{Z}^{\mathrm{T}}) \in \mathbb{R}^{n \times n}$ be the covariance matrix of $\mathbf{Z}$ and assume it is nonsingular for all $\boldsymbol{\theta} \in \mathbb{R}^w$. Define $\mathbf{M} = \big(\mathbf{m}(\mathbf{s}_1) \ldots \mathbf{m}(\mathbf{s}_n)\big)^{\mathrm{T}} \in \mathbb{R}^{n \times p}$ and assume it is of full rank, p. The model (1) leads to the vectorial formulation

$$\mathbf{Z} = \mathbf{M}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2}$$

where $\boldsymbol{\varepsilon}$ is a Gaussian random vector, $\boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \mathbf{C}(\boldsymbol{\theta}))$. Then the log-likelihood function is

$$\ell(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det\{\mathbf{C}(\boldsymbol{\theta})\} - \frac{1}{2}(\mathbf{Z} - \mathbf{M}\boldsymbol{\beta})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{Z} - \mathbf{M}\boldsymbol{\beta}), \tag{3}$$

which can be profiled by generalized least squares with

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) = \{\mathbf{M}^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{M}\}^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{Z}. \tag{4}$$

A consequence of profiling is that the maximum likelihood estimator (MLE) of $\boldsymbol{\theta}$ then tends to be biased. A solution to this problem is to use restricted maximum likelihood (REML) estimation which consists in calculating the log-likelihood of $n - p$ linearly independent con-

trasts, that is, linear combinations of observations whose joint distribution does not depend on $\boldsymbol{\beta}$, from the set $\mathbf{Y} = \{\mathbf{I}_n - \mathbf{M}(\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\}\mathbf{Z}$. In this paper, we propose a new set of contrasts that lead to significant computational benefits (with good accuracy) when computing the REML estimator of $\boldsymbol{\theta}$ for large sample size n.

To address the prediction part, consider the best unbiased predictor $\hat{Z}(\mathbf{s}_0) = \lambda_0 + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{Z}$ where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^{\mathrm{T}}$. The unbiasedness constraint implies $\lambda_0 = 0$ and $\mathbf{M}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{m}(\mathbf{s}_0)$. The minimization of the mean squared prediction error $\mathrm{E}[\{Z(\mathbf{s}_0) - \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{Z}\}^2]$ under the constraint $\mathbf{M}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{m}(\mathbf{s}_0)$ yields

$$\hat{Z}(\mathbf{s}_0) = \mathbf{m}(\mathbf{s}_0)^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \mathbf{c}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{Z} - \mathbf{M}\hat{\boldsymbol{\beta}}), \tag{5}$$

where $\mathbf{c}(\boldsymbol{\theta}) = \mathrm{cov}\{\mathbf{Z}, Z(\mathbf{s}_0)\} \in \mathbb{R}^n$ and $\hat{\boldsymbol{\beta}}$ is defined in (4). In this paper, we propose a new transformation of the data vector $\mathbf{Z}$ leading to a decoupled multi-level description of the model (1) without any loss of structure. This multi-level representation leads to significant computational benefits when computing the kriging predictor $\hat{Z}(\mathbf{s}_0)$ in (5) for large sample size n.

Previous work has been performed to maximize (3). The classical technique is to compute a Cholesky factorization of $\mathbf{C}$. However, this requires $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ computational steps, thus impractical for large scale problems.

Under special structures of the covariance matrix, i.e., fast decay of the covariance function, a tapering technique can be used to sparsify the covariance matrix and thus increase memory and computational efficiency (Furrer et al. (2006); Kaufman et al. (2008)). These techniques are good when applicable but tend to be restrictive. For a review of various approaches to spatial statistics for large datasets, see Sun et al. (2012).

Recently we have seen the advent of solving the optimization problem (3) from a computational numerical perspective. Anitescu et al. (2012) developed a matrix-free approach for computing the maximum of the log-likelihood (3) based on a stochastic programming refor-

mulation. This method relies on Monte Carlo approximation of the derivative of the score function with respect to the covariance parameters $\boldsymbol{\theta}$ to compute the maximization (3). The authors show promising results for a grid geometry of the placement of the observations. However for a non-grid geometry the cost of computing the preconditioner becomes $\mathcal{O}(n^2)$ and it is not clear how many iterations for convergence are needed as the geometry deviates from a grid. Moreover, due to the slow convergence rate of the Monte Carlo method many samples might be potentially required before a suitable estimate is obtained. The previous work was extended in Stein et al. (2013). Although the results are impressive (1,000,000 + size problems), the approach is restricted to regular grid geometries with partially occluded areas.

Stein et al. (2012) presented a difference filter preconditioning for large covariance matrices not unlike our multi-level method. By constructing a preconditioner based on the difference filter the number of iterations of a Preconditioned Conjugate Gradient (PCG) drops significantly. However, the authors can only construct a preconditioner for irregularly placed observations in 1D and for a regular grid in higher dimension. Moreover, the authors point out that the restrictions on the spectral density of the random field Z are strong.

In Stein et al. (2004) the authors proposed a REML method in combination with an approximation of the likelihood. This approach uses a truncation method to compute an approximation of the likelihood function. It appears to be effective if the truncated terms have small correlations. However, if the covariance function has a slow decay then we expect that this approximation might not be accurate unless a large neighborhood is incorporated. Moreover, this paper does not include an analysis of the error with respect to the truncation.

In Sun and Stein (2015) the authors proposed new unbiased estimating equations based on score equation approximations. The inverse covariance matrix is approximated with a sparse inverse Cholesky decomposition. As in Stein et al. (2004) the approximation should be fast and accurate for locally correlated observations but will suffer from slow decay of the covariance function. Moreover, the results are limited to grid-like geometries.

In the next section we present the basic ideas behind our approach. In Section 3 we show the construction of a multi-level basis from the observations points. In Section 4 we describe how to efficiently construct a multi-level covariance matrix that arises from the new basis. In Section 5 a multi-level estimator is proposed. In Section 6 the multi-level kriging approach is described. In Section 7 hard to solve numerical examples are provided and compared with traditional methods. In Section 8 we give concluding remarks. Proofs are relegated to the Appendix A and a notation summary can be found in Appendix B. We also include computational and mathematical details in the remarks. However, these may be skipped on a first reading except for the more mathematically oriented reader.

2 Multi-Level REML and Kriging Basic Approach

We now present the main ideas of our proposal. Denote by $\mathcal{P}^p(\mathbb{S})$ the span of the columns of the design matrix $\mathbf{M}$. Let $\mathbf{L} \in \mathbb{R}^{p\times n}$ be an orthogonal projection from $\mathbb{R}^n$ to $\mathcal{P}^p(\mathbb{S})$ and $\mathbf{W} \in \mathbb{R}^{(n-p)\times n}$ be an orthogonal projection from $\mathbb{R}^n$ to $\mathcal{P}^p(\mathbb{S})^\perp$, the orthogonal complement of $\mathcal{P}^p(\mathbb{S})$. By applying the operator $\mathbf{W}$ to (2) we obtain $\mathbf{Z}_W = \mathbf{W}\mathbf{Z} = \mathbf{W}(\mathbf{M}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) = \mathbf{W}\boldsymbol{\varepsilon}$. Our first observation is that the trend contribution $\mathbf{M}\boldsymbol{\beta}$ is filtered out from the data $\mathbf{Z}$. We can now formulate the estimation of the covariance parameters $\boldsymbol{\theta}$ without the trend. The new log-likelihood function becomes

$$\ell_W(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det\{\mathbf{C}_W(\boldsymbol{\theta})\} - \frac{1}{2}\mathbf{Z}_W^{\mathrm{T}}\mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{Z}_W, \tag{6}$$

where $\mathbf{C}_W(\boldsymbol{\theta}) = \mathbf{W}\mathbf{C}(\boldsymbol{\theta})\mathbf{W}^{\mathrm{T}}$ and $\mathbf{Z}_W \sim \mathcal{N}_{n-p}(\mathbf{0}, \mathbf{W}\mathbf{C}(\boldsymbol{\theta})\mathbf{W}^{\mathrm{T}})$. As shown in Section 5 , to estimate the coefficients $\boldsymbol{\theta}$ it is not necessary to compute $\ell_W(\boldsymbol{\theta})$ but a multi-resolution version.

A consequence of the filtering is that we obtain an unbiased estimator. Moreover, a further consequence is that if $\mathbf{v} \neq 0$ then

$$0 < \min_{\mathbf{v}\in\mathbb{R}^n} \mathbf{v}^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\mathbf{v} \leq \min_{\mathbf{v}\in\mathbb{R}^n\setminus\mathcal{P}^p(\mathbb{S})} \mathbf{v}^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\mathbf{v} \leq \max_{\mathbf{v}\in\mathbb{R}^n\setminus\mathcal{P}^p(\mathbb{S})} \mathbf{v}^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\mathbf{v} \leq \max_{\mathbf{v}\in\mathbb{R}^n} \mathbf{v}^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\mathbf{v}. \tag{7}$$

This implies that the condition number of $\mathbf{C}_W(\boldsymbol{\theta})$ is less than or equal to the condition number of $\mathbf{C}(\boldsymbol{\theta})$. Thus computing the inverse of $\mathbf{C}_W(\boldsymbol{\theta})$ will be in general more stable than for $\mathbf{C}(\boldsymbol{\theta})$. In practice, computing the inverse of $\mathbf{C}_W(\boldsymbol{\theta})$ will be much more stable than $\mathbf{C}(\boldsymbol{\theta})$ (See the results in Tables 4 and 5). Finally, the uncertainties in the parameter estimates obtained from (6) can be quantified using the Godambe information matrix as described in Sect. 2 and Appendix B of Stein et al. (2004).

As shown in Section 4, for covariance functions that are differentiable up to a degree $\tilde{f}+1$ (except at the origin), such as the Matérn, our approach leads to covariance matrices $\mathbf{C}_W$ where most of the coefficients are small and thus can be safely eliminated. We construct a level dependent criterion approach to determine which entries are computed and the rest are set to zero. With this approach we can now construct a sparse covariance matrix $\tilde{\mathbf{C}}_W$ that is close to $\mathbf{C}_W$ in a matrix norm sense even if the observations are highly correlated with distance.

The sparsity of $\tilde{\mathbf{C}}_W$ will depend on the following: i) a positive integer τ, which is a multi-level distance criterion; ii) a positive integer $\tilde{f}$, which is the degree of the multi-level basis and associated accuracy parameters $\tilde{p}$; and iii) the smoothness of the covariance function. The accuracy of $\tilde{\mathbf{C}}_W$ will depend monotonically on these parameters, i.e., as we increase τ and $\tilde{f}$ (and respectively $\tilde{p}$) the matrix $\tilde{\mathbf{C}}_W$ will be closer to $\mathbf{C}_W$ in a norm sense. This is explained in detail in Section 4.

The choice of the projectors $\mathbf{L}$ and $\mathbf{W}$ will determine how efficiently each likelihood function (6) evaluation is solved. *Indeed, we desire the transformation to have the following properties:* i) **Stability:** The matrices $\mathbf{L}$ and $\mathbf{W}$ have orthogonal rows and the stacked matrix $[\mathbf{L};\mathbf{W}]$ is orthonormal; ii) **Fast computation:** The computational cost of applying the matrix $[\mathbf{L};\mathbf{W}]$ to a vector is $\mathcal{O}(n(\log n)^{\xi})$ for some small integer ξ; iii) **Fast log determinant computation:** The computational cost of computing $\log\det\{\tilde{\mathbf{C}}_W(\boldsymbol{\theta})\}$ to be bounded by $\mathcal{O}(n^{3/2})$ in 2D and $\mathcal{O}(n^2)$ in 3D. We also want to restrict the memory storage to $\mathcal{O}(n(\log n)^{\xi})$; iv) **Fast inversion:** The

computational cost of computing $\mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{Z}_W$ to a desired accuracy ε is better than $\mathcal{O}(n^2)$. Memory storage is also desirable to be restricted to $\mathcal{O}(n(\log n)^{\xi})$; v) **Accuracy:** Determinant computation and inversion are also required to be accurate. We achieve the properties i) - v) in this paper.

In Section 3 we describe how to construct multi-level matrices $\mathbf{L}$ and $\mathbf{W}$ that satisfy properties i) and ii) for most practical observation location placements (random for example). We apply $\mathbf{L}$ and $\mathbf{W}$ to construct the sparse multi-level covariance matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$. The determinant of the multi-level sparse covariance matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ and the term $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})^{-1}\mathbf{Z}_W$ are computed by exploiting an accurate sparse Cholesky representation of $\mathbf{C}_W$ (properties iii) and v)). The term $\mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{Z}_W$ can also be computed by applying a Preconditioned Conjugate Gradient (PCG) to a desired accuracy (properties iv) and v)). In Section 6 the multi-level kriging method is described. In Section 7 we demonstrate the efficiency of our method for numerous covariances and irregularly placed observations. We are able to solve the fast inversion for up to 512,000 observations to a relative accuracy of 10^{-3} with respect to the *unpreconditioned* system. It is important to note that the achieved accuracy of preconditioned system will not necessarily imply accuracy of unpreconditioned system if the condition number of the preconditioner is high. Furthermore, we test our approach to estimate the covariance parameters of problems of up to 128,000 observations. In addition, the accuracy of the kriging estimates are tabulated for different size problems.

3 Multi-Level Basis

In this section we establish the general structure of the Multi-Level Basis (MB) that is used solve the estimation and prediction problem. We refer the reader to Castrillón-Candás et al. (2013) for a detailed description. The MB can then be used to: (i) form the multi-level REML function (6); (ii) sparsify the covariance matrix $\mathbf{C}_W(\boldsymbol{\theta})$; and (iii) improve the conditioning over the covariance matrix $\mathbf{C}(\boldsymbol{\theta})$. But first, we establish some notation and definitions:

- Let $\alpha := (\alpha_1, \ldots, \alpha_d) \in \mathbb{Z}^d$, $|\alpha| := \alpha_1 + \cdots + \alpha_d$, $\mathbf{x} := [x_1, \ldots, x_d]$ and $D^{\alpha}_{\mathbf{x}} := \frac{\partial^{\alpha_1 + \cdots + \alpha_d}}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}$. For any $h \in \mathbb{N}_0$ (where $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$) let $\mathcal{Q}^d_h$ be the set of monomials $\{x_1^{\alpha_1} \ldots x_d^{\alpha_d} \mid |\alpha| \leq h\}$. Furthermore, let $\mathbf{M}_h$ be the design matrix with respect to all the monomials in $\mathcal{Q}^d_h$. The number of monomials of degree h with dimension d is $\binom{d+h}{h}$.

- We shall restrict the Gaussian spatial random field (1) design matrix $\mathbf{M}$ to $\mathbf{M}_f$, where f is the degree of the model. Thus p will be equal to the number of monomials in $\mathcal{Q}^d_f$, which is $p := \binom{d+f}{f}$.

- Let $\tilde{f} \geq f$ be the degree of the multi-level basis and $\mathbf{M}_{\tilde{f}}$ the associated design matrix. The number of monomials in $\mathcal{Q}^d_{\tilde{f}}$ shall be referred as the accuracy parameter $\tilde{p} := \binom{d+\tilde{f}}{\tilde{f}}$. These parameters are chosen by the user and are used to construct the multi-level basis.

- Let $\mathbf{C}(\boldsymbol{\theta}) := \{\phi(r_{i,j}; \boldsymbol{\theta})\}$ where ϕ is the covariance function, $r_{i,j} := \|\mathbf{s}_i - \mathbf{s}_j\|_2$ and $\mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^d$ for $i, j = 1, \ldots, n$. Alternatively we refer to $\phi(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$ as $\phi(r; \boldsymbol{\theta})$, where $r := \|\mathbf{x} - \mathbf{y}\|_2$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Suppose $\mathcal{P}^p(\mathbb{S})$ is the span of the columns of the design matrix $\mathbf{M}_f$. We now assume that $\phi(r; \boldsymbol{\theta})$ is a positive definite function and $C^{\tilde{f}+1}(\mathbb{R})$ for all $r \in \mathbb{R}$ except at the origin.

- For any index $i, j \in \mathbb{N}_0$, $1 \leq i \leq n$, $1 \leq j \leq n$, let $\mathbf{e}_i[j] = \delta[i - j]$, where $\delta[\cdot]$ is the discrete Kronecker delta function.

Remark 1 *In practice instead of using the set of monomials $\mathcal{Q}^d_f$ we use the set of Chebyshev polynomials of the first kind as these lead to a more stable numerical scheme. However, for simplicity of the presentation we keep it to monomials.*

The first step is to decompose the locations into a series of multi-level cubes of dimension d. Without loss of generality we assume that all the locations are contained in a unit cube B^0_0 at level 0 and index 0. If the number of locations inside B^0_0 is more than p then equally subdivide B^0_0 into 2^d cubes ($B^1_0, \ldots, B^1_{2^d-1}$), where d is the number of dimensions. If the number of

locations is p or less then stop, associate every location with the cube B_0^0 and denote this a leaf cube. Otherwise, for each non-empty cube B_k^q at level $q = 1$ and index k if the number of locations is more than p then subdivide, otherwise associate all the locations to B_k^q and denote this a leaf cube. This process is repeated for all the subdivided cubes at levels $q = 2, \ldots,$ until no subdivisions are possible. The result is a tree structure with $0, \ldots, t$ levels (See Algorithm 1 in Castrillón-Candás et al. (2013) for more details). We denote the leaf cubes as all the non-empty cubes that contain at most p locations, i.e., they will correspond to the leafs of the tree structure.

Remark 2 *For practical cases, t increases proportionally to $\log n$. If the inter location spacing collapses as n^{-q} , where q is independent of n , then $q \log n$ levels are needed, see Section 4 in Beatson and Greengard (1997) for details.*

Suppose that there is a one-to-one mapping between the set of unit vectors $\mathcal{E} := \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$, which we denote as leaf unit vectors, and the set of locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, i.e. $\mathbf{s}_i \longleftrightarrow \mathbf{e}_i$ for all $i = 1, \ldots, n$. It is clear that the space of $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ is $\mathbb{R}^n$. The next step is to replace $\mathcal{E}$ with a new basis of $\mathbb{R}^n$ that is multi-level, orthonormal and gives us the desired properties i) and ii) from Section 1. In Castrillón-Candás et al. (2013) the reader can find full details on such a construction. However, for the sake of clarity for the rest of the paper we associate the multi-level domain decomposition to the multi-level basis:

- For each non empty box B_k^i, for $i = 0, \ldots, t$, associate a series of multi-level basis vectors $\{\boldsymbol{\psi}_{\tilde{k}_1}^{i,k}, \boldsymbol{\psi}_{\tilde{k}_2}^{i,k}, \ldots\}$ that have the following property:

$$\mathbf{g}^T \boldsymbol{\psi}_{\tilde{k}_j}^{i,k} = \sum_{a=1}^{n} \mathbf{g}[a] \boldsymbol{\psi}_{\tilde{k}_j}^{i,k}[a] = 0, \tag{8}$$

  for $j = 0, \ldots$ and for all the vectors $\mathbf{g}$ that are columns of $\mathbf{M}_{\tilde{f}}$. Furthermore, let $\mathbf{W}^{i,k}$ be a matrix $[\boldsymbol{\psi}_{\tilde{k}_1}^{i,k}, \boldsymbol{\psi}_{\tilde{k}_2}^{i,k}, \ldots]$.

- For $i = 0, \ldots, t$ let $\mathbf{W}_i := [\mathbf{W}^{i,0}, \mathbf{W}^{i,1}, \ldots]$

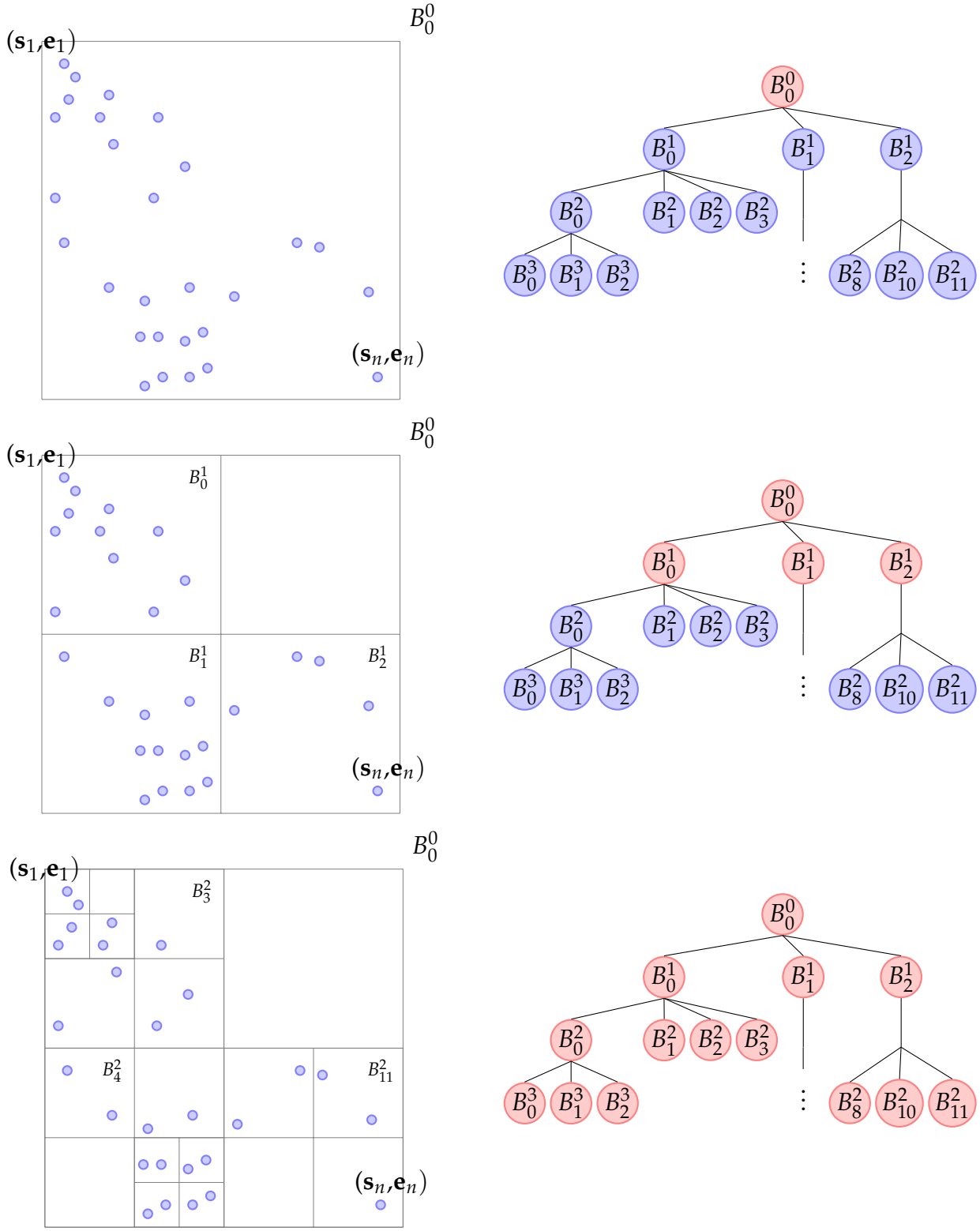


Figure 1: Multi-level domain decomposition of the location of observations for $d = p = 2$. All the observation locations $\mathbf{s}_i$ are assumed to be contained in the unit box B_0^0 (colored red node in tree). If the number of observations are greater than $p = 2$ we subdivide into equal boxes. At the end we obtain a multi-level decomposition of the observation points.

- If $\tilde{p} > p$ we will have an extra $\tilde{p} - p$ vectors corresponding to a level -1 for the initial box B_0^0. Now, associate $\tilde{p} - p$ multi-level vectors $\{\boldsymbol{\psi}_0^{-1,0}, \boldsymbol{\psi}_1^{-1,0}, \ldots, \boldsymbol{\psi}_{\tilde{p}-p}^{-1,0}\}$ that have the following property:

$$\mathbf{g}^{\mathrm{T}} \boldsymbol{\psi}_j^{-1,0} = \sum_{a=1}^{n} \mathbf{g}[a] \boldsymbol{\psi}_j^{-1,0}[a] = 0, \tag{9}$$

for $j = 1, \ldots, \tilde{p} - p$ and for all the vectors $\mathbf{g}$ that are columns of $\mathbf{M}_{\tilde{f}}$. Similarly as above, let $\mathbf{W}_{-1} := [\boldsymbol{\psi}_0^{-1,0}, \boldsymbol{\psi}_1^{-1,0}, \ldots, \boldsymbol{\psi}_{\tilde{p}-p}^{-1,0}]$.

- In total we will have $n - p$ multi-level vectors and the transform matrix $\mathbf{W} \in \mathbb{R}^{(p-n)\times n}$ is built as $\mathbf{W} := [\mathbf{W}_t, \ldots, \mathbf{W}_0, \mathbf{W}_{-1}]^T$.

- Now, it is clear that $\mathbf{W}\mathbf{g} = \mathbf{0}$ for any $\mathbf{g} \in \mathbf{M}_f$. To complete the basis to span $\mathbb{R}^n$ we need p more orthonormal vectors. In Castrillón-Candás et al. (2013) it is shown how to compute such a basis and stack the vectors as rows in the matrix $\mathbf{L} \in \mathbb{R}^{p\times n}$.

With the construction of $\mathbf{W}$ and $\mathbf{L}$ we will have the following properties: a) the matrix $\mathbf{P} := \begin{bmatrix} \mathbf{W} \\ \mathbf{L} \end{bmatrix}$ is orthonormal, i.e., $\mathbf{P}\mathbf{P}^{\mathrm{T}} = \mathbf{I}_n$; b) any vector $\mathbf{v} \in \mathbb{R}^n$ can be written as $\mathbf{v} = \mathbf{L}^{\mathrm{T}}\mathbf{v}_L + \mathbf{W}^{\mathrm{T}}\mathbf{v}_W$ where $\mathbf{v}_L \in \mathbb{R}^p$ and $\mathbf{v}_W \in \mathbb{R}^{n-p}$ are unique; c) the matrix $\mathbf{W}$ contains at most $\mathcal{O}(nt)$ non-zero entries and $\mathbf{L}$ contains at most $\mathcal{O}(np)$ non-zero entries. This implies that for any vector $\mathbf{v} \in \mathbb{R}^n$ the computational cost of applying $\mathbf{W}\mathbf{v}$ is at most $\mathcal{O}(nt)$ and $\mathbf{L}\mathbf{v}$ is at most $\mathcal{O}(np)$.

4 Multi-Level Covariance Matrix

In this section we show how we can use the matrix $\mathbf{W}$ to produce a highly sparse representation of $\mathbf{C}_W(\boldsymbol{\theta})$ with a level-dependent tapering technique.

With the MB we can transform the observation data vector $\mathbf{Z}$ by applying the matrix $\mathbf{W}$. This leads to the multi-level log-likelihood function (6). The covariance matrix $\mathbf{C}(\boldsymbol{\theta})$ is now transformed into $\mathbf{C}_W(\boldsymbol{\theta})$ with the structure shown in Figure 2 where each of the blocks $\mathbf{C}_W^{i,j} = \mathbf{W}_i\mathbf{C}(\boldsymbol{\theta})\mathbf{W}_j^{\mathrm{T}}$ for all $i, j = 0, \ldots, t$. This implies that the entries of the matrix $\mathbf{C}_W^{i,j}$ are formed from

all the interactions of the MB vectors between level i and j. Thus for any $\boldsymbol{\psi}_{\tilde{l}}^{i,k}$ and $\boldsymbol{\psi}_{\tilde{k}}^{j,l}$ vectors there is a unique entry of $\mathbf{C}_W^{i,j}$ of the form $(\boldsymbol{\psi}_{\tilde{k}}^{i,k})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}_{\tilde{l}}^{j,l}$. The blocks $\mathbf{C}_W^{i,j}$, where $i = -1$ or $j = -1$, correspond to the case where the accuracy term $\tilde{p} > p$.

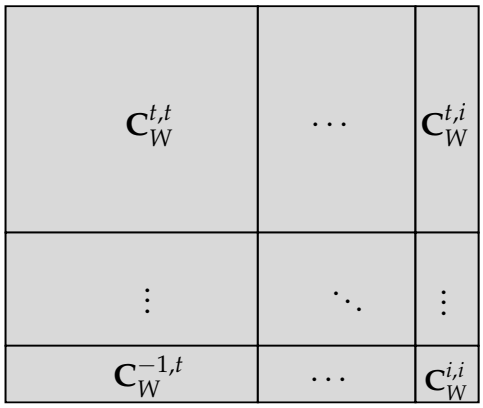


Figure 2: Organization of the multi-level covariance matrix $\mathbf{C}_W(\boldsymbol{\theta})$. For this figure $i = -1$.

The following lemma relates the covariance function ϕ, the degree $\tilde{f}$ (corresponding to the accuracy parameter $\tilde{p}$) of the design matrix $\mathbf{M}_{\tilde{f}}$ to the decay of the entries of the matrix $\mathbf{C}_W(\boldsymbol{\theta})$.

Lemma 1 *Let $B_{\mathbf{a}}$ be the smallest ball in $\mathbb{R}^d$ with radii $r_{\mathbf{a}}$ centered around the midpoint $\mathbf{a} \in \mathbb{R}^d$ of the cube B_l^i such that $B_l^i \subset B_{\mathbf{a}}$. Similarly, let $B_{\mathbf{b}}$ be the smallest ball in $\mathbb{R}^d$ with radii $r_{\mathbf{b}} \in \mathbb{R}^d$ centered around the midpoint $\mathbf{b}$ of the cube B_k^j such that $B_k^j \subset B_{\mathbf{b}}$. Now, since $\boldsymbol{\psi}_{\tilde{l}}^{i,l} \in W_i(\mathbb{S})$ and $\boldsymbol{\psi}_{\tilde{k}}^{j,k} \in W_j(\mathbb{S})$ satisfy the moment orthogonality condition from equations* (8) *and* (9) *for all $\mathbf{g} \in \mathcal{P}^{\tilde{p}}(\mathbb{S})$ then the following bound holds:*

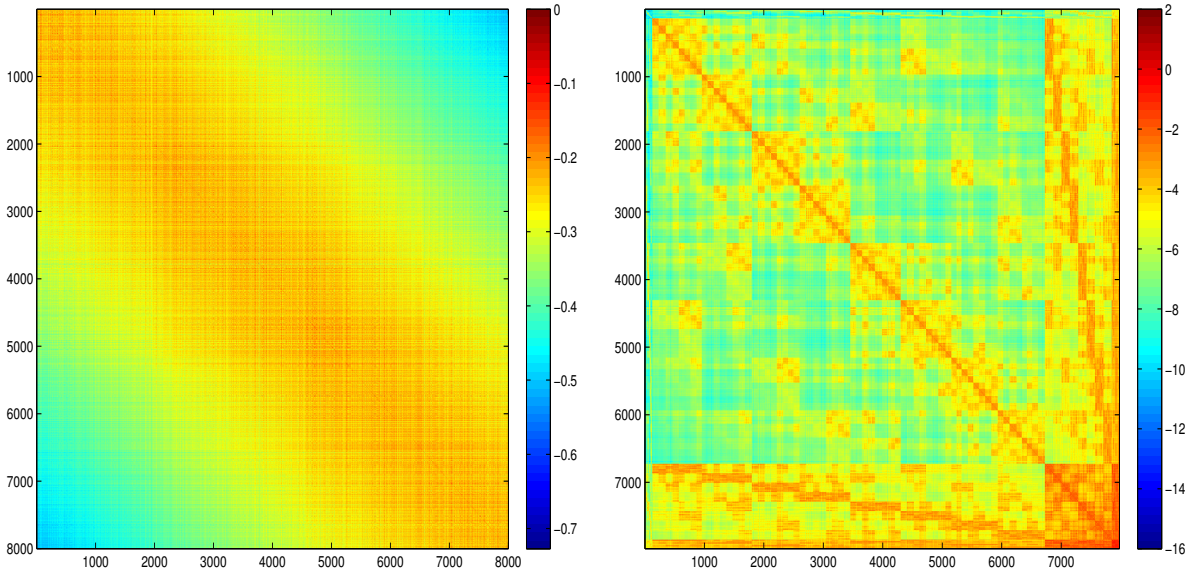
$$|(\boldsymbol{\psi}_{\tilde{k}}^{i,k})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}_{\tilde{l}}^{j,l}| \leq \sum_{|\alpha|=\tilde{f}+1} \sum_{|\beta|=\tilde{f}+1} \frac{r_a^\alpha}{\alpha!}\frac{r_b^\beta}{\beta!} \sup_{\mathbf{x}\in B_{\mathbf{a}}, \mathbf{y}\in B_{\mathbf{b}}} |D_{\mathbf{x}}^\alpha D_{\mathbf{y}}^\beta \phi(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})|, \tag{10}$$

for $i, j = 0, \ldots, t$.

From Lemma 1 we observe that the decay of the entries of $\mathbf{C}_W(\boldsymbol{\theta})$ is dependent on the magnitude of the derivatives of the covariance function $\phi(r; \boldsymbol{\theta})$, the size of $B_{\mathbf{a}}$ and $B_{\mathbf{b}}$ and the degree of $\mathcal{Q}_{\tilde{f}}^d$. Thus if $\phi(r; \boldsymbol{\theta})$ is smooth on $B_{\mathbf{a}}$ and $B_{\mathbf{b}}$ the entries of $\mathbf{C}_W(\boldsymbol{\theta})$ will be small.

Example 1 *In Figure 3 we show a comparison between (a) the covariance matrix $\mathbf{C}(\boldsymbol{\theta})$ and (b) the*

multi-level covariance matrix $\mathbf{C}_W(\boldsymbol{\theta})$ *for the following example: 1)* $\phi(r;\boldsymbol{\theta}) := \exp(-r)$ *and* $d = 3$*. 2) The observation locations (*$n = 8000$*) are sampled from a uniform distribution on the unit cube* $[0,1]^3$*. The actual values of the observations are not necessary for this example. 3)* $f = 3$ *(leading to* $p = 20$ *monomials). 4) We sort the* x_1 *direction location from 0 to 1. This is done for visualization reasons so that we may observe the decay in the matrix* $\mathbf{C}(\boldsymbol{\theta})$*. Notice that the decay of* $\mathbf{C}(\boldsymbol{\theta})$ *is dependent on*



(a) $\log_{10}\operatorname{abs}(\mathbf{C}(\boldsymbol{\theta}))$ (b) $\log_{10}\operatorname{abs}(\mathbf{C}_W(\boldsymbol{\theta}))$

Figure 3: Covariance matrix comparison between covariance matrices (a) $\log_{10}\operatorname{abs}(\mathbf{C}(\boldsymbol{\theta}))$ and (b) $\log_{10}\operatorname{abs}(\mathbf{C}_W(\boldsymbol{\theta}))$ for the exponential covariance function $\phi(r) = \exp(-r)$ with $n = 8000$, $p = 20$ and $d = 3$.

the covariance function $\phi(r;\boldsymbol{\theta})$*. It is clear that for this case a tapering technique would not be very effective as most of the entries are comparable in magnitude. In contrast a few of the entries of* $\mathbf{C}_W(\boldsymbol{\theta})$ *with high magnitude are concentrated around particular regions while most of the entries have very small magnitudes making a hierarchical tapering technique to sparsify the matrix a viable option.*

To produce a sparse matrix from $\mathbf{C}_W(\boldsymbol{\theta})$ we execute the following multi-level tapering technique:

- For all cubes B_k^i at level i let $L_k^{i,0} := B_k^i$ and $L_k^{i,j} := L_k^{i,j-1} \cup \{$union of all boxes at level i that share a face or corner with $L_k^{i,j-1}\}$ for $j = 0, 1, \ldots$. A construction example is shown

in Figure 4 for level i. Now, perform this construction for $i = 0, \ldots, t$.

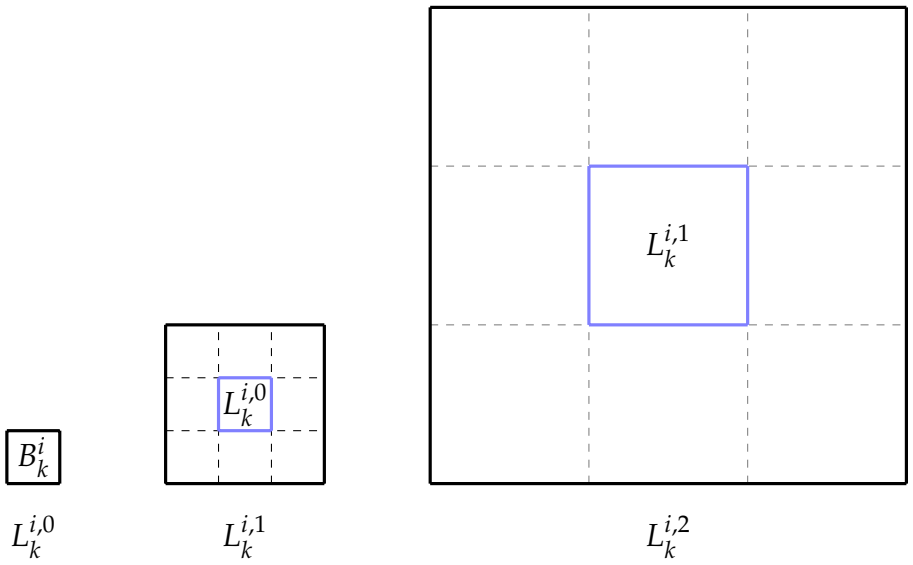


Figure 4: Construction of expanded boxes $L_k^{i,\tau}$, $\tau = 0, 1, \ldots$ from initial box B_k^i.

- Set a user given constant $\tau \in \mathbb{N}_0$
- The entry of $\mathbf{C}_W(\boldsymbol{\theta})$ corresponding to $(\boldsymbol{\psi}_{\tilde{k}}^{i,k})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}_{\tilde{l}}^{j,l}$, for $i, j = 0, \ldots, \tau$ is computed if the following level dependent criterion is true: If ($j \geq i$ and $B_l^j \subset L_k^{i,\tau}$) or ($j < i$ and $B_k^i \subset L_l^{j,\tau}$) is true for the given $\tau \in \mathbb{N}_0$ then compute the entry $(\boldsymbol{\psi}_{\tilde{k}}^{i,k})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}_{\tilde{l}}^{j,l}$.
- For the case that $i = -1$ or $j = -1$ the entry corresponding to $(\boldsymbol{\psi}_{\tilde{k}}^{i,k})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}_{\tilde{l}}^{j,l}$ is always computed.

From this distance criterion we can apriori determine which entries of the sparse matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ are to be computed. For any given row of the matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ corresponding to level i and index k construct the expanded box $L_k^{i,\tau}$. Now, for $j = i, \ldots, t$ find all the boxes B_l^j with the corresponding index l that are contained in $L_k^{i,\tau}$ (See Figure 5). For $j = 0, \ldots, i-1$ find all the extended boxes $L_l^{j,\tau}$ such that $B_k^i \subset L_l^{j,\tau}$. For $i, j = 0, \ldots, t$ this action can be performed efficiently by using the tree shown in Figure 1.

With this criterion we can produce a highly sparse matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ that is close to $\mathbf{C}_W(\boldsymbol{\theta})$ in the matrix 2-norm sense.

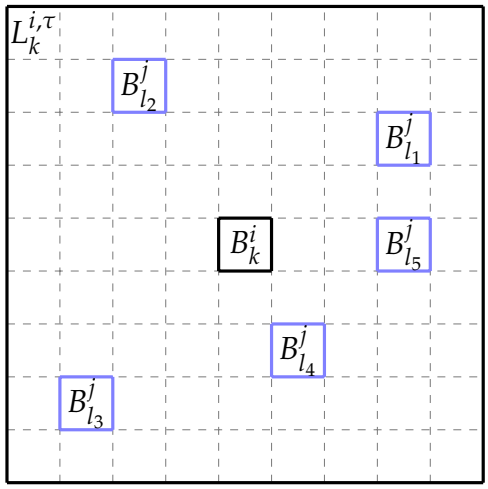


Figure 5: Example of finding all the boxes B_l^j that are contained in $L_k^{i,\tau}$ for $\tau = 2$.

Remark 3 *The error* $\|\tilde{\mathbf{C}}_W(\boldsymbol{\theta}) - \mathbf{C}_W(\boldsymbol{\theta})\|_2$ *will be monotonically decreasing with respect to the smoothness of the covariance function, the size of the degree of the multi-level basis* $\tilde{f} \geq f$ *(accuracy parameter* $\tilde{p} \geq p$*) and the size of* τ*. For a sufficiently large* τ *and* $\tilde{f}$ *the error* $\|\tilde{\mathbf{C}}_W(\boldsymbol{\theta}) - \mathbf{C}_W(\boldsymbol{\theta})\|_2$ *will be small and the matrix* $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ *becomes positive definite.*

Now, the number of nonzeros of $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ *will increase as we increase* τ *and* $\tilde{p}$*. To be able to determine the size for* τ *and* $\tilde{p} \geq p$ *it is helpful to derive an expression for the error* $\|\tilde{\mathbf{C}}_W(\boldsymbol{\theta}) - \mathbf{C}_W(\boldsymbol{\theta})\|_2$ *vs the number of non zeros of* $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$*.*

Error estimates can be derived for $\|\tilde{\mathbf{C}}_W(\boldsymbol{\theta}) - \mathbf{C}_W(\boldsymbol{\theta})\|_2$ *with respect to the smoothness of the covariance function,* $\tilde{p}$ *and* τ*, but this is beyond the scope of the present paper. In practice for the polynomial based model* $\mathcal{Q}_{\tilde{f}}^d$ *we set the level dependent criterion parameter* $\tau := 1$ *and increase* $\tilde{f}$ *(and* $\tilde{p}$*) until at least* $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ *is positive definite. Moreover, the sparse Cholesky factorization code in the Suite Sparse package (Chen et al. (2008); Davis and Hager (2009, 2005, 2001, 1999)) that is used in this paper informs the user if the matrix is not positive definite.*

In Castrillón-Candás et al. (2013) the authors described how to apply a Kernel Independent Fast Multipole Method (KIFMM) by Ying et al. (2004) to compute all the diagonal blocks

$\tilde{\mathbf{C}}_W^{i,i}(\boldsymbol{\theta})$ for $i = 0, \ldots, t$ in $\mathcal{O}(nt)$ computational steps to a fixed accuracy $\varepsilon_{FM} > 0$. This approach can be easily extended to compute all the blocks $\tilde{\mathbf{C}}_W^{i,j}(\boldsymbol{\theta})$ for $i, j = -1, \ldots, t$ in $\mathcal{O}(n(t+1)^2)$.

Remark 4 *The KIFMM by Ying et al. (2004) is very flexible as it allows a large class of covariance functions to be used including the exponential, Gaussian and Matérn. However, the computational efficiency is mostly dependent on the implementation of the covariance function (since the KIFMM computational cost is $\mathcal{O}(n)$) and the accuracy parameter of the solver. For all the numerical experiments in this paper the accuracy parameter is set to medium (10^{-6} to 10^{-8}) or high (10^{-8} or higher).*

Due to the lack of a fast math C++ library for the Matérn covariance function, we create a Hermite cubic spline interpolant of the covariance function with the multithreaded Intel Math Kernel Library (MKL) data fitting package. To generate a compact representation of the interpolant we implement an h-adaptive mesh generator in 1D such that the absolute error over the range $(0, 2.5]$ is less than TOL. From Elden et al. (2004) given that the covariance function $\phi(r; \boldsymbol{\theta}) \in C^4(\mathbb{R})$, $r \in \mathbb{R}$, on each mesh element (starting at $x_0 \in \mathbb{R}$) with length h we can guarantee that the absolute error for the cubic Hermite interpolant is less than TOL if $\frac{h^4}{384} \max_{x \in [x_0, x_0+h]} \phi^{(4)}(x; \boldsymbol{\theta}) < TOL$, where $\phi^{(4)}$ refers to the fourth derivative with respect to x. In this work we set $TOL = 5 \times 10^{-9}$. Numerical test confirmed TOL accuracy for the Matérn covariance function with less than 200 adaptive mesh nodes. This is sufficient for the numerical examples in this paper.

In Figure 6 we show an example of a sparse matrix produced for $\tau = 1$ for $n = 8,000$ observation locations sampled from a uniform distribution on the unit cube. Notice that the entries of the matrix that are not covered by the sparsity pattern are around 10^{-7} times smaller, implying the hierarchical sparsity technique will lead to good accuracy.

The total sparsity for this example is 46% (23% since the matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ is symmetric), however, the sparsity density improves significantly as n increases as we expect the number of non-zero entries of $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ to increase at most as $\mathcal{O}((t+2)^2 n)$ with the number of observations n (See Castrillón-Candás et al. (2013)).

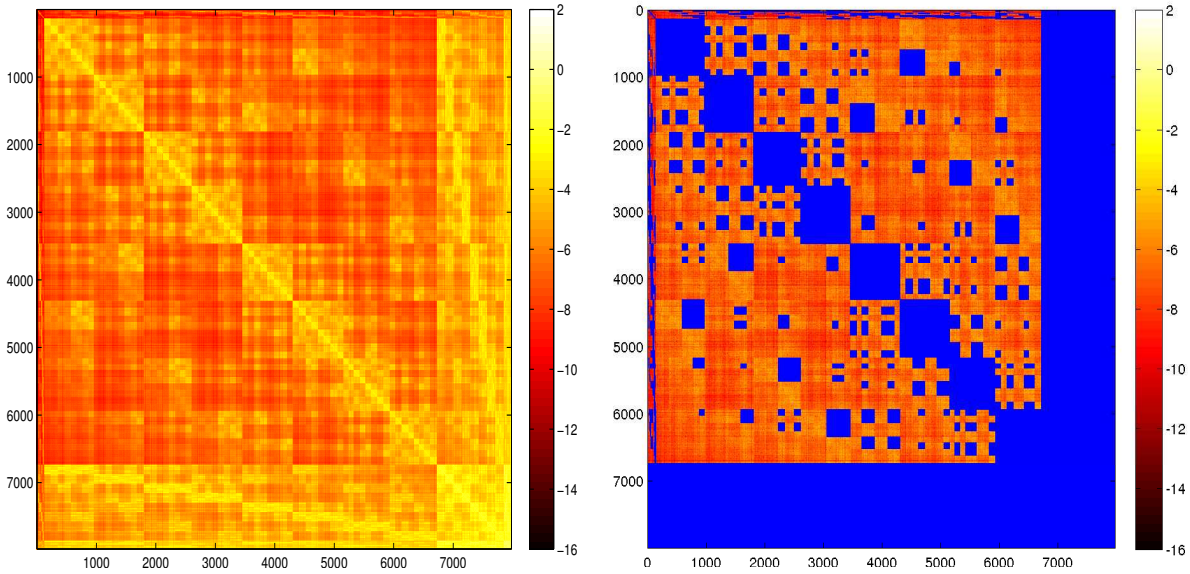
In Figure 7 the sparsity pattern of the matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ is shown for $n = 64,000$ observation locations sampled from a uniform distribution on the unit cube. For this case the design matrix $\mathbf{M}_f$ is constructed from $p = 20$ monomials (i.e. up to cubic polynomials) and $\tau = 1$. The sparsity of this example is 8.2% (4.1 % since the matrix $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ is symmetric).

5 Multi-Level Estimator

As the result section shows it is not necessary to compute the entire sparse matrix $\mathbf{C}_W(\boldsymbol{\theta})$ to obtain a good estimate of the covariance parameter $\boldsymbol{\theta}$. Due to the multi-resolution properties of the MB we can construct a partial multi-resolution likelihood function that is effective.

We can produce a series of multi-resolution likelihood functions $\tilde{\ell}^i_W(\boldsymbol{\theta})$, $i = -1, \ldots t$ by applying the partial transform $[\mathbf{W}_t^{\mathrm{T}}, \ldots, \mathbf{W}_i^{\mathrm{T}}]$ to the data $\mathbf{Z}$, thus

$$\tilde{\ell}^i_W(\boldsymbol{\theta}) = -\frac{\tilde{n}}{2}\log(2\pi) - \frac{1}{2}\log\det\{\tilde{\mathbf{C}}^i_W(\boldsymbol{\theta})\} - \frac{1}{2}(\tilde{\mathbf{Z}}^i_W)^{\mathrm{T}}\tilde{\mathbf{C}}^i_W(\boldsymbol{\theta})^{-1}\tilde{\mathbf{Z}}^i_W, \tag{11}$$



(a) $\log_{10} abs(\mathbf{C}_W(\boldsymbol{\theta}))$ (b) Overlayed sparsity pattern

Figure 6: Sparsity pattern overlayed on $\mathbf{C}_W(\boldsymbol{\theta})$. Notice that most of the entries that are not covered by the blue boxes are around 10^{-7} times smaller in magnitude than the covered entries.

where $\tilde{\mathbf{Z}}_W^i := [\mathbf{W}_t^{\mathrm{T}}, \ldots, \mathbf{W}_i^{\mathrm{T}}]^{\mathrm{T}}\mathbf{Z}$, $\tilde{n}$ is the length of $\tilde{\mathbf{Z}}_W^i$ and $\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})$ is the $\tilde{n} \times \tilde{n}$ upper-left submatrix of $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$.

5.1 Computation of $\log\det\{\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})\}$

Since $\mathbf{C}(\boldsymbol{\theta})$ is symmetric positive definite from (7) it can be shown that $\mathbf{C}_W(\boldsymbol{\theta})$ is also symmetric positive definite. It can also be shown that for a sufficiently large τ and/or $\tilde{p}$ the matrix $\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})$ will also be symmetric positive definite. An approach to computing the determinant of $\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})$ is to apply a sparse Cholesky factorization technique such that $\mathbf{G}\mathbf{G}^{\mathrm{T}} = \tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})$ where $\mathbf{G}$ is a lower triangular matrix. Since the eigenvalues of $\mathbf{G}$ are located on the diagonal we have that $\log\det\{\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})\} = 2\sum_{i=1}^{\tilde{n}} \log \mathbf{G}_{ii}$.

To reduce the fill-in of the factorization matrix $\mathbf{G}$ we apply the matrix reordering technique in Suite Sparse 4.2.1 package (Chen et al. (2008); Davis and Hager (2009, 2005, 2001, 1999)) with the Nested Dissection (NESDIS) function package. The sparse Cholesky factorization is performed with the *lchol* command from Suite Sparse 4.2.1 package.

Although in practice the combined NESDIS and sparse Cholesky factorization is highly

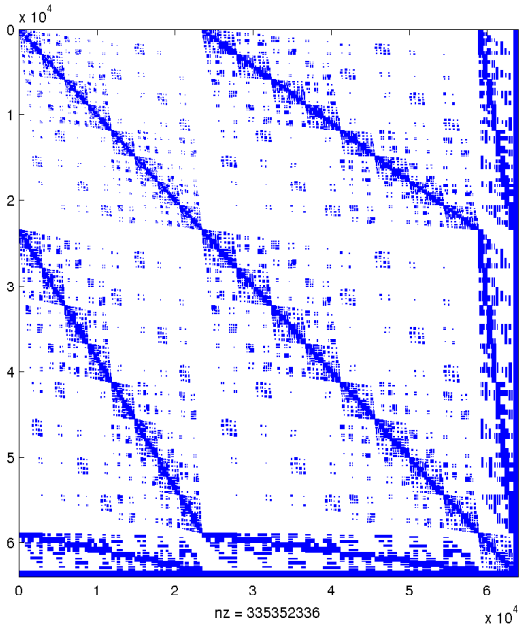


Figure 7: Sparsity pattern (8.2 % non zeros) for $\tilde{\mathbf{C}}_W$ with $\tau = 1$ and $n = 64,000$.

efficient, as shown by our numerical results, a worse case complexity bound can be obtained. For example, it can be shown that by using the planar graph separation theorem (see George (1973), Gilbert and Tarjan (1987)) a worse case complexity of $\mathcal{O}(n^{3/2})$ and $\mathcal{O}(n(\log n)^2)$ storage is achieved in 2D. Similarly, the worse case complexity in 3D is $\mathcal{O}(n^2)$.

Example 2 *Continuing Example 1 we compute* $\log\det\{\mathbf{C}_W(\boldsymbol{\theta})\}$ *and the approximation* $\log\det\{\tilde{\mathbf{C}}_W(\boldsymbol{\theta})\}$ *for* $\tau = 0, 1, 2, \infty$ *by applying the sparse Cholesky factorization. In Table 1 we tabulated the absolute* $\varepsilon_{abs} := |\log\det\{\tilde{\mathbf{C}}_W(\boldsymbol{\theta})\} - \log\det\{\mathbf{C}_W(\boldsymbol{\theta})\}|$ *and relative* $\varepsilon_{rel} := \frac{\varepsilon_{abs}}{|\log\det\{\mathbf{C}_W(\boldsymbol{\theta})\}|}$ *errors. For* $\tau = 0$ *we obtain a very sparse matrix (4% density), but leads to a non-positive definite matrix, which is not valid for the computation of the determinant. For* $\tau = 1$ *the matrix* $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ *becomes positive definite. As we increase* τ *the approximation* $\log\det\{\tilde{\mathbf{C}}_W(\boldsymbol{\theta})\}$ *becomes more accurate. However, the density of* $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ *also increases.*

Table 1: Log determinant errors comparisons between $\tilde{\mathbf{C}}_W(\boldsymbol{\theta})$ and $\mathbf{C}_W(\boldsymbol{\theta})$.

τ	density (%)	$\log\det\{\tilde{\mathbf{C}}_W(\boldsymbol{\theta})\}$	ε_{abs}	ε_{rel}
0	4	not positive definite	–	–
1	23	-5.184354×10^3	2.23×10^{-2}	4.31×10^{-6}
2	38	-5.184332×10^3	3.78×10^{-4}	7.29×10^{-8}
∞	50	-5.184331×10^3	0	0

5.2 Computation of $(\tilde{\mathbf{Z}}_W^i)^{\mathrm{T}}\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})^{-1}\tilde{\mathbf{Z}}_W^i$

We have two choices for the computation of $(\tilde{\mathbf{Z}}_W^i)^{\mathrm{T}}\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})^{-1}\tilde{\mathbf{Z}}_W^i$. We can either use a Cholesky factorization of $\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})$ or Preconditioned Conjugate Gradient (PCG) coupled with a KIFMM. The PCG choice requires significantly less memory and allows more control of the error. However, we already compute the sparse Cholesky factorization of $\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})$ for the computation of the determinant. Thus we can use the same factors to compute $(\tilde{\mathbf{Z}}_W^i)^{\mathrm{T}}\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})^{-1}\tilde{\mathbf{Z}}_W^i$.

6 Multi-Level Kriging

An alternative formulation for obtaining the estimate $\hat{\mathbf{Z}}(\mathbf{s}_0)$ is by solving the following problem

$$\begin{pmatrix} \mathbf{C}(\boldsymbol{\theta}) & \mathbf{M}_f \\ \mathbf{M}_f^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\gamma}} \\ \hat{\boldsymbol{\beta}} \end{pmatrix} = \begin{pmatrix} \mathbf{Z} \\ \mathbf{0} \end{pmatrix}. \tag{12}$$

It is not hard to show that the solution of this problem leads to equation (4) and $\hat{\boldsymbol{\gamma}}(\boldsymbol{\theta}) = \mathbf{C}^{-1}(\boldsymbol{\theta})\{\mathbf{Z} - \mathbf{M}_f\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})\}$ or alternatively $\hat{\boldsymbol{\beta}} = (\mathbf{M}_f^{\mathrm{T}}\mathbf{M}_f)^{-1}\mathbf{M}_f(\mathbf{Z} - \mathbf{C}(\boldsymbol{\theta})\hat{\boldsymbol{\gamma}})$. The best unbiased predictor is evaluated as

$$\hat{Z}(\mathbf{s}_0) = \mathbf{m}(\mathbf{s}_0)^{\mathrm{T}}\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) + \mathbf{c}(\boldsymbol{\theta})^{\mathrm{T}}\hat{\boldsymbol{\gamma}}(\boldsymbol{\theta}) \tag{13}$$

and the Mean Squared Error (MSE) at the target point $\mathbf{s}_0$ is given by

$$1 + \tilde{\mathbf{u}}^{\mathrm{T}}(\mathbf{M}_f^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{M}_f)^{-1}\tilde{\mathbf{u}} - \mathbf{c}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{\theta})\mathbf{c}(\boldsymbol{\theta}) \tag{14}$$

where $\tilde{\mathbf{u}}^{\mathrm{T}} := (\mathbf{M}_f\mathbf{C}^{-1}(\boldsymbol{\theta})\mathbf{c}(\boldsymbol{\theta}) - \mathbf{m}(\mathbf{s}_0))$.

The computational cost for computing $\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})$, $\hat{\boldsymbol{\gamma}}(\boldsymbol{\theta})$ and the MSE accurately using a direct method is $\mathcal{O}(n^3)$, which is unfeasible for large size problems. We propose a much faster approach.

From (12) we observe that $\mathbf{M}_f^{\mathrm{T}}\hat{\boldsymbol{\gamma}}(\boldsymbol{\theta}) = \mathbf{0}$. This implies that $\hat{\boldsymbol{\gamma}} \in \mathbb{R}^n \backslash \mathcal{P}^p(\mathrm{S})$ and can be uniquely rewritten as $\hat{\boldsymbol{\gamma}} = \mathbf{W}^{\mathrm{T}}\boldsymbol{\gamma}_W$ for some $\boldsymbol{\gamma}_W \in \mathbb{R}^{n-p}$. We can rewrite $\mathbf{C}(\boldsymbol{\theta})\hat{\boldsymbol{\gamma}} + \mathbf{M}_f\hat{\boldsymbol{\beta}} = \mathbf{Z}$ as

$$\mathbf{C}(\boldsymbol{\theta})\mathbf{W}^{\mathrm{T}}\hat{\boldsymbol{\gamma}}_W + \mathbf{M}_f\hat{\boldsymbol{\beta}} = \mathbf{Z}. \tag{15}$$

Now apply the matrix $\mathbf{W}$ to equation (15) and we obtain $\mathbf{W}\{\mathbf{C}(\boldsymbol{\theta})\mathbf{W}^{\mathrm{T}}\hat{\boldsymbol{\gamma}}_W + \mathbf{M}_f\hat{\boldsymbol{\beta}}\} = \mathbf{W}\mathbf{Z}$. Since $\mathbf{W}\mathbf{M}_f = \mathbf{0}$ then $\mathbf{C}_W(\boldsymbol{\theta})\hat{\boldsymbol{\gamma}}_W = \mathbf{Z}_W$. Applying the preconditioner $\mathbf{D}_W^{-1}(\boldsymbol{\theta})$, where $\mathbf{D}_W(\boldsymbol{\theta}) := \mathrm{diag}(\mathbf{C}_W(\boldsymbol{\theta}))$, we have the system of equations $\bar{\mathbf{C}}_W(\boldsymbol{\theta})\bar{\boldsymbol{\gamma}}_W(\boldsymbol{\theta}) = \bar{\mathbf{Z}}_W$ where $\bar{\boldsymbol{\gamma}}_W(\boldsymbol{\theta}) := \mathbf{D}_W(\boldsymbol{\theta})\boldsymbol{\gamma}_W(\boldsymbol{\theta})$, $\bar{\mathbf{C}}_W(\boldsymbol{\theta}) := \mathbf{D}_W^{-1}(\boldsymbol{\theta})\mathbf{C}_W(\boldsymbol{\theta})\mathbf{D}_W^{-1}(\boldsymbol{\theta})$ and $\bar{\mathbf{Z}}_W := \mathbf{D}_W^{-1}(\boldsymbol{\theta})\mathbf{Z}_W$.

This system of equations is solved by a combination of a KIFMM and PCG. If $\mathbf{C}_W(\boldsymbol{\theta})$ and

$\mathbf{D}_W(\boldsymbol{\theta})$ are symmetric positive definite then an effective method to solve $\bar{\mathbf{C}}_W(\boldsymbol{\theta})\bar{\boldsymbol{\gamma}}_W(\boldsymbol{\theta}) = \bar{\mathbf{Z}}_W$ is the PCG method implemented in PETSc by Balay et al. (2013b,a, 1997).

Lemma 2 *If the covariance function ϕ is positive definite, then the matrix $\mathbf{D}_W(\boldsymbol{\theta})$ is always symmetric positive definite.*

The matrix-vector products $\mathbf{C}_W(\boldsymbol{\theta})\mathbf{v}$, where $\mathbf{v} \in \mathbb{R}^{n-p}$, are computed in $\mathcal{O}(n)$ computational steps to a fixed accuracy $\varepsilon_{FM} > 0$. The total computational cost is $\mathcal{O}(kn(t+2))$, where k is the number of iterations needed to solve $\bar{\mathbf{C}}_W(\boldsymbol{\theta})\bar{\boldsymbol{\gamma}}_W(\boldsymbol{\theta}) = \bar{\mathbf{Z}}_W$ to a predetermined accuracy $\varepsilon_{PCG} > 0$.

It is important to point out that the introduction of a preconditioner can degrade the performance of the PCG, in particular, if the preconditioner is ill-conditioned. The accuracy of the PCG method ε_{PCG} has to be set such that the accuracy of the *unpreconditioned* system $\mathbf{C}_W(\boldsymbol{\theta})\boldsymbol{\alpha}_W(\boldsymbol{\theta}) = \mathbf{Z}_W$ is below a user given tolerance $\varepsilon > 0$.

We compute $\hat{\boldsymbol{\gamma}} = \mathbf{W}^{\mathrm{T}}\boldsymbol{\gamma}_W$ and $\hat{\boldsymbol{\beta}} = (\mathbf{M}_f^{\mathrm{T}}\mathbf{M}_f)^{-1}\mathbf{M}_f(\mathbf{Z} - \mathbf{C}(\boldsymbol{\theta})\hat{\boldsymbol{\gamma}})$ in $\mathcal{O}(np^3)$ computational steps. The matrix vector product $\mathbf{c}(\boldsymbol{\theta})^{\mathrm{T}}\hat{\boldsymbol{\gamma}}(\boldsymbol{\theta})$ is computed $\mathcal{O}(n)$ steps. Finally, the total cost for computing the estimate $\hat{\mathbf{Z}}(\mathbf{s}_0)$ from (13) is $\mathcal{O}(n + np + np^2 + p^3 + kn(t+2))$.

In Appendix C we show a procedure to compute the MSE fast.

7 Numerical Study and Statistical Examples

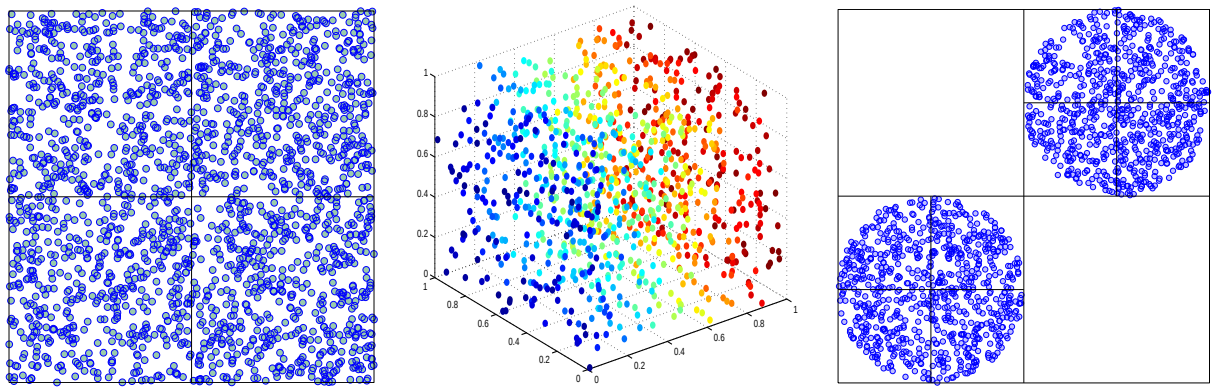
In this section we test the numerical efficiency and accuracy of our solver for computing the terms $\log\det\{\tilde{\mathbf{C}}_W^i(\boldsymbol{\theta})\}$ and $\bar{\boldsymbol{\gamma}}_W(\hat{\boldsymbol{\theta}}) = \bar{\mathbf{C}}_W^{-1}(\hat{\boldsymbol{\theta}})\bar{\mathbf{Z}}_W$ for Matérn covariances. Our results show that we are able to solve problems of up to 128,000 observations and kriging up to 512,000 size problems with good accuracy. Our approach is not limited to 128,000 for parameter estimation. This was the maximum we could test due to the memory limitation on our workstation in creating observations larger than 128,000. We now describe the data sets.

Data set #1 and #2: The sets of observation locations $\mathbf{S}_1^d, \ldots, \mathbf{S}_{10}^d$ vary from 1,000 to 512,000 and we assume that $\mathbf{S}_l^d \subset \mathbf{S}_{l+1}^d$ for $l = 1, \ldots, 9$ for $d = 2$ and $d = 3$. The observations locations

are sampled from a uniform distribution over the unit square $[0,1]^2$ for $d = 2$ (data set #1) and for $[0,1]^3$ for $d = 3$ (data set #2), as shown in Figure 8(a). The target points $\mathbf{s}_0$ are set to 1000 random points across the domain $[0,1]^2$ (data set #1) and $[0,1]^3$(data set #2). We shall refer to $\mathbf{Z}_1^d, \ldots \mathbf{Z}_{10}^d$ as the observation values associated with $\mathbf{S}_1^d, \ldots \mathbf{S}_{10}^d$.

Data set #3: We take the data set generated by $\mathbf{S}_9^d$ for $d = 2$ (256,000 observation points) and carve out two disks located at (1/4,1/4) and (3/4,3/4) with radii 1/4. This generates 100,637 observation points; see Figure 8(c) for an example with 1,562 observation points randomly extracted from the data set.

We now test our approach on the Matérn covariance function $\phi(r;\boldsymbol{\theta}) := \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\sqrt{2\nu}\frac{r}{\rho}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{r}{\rho}\right)$, where Γ is the gamma function and K_ν is the modified Bessel function of the second kind. All results are executed on a single CPU (4 core Intel i7-3770 CPU @ 3.40GHz.) with Linux Ubuntu 13.04.



(a) (b) (c)

Figure 8: Data set examples: (a) Data set #1: One thousand observation points randomly generated from a uniform distribution on $[0,1]^2$. (b) Data set #2: One thousand observation points randomly generated from a uniform distribution on $[0,1]^3$. The color of the observation locations represent the distance along the right axis coordinate. (c) Data set #3: Two disks of 1562 randomly generated observation locations. The disks are contained in a square but are represented in a multi-level representation.

7.1 Parameter Estimation

In this section we present the results for data set #1 and #3 for the Matérn parameter estimation.

Suppose we have two realizations of the Gaussian spatial random field $Z(\mathbf{s})$ with the Matérn covariance for data set #1 (2D). We set $f = 3$ ($p = 10$) and $\tilde{f} = 4, 5, 6$ (corresponding to $\tilde{p} = 15, 21, 28$) and fix the covariance parameters to $\boldsymbol{\theta} = (\nu, \rho) = (3/4, 1/6)$. Two realizations $\mathbf{Z}_6^2$ ($n = 64,000$) and $\mathbf{Z}_7^2$ ($n = 128,000$) are generated from these parameters. For each observation values $\mathbf{Z}_6^2$ and $\mathbf{Z}_7^2$ (and locations) apply the transformation $\mathbf{W}$ to compute $\mathbf{Z}_{W,6}^d$ and $\mathbf{Z}_{W,7}^d$ and solve the optimization problems $\hat{\boldsymbol{\theta}}_i := \text{argmax}_{\boldsymbol{\theta} \in (0,\infty)\times(0,\infty)} \hat{\ell}_W(\mathbf{Z}_{W,i}, \boldsymbol{\theta})$ for $i = 6$ and 7.

The optimization problem from the log-likelihood function (6) is solved using *fminsearch* from the optimization toolbox in MATLAB with the local minimizer search for ν in the interval $[1/2, 5/4]$ and ρ in the interval $[1/5, 1/7]$. We set the parameter criterion to $\tau := 1$, and the *fminsearch* tolerance is set to 10^{-3}. In Table 2 we tabulate the results for the parameter estimates $\hat{v}$ and $\hat{\rho}$ for different problem sizes of data sets #1 and #3 for the user defined parameters $\tilde{f}$ for the construction of the MB and i for the construction of the matrix $\mathbf{C}_W^i$. We notice that the estimates of (ν, ρ) seem to approach the actual values as we increase the number of observations. In particular, for $n = 128,000$ the estimate $(\hat{v}, \hat{\rho}) := (0.7498, 0.1672)$ is very close to the actual noise model parameters $(\nu, \rho) = (0.75, 1/6)$ of the covariance function. We observe that as we increase the number of levels (i.e. decrease i) in the covariance matrix the absolute error decays until it stagnates, usually by the time that the covariance matrix is for two levels. We also report the wall clock times (i.e. actual time it took the executable to run, not to be confused with CPU time that is unreliable as a measure) for computing each Newton iteration. The total number of Newton iterations is approximately 50.

In Table 2(b) the results for parameter estimation with data set #3 are tabulated. The realization is obtained from the Gaussian random field $Z(\mathbf{s})$ with $n = 100,637$, $\nu = 1.25$ and $\rho = 1/6$. For this case the absolute error is 0.25% for the estimate $\hat{v}$ and 0.04% for $\hat{\rho}$.

In Table 3 we generate $M = 100$ realizations of the stochastic model for data set #1, to analyze the mean and standard deviation of the Matérn covariance parameter estimates. The

mean estimate $\mathbb{E}_M[\hat{v}]$ refers to the mean of M estimates $\hat{v}$ for the the M realizations of the stochastic model. Similarly, $std_M[\hat{v}]$ refers to the the standard deviation of the M realizations. We observe that the mean appears to approach the covariance parameters as we decrease i.

Table 2: Estimation results for data sets #1 and #3. (a) The observation data is generated with covariance parameters $\nu = 0.75$ and $\rho = 1/6$. The degree of the model is $f = 3$ (cubic), which gives $p = 10$ monomials and we set $\tau := 1$. The first column is the size of the problem. Columns 2 and 3 are the parameters that are chosen to build the MB and the multi-level estimator. Columns 4 and 5 are the corresponding $\tilde{p}$ for $\tilde{f}$ and the maximum level. Columns 6 and 7 are the estimates of ν and ρ. Column 8 is the percentage of non-zeros of the Cholesky factors. Column 9 is self-explanatory. Column 10 and 11 are the approximate wall clock computational time (in seconds) needed to compute $\tilde{\mathbf{C}}^i_W(\boldsymbol{\theta})$ and to perform Cholesky factorization respectively. The total time for each Newton iteration is about $t_{cons}(s) + t_{chol}(s)$. For each problem it takes about 50 Newton iterations to converge. (b) Estimation results for data set #3 with observation data generated with covariance parameters $\nu = 1.25$ and $\rho = 1/6$.

(a) Estimation results for data set #3 (2D).

n	$\tilde{f}$	i	$\tilde{p}$	t	$\hat{\nu}$	$\hat{\rho}$	$nz(\mathbf{G})(\%)$	size($\tilde{\mathbf{C}}^i_W$)	$t_{cons}(s)$	$t_{chol}(s)$
64,000	6	6	28	6	0.6741	0.2000	8.9	23	14	0
64,000	6	5	28	6	0.7682	0.1534	1.7	35328	40	1
64,000	6	4	28	6	0.7457	0.1713	4.5	56832	230	11
64,000	6	3	28	6	0.7451	0.1715	10.7	62208	961	65
64,000	5	6	21	6	0.7571	0.1522	0.4	810	13	0
64,000	5	5	21	6	0.7537	0.1640	1.7	42496	43	2
64,000	5	4	21	6	0.7470	0.1715	3.7	58624	220	12
64,000	5	3	21	6	0.7452	0.1713	7.0	62656	750	32
64,000	4	6	15	6	0.7420	0.1765	0.2	7749	13	0
64,000	4	5	15	6	0.7453	0.1710	2.0	48640	53	3
64,000	4	4	15	6	0.7432	0.1729	3.4	60160	161	9
64,000	4	3	15	6	0.7449	0.1715	4.4	63040	550	15
128,000	6	6	28	6	0.7510	0.1655	0.3	17179	75	0
128,000	6	5	28	6	0.7525	0.1647	2.1	99328	350	13
128,000	6	4	28	6	0.7498	0.1672	4.0	120832	1200	70
128,000	5	6	21	6	0.7490	0.1682	0.5	42154	80	0
128,000	5	5	21	6	0.7504	0.1665	1.9	106496	300	14
128,000	5	4	21	6	0.7484	0.1684	3.3	122624	1000	50

(b) Estimation results for data set #3 (2D).

n	f	i	$\tilde{p}$	t	$\hat{\nu}$	$\hat{\rho}$	$nz(\mathbf{G})(\%)$	size($\tilde{\mathbf{C}}^i_W$)	$t_{cons}(s)$	$t_{chol}(s)$
100,637	6	6	12	66	1.3048	0.1429	0.5	2613	60	0
100,637	6	5	12	66	1.2469	0.1687	3.1	72231	600	12

As i is reduced from t to $t-1$ there is a significant drop in the term $std_M[\hat{\nu}]$. However, for $i < t-1$ the standard deviation $std_M[\hat{\nu}]$ does not improve significantly. Therefore, there is not much gain in improving the estimate by decreasing i, which increases the computational cost in computing $\tilde{\mathbf{C}}_W^i$.

7.2 Numerical examples for computing $\mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{Z}_W$ and Kriging

We test our approach for solving the system of equations $\bar{\mathbf{C}}_W(\boldsymbol{\theta})\bar{\gamma}_W = \bar{\mathbf{Z}}_W$ (that we have to solve to obtain the kriging predictor) on the data sets #1 and #2. We also include results showing the kriging prediction error between the multi-level and direct methods.

We first test the PCG method with data set #2 (3D) on three test cases: (a) $\boldsymbol{\theta}_a = (\nu, \rho) = (3/4, 1/6)$, (b) $\boldsymbol{\theta}_b = (1, 1/6)$ and (c) $\boldsymbol{\theta}_c = (5/4, 1/6)$. The value $\rho = 1/6$ gives us an approximate decay of 5% (which is reasonable in practice) from the center of the cube along each dimensional axis. The PCG relative error tolerance $\varepsilon_{PCG} > 0$ is set to a value that leads to a relative error $\epsilon = 10^{-3}$ of the *unpreconditioned* system $\mathbf{C}_W(\boldsymbol{\theta})\hat{\gamma}_W = \mathbf{Z}_W$.

Table 3: (a) Statistical results for data set #1 with observation data generated with covariance parameters $\nu = 0.75$ and $\rho = 1/6$. The parameter M is the number of realizations of the stochastic model, $\mathbb{E}_M[\hat{\nu}]$ is the mean of M estimates of ν and $std_M[\hat{\nu}]$ is the standard deviation of M estimates of ν. As i is reduced from t to $t-1$ there is a significant drop in $std_M[\hat{\nu}]$. However, for $i < t-1$ the standard deviation of the estimates does not improve significantly. This indicates that a good estimate can be obtained for $i = t-1$ and there is not much gain in reducing i, which increases the computational cost in computing $\tilde{\mathbf{C}}_W^i$.

Statistical results for data set #1 with multiple realizations.

n	$\tilde{f}$	i	$\tilde{p}$	t	M	$\mathbb{E}_M[\hat{\nu}]$	$\mathbb{E}_M[\hat{\rho}]$	$std_M[\hat{\nu}]$	$std_M[\hat{\rho}]$
32,000	4	5	15	5	100	0.7494	0.1677	1.3×10^{-2}	1.0×10^{-2}
32,000	4	4	15	5	100	0.7493	0.1674	5.9×10^{-3}	4.5×10^{-3}
32,000	4	3	15	5	100	0.7507	0.1673	5.6×10^{-3}	4.0×10^{-3}
64,000	4	6	15	6	100	0.7507	0.1678	2.0×10^{-2}	1.9×10^{-2}
64,000	4	5	15	6	100	0.7507	0.1661	5.4×10^{-3}	4.6×10^{-3}
64,000	4	4	15	6	100	0.7502	0.1665	3.9×10^{-3}	3.3×10^{-3}
128,000	6	6	28	6	100	0.7487	0.1682	8.3×10^{-3}	7.7×10^{-3}
128,000	6	5	28	6	100	0.7494	0.1673	3.7×10^{-3}	3.3×10^{-3}

In Table 4 we report the total wall clock times and iterations for computing $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\gamma}}$ and the target $\hat{\mathbf{Z}}(\mathbf{s}_0)$ for data set #2 (3D) with the Matérn covariance function . The polynomial accuracy of the model is set to cubic ($f = 3$, $p = 20$) and the accuracy parameter $\tilde{p}$ is set to 20 (which corresponds to $\tilde{f} = 3$). We look at three cases: For (a) ($\boldsymbol{\theta}_a = (3/4, 1/6)$) we set the KIFMM accuracy to medium and the number of iterations increase as $\mathcal{O}(n^{0.58})$. For (b) ($\boldsymbol{\theta}_b = (1, 1/6)$) we set the KIFMM accuracy to medium and the number of iterations increases as $\mathcal{O}(n^{0.58})$. For (c) ($\boldsymbol{\theta}_c = (5/4, 1/6)$) we set the KIFMM accuracy to high and the number of iterations increases as $\mathcal{O}(n^{0.77})$.

In Table 4 we also report the number of iterations needed for solving $\mathbf{C}^{-1}(\boldsymbol{\theta})\mathbf{Z}$ with 10^{-3} accuracy with a Conjugate Gradient (CG) method. In this case the number of iterations is about 10 times larger than the multi-level version. Moreover, for solving the kriging problem (e.g. equation (4)), p such problems have to be solved. Thus, it is at least about 200 times faster since $p = 20$ for this case. An alternative is to solve (12). However, in general it is not positive definite. The matrix is highly ill-conditioned also making it difficult to solve with an iterative solver such as generalized minimal residual method (see Castrillón-Candás et al. (2013)).

In Table 5 the results for computing $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\gamma}}$ and $\hat{Z}(\mathbf{s}_0)$ for 1000 target points $\mathbf{s}_0$ for data set #1. (2D) with the Matérn covariance function are tabulated. We have three test cases: (a) $\boldsymbol{\theta}_a = (\nu, \rho) = (1/2, 1/6)$ (note for this case we obtain an exponential covariance function), (b) $\boldsymbol{\theta}_b = (3/4, 1/6)$, and (c) $\boldsymbol{\theta}_c = (1, 1/6)$. For (a) and (b) the KIFMM accuracy is set to medium. For (c) the KIFMM accuracy is set to high. For this case the relative residual accuracy for the unpreconditioned system is fixed at 10^{-2}.

We note that for this case the result are even more impressive that for the 3D case. For Table 5 (c) the CG solver stagnated and we terminated the iteration after 100,000. At this point the matrix $\mathbf{C}(\boldsymbol{\theta})$ is highly ill-conditioned. In contrast, with $\mathbf{C}_W(\boldsymbol{\theta})$ we are still able to solve the problem even for 128,000 size problem.

As can be observed, the results are not as good as for the 3D case. One of the reasons

Table 4: Diagonal pre-conditioned results for computing $\bar{\mathbf{C}}_W(\boldsymbol{\theta})\bar{\boldsymbol{\gamma}}_W = \bar{\mathbf{Z}}_W$ for data set #2 (3D) with the Matérn covariance $\boldsymbol{\theta} = (\nu, \rho)$. We look at three cases: (a) $\boldsymbol{\theta}_a = (3/4, 1/6)$ (b) $\boldsymbol{\theta}_b = (1, 1/6)$ and (c) $\boldsymbol{\theta}_c = (5/4, 1/6)$. The relative error of the residual of the unpreconditioned system is set to $\varepsilon = 10^{-3}$. The KIFMM is set to medium accuracy for (a) and (b), and set to high accuracy for (c). The second column is the number of iterations needed to obtain 10^{-3} relative error of the unpreconditioned system with $\mathbf{C}_W(\boldsymbol{\theta})$. We denote as itr($\mathbf{C}_W$) as the number of CG iterations needed for convergence until the desired accuracy is achieved. The third column is the number iterations for solving $\mathbf{C}^{-1}(\boldsymbol{\theta})\mathbf{Z}$ with 10^{-3} accuracy. The fourth column is the residual tolerance needed for convergence of 10^{-3} relative error for the unpreconditioned system. The fifth column presents the wall clock times for initialization (basis construction and preconditioner computation). The PCG iteration wall clock times for $\mathbf{C}_W$ are given in the fifth column. The last column presents the total wall clock time to compute $\bar{\boldsymbol{\gamma}}_W = \bar{\mathbf{C}}_W(\boldsymbol{\theta})^{-1}\bar{\mathbf{Z}}_W$.

n	itr($\mathbf{C}_W$)	itr($\mathbf{C}$)	ε_{PCG}	Diag. (s)	Itr (s)	Total (s)
16,000	166	1296	1.02×10^{-4}	80	113	193
32,000	247	3065	9.88×10^{-5}	215	321	536
64,000	372	5517	1.00×10^{-4}	665	1226	1891
128,000	547	-	4.84×10^{-5}	2060	3237	5397
256,000	847	-	5.00×10^{-5}	5775	9885	15660
512,000	1129	-	3.74×10^{-5}	17896	33116	51012

(a) $\boldsymbol{\theta}_a = (3/4, 1/6)$, $d = 3$, $f = 3$ ($p = 20$), $\tilde{f} = 3$ ($\tilde{p} = 20$)

n	itr($\mathbf{C}_W$)	itr($\mathbf{C}$)	ε_{PCG}	Diag. (s)	Itr (s)	Total (s)
16,000	293	2970	8.23×10^{-5}	79	198	277
32,000	470	7786	8.18×10^{-5}	213	607	820
64,000	760	15808	7.09×10^{-5}	662	2495	3157
128,000	1167	-	3.00×10^{-5}	2050	7109	9159
256,000	1961	-	3.27×10^{-5}	5789	22878	28667

(b) $\boldsymbol{\theta}_b = (1, 1/6)$, $d = 3$, $f = 3$ ($p = 20$), $\tilde{f} = 3$ ($\tilde{p} = 20$)

n	itr($\mathbf{C}_W$)	itr($\mathbf{C}$)	ε_{PCG}	Diag. (s)	Itr (s)	Total (s)
16,000	500	5953	6.27×10^{-5}	138	580	718
32,000	827	17029	7.29×10^{-5}	346	1574	1920
64,000	1567	37018	4.45×10^{-5}	910	6474	7384
128,000	2381	-	2.23×10^{-5}	3974	25052	29026
256,000	4299	-	2.61×10^{-5}	10322	72374	82696

(c) $\boldsymbol{\theta}_c = (5/4, 1/6)$, $d = 3$, $f = 3$ ($p = 20$), $\tilde{f} = 3$ ($\tilde{p} = 20$)

behind this is that the observation points are closer to each other, which leads to more ill-conditioned covariance matrices. For $\nu \leq 0.75$ the results look good and are comparable to the 3D case. However for $\nu \geq 1$ the preconditioner starts to suffer, although the convergence rate is still better than quadratic.

Table 5: Diagonal pre-conditioned results for computing $\bar{\mathbf{C}}_W(\boldsymbol{\theta})\bar{\boldsymbol{\gamma}}_W = \bar{\mathbf{Z}}_W$ for the Matérn covariance with $\boldsymbol{\theta} = (\nu, \rho)$ for data set #1 (2D). We look at three cases: (a) $\boldsymbol{\theta}_a = (3/4, 1/6)$, (b) $\boldsymbol{\theta}_b = (1, 1/6)$ and (c) $\boldsymbol{\theta}_c = (5/4, 1/6)$. The relative error of the residual of the unpreconditioned system is set to $\varepsilon = 10^{-2}$. The KIFMM is set to medium accuracy for (a), (b) and set to high accuracy for (c).

n	itr($\mathbf{C}_W$)	itr($\mathbf{C}$)	ε_{PCG}	Diag. (s)	Itr (s)	Total (s)
16,000	330	3603	2.39×10^{-3}	246	115	361
32,000	333	5429	1.39×10^{-3}	750	251	1001
64,000	455	8152	1.32×10^{-3}	1947	589	2536
128,000	564	-	7.10×10^{-4}	5570	1577	7147
256,000	619	-	9.78×10^{-4}	15266	3065	18331
512,000	1230	-	4.50×10^{-4}	42254	13101	55355

(a) $\boldsymbol{\theta}_a = (1/2, 1/6)$, $d = 2$, $f = 3$ ($p = 10$), $\tilde{f} = 12$ ($\tilde{p} = 91$)

n	itr($\mathbf{C}_W$)	itr($\mathbf{C}$)	ε_{PCG}	Diag. (s)	Itr (s)	Total (s)
16,000	965	26795	2.78×10^{-3}	370	397	767
32,000	1110	41079	2.04×10^{-3}	1125	1061	2186
64,000	2239	82166	1.35×10^{-3}	2892	3714	6606
128,000	3443	-	1.09×10^{-3}	8268	13130	21398
256,000	4557	-	7.63×10^{-4}	23175	30302	53477

(b) $\boldsymbol{\theta}_b = (3/4, 1/6)$, $d = 2$, $f = 14$ ($p = 120$), $\tilde{f} = 14$ ($\tilde{p} = 120$)

n	itr($\mathbf{C}_W$)	itr($\mathbf{C}$)	ε_{PCG}	Diag. (s)	Itr (s)	Total (s)
16,000	2710	$> 100,000$	1.90×10^{-3}	553	1844	2397
32,000	4261	-	1.43×10^{-3}	1522	5713	7235
64,000	8801	-	1.00×10^{-4}	5022	23785	28807
128,000	14405	-	7.28×10^{-4}	12587	75937	88524

(c) $\boldsymbol{\theta}_c = (1, 1/6)$, $d = 2$, $f = 14$ ($p = 120$), $\tilde{f} = 14$ ($\tilde{p} = 120$)

The diagonal preconditioner we use is one of the simplest. We plan to extend this approach to more sophisticated preconditioners such as block Symmetric Successive OverRelaxation (SSOR) (see Castrillón-Candás et al. (2013)) in the future.

The residual errors are then propagated to the final estimate $Z(\mathbf{s}_0)$ around the same magnitude. However, as a final experiment in Table 6 we tabulate the relative l_2 error between the multi-level kriging approach and the direct method for data set #2 with exponential covariance function $\exp(-\theta r)$, where $\theta = 5.9915$ and $f = 3$. The PCG tolerance is set to 10^{-5} and $\tilde{f} = 3$. Notice that the error increases with n. This is expected since the unpreconditioned system error will degrade.

Table 6: Tabulation of the kriging estimate relative l_2 error between the multi-level kriging approach and the direct method for Data Set #1 (3D) for 1000 target points. The covariance function is $\exp(-\theta r)$, where $\theta = 5.9915$. The polynomial bias term in the Gaussian model is set to $f = 3$ ($p = 20$). The solver tolerance is set to $\varepsilon_{PCG} = 10^{-5}$ and accuracy parameter is set to $\tilde{p} = 20$ ($\tilde{f} = 3$).

n	l_2 Relative Error
1,000	1.53×10^{-6}
2,000	6.71×10^{-5}
4,000	6.42×10^{-5}
8,000	1.01×10^{-4}
16,000	9.14×10^{-5}
32,000	1.05×10^{-4}

8 Conclusions

In this paper we developed a multi-level restricted Gaussian maximum likelihood method for estimating the covariance function parameters and the computation of the best unbiased predictor. Our approach produces a new set of multi-level contrasts that decouples the covariance parameters from the deterministic components. In addition, the covariance matrix exhibits fast decay independently from the decay rate of the covariance function. Due to the fast decay of the covariance matrix only a small set of coefficients of the covariance matrix are computed with a level-dependent criterion. We showed results of our method for the Matérn covariance with highly irregular placement of the observation locations to good accuracy.

We are currently working on deriving error estimates of the kriging estimate and determinant computation with respect to the number of degrees of freedom n. We are also contemplating extending our multi-level approach to multivariate random fields and cokriging (e.g. Furrer and Genton (2011)).

Our method also applies to non-stationary problems if the covariance function is differentiable to degree $\tilde{f}$. For example, if the covariance function changes smoothly with respect to the location, Lemma 1 still applies and the multi-level covariance matrix decays at the same rate as a stationary one. Now, even if the covariance function is non differentiable everywhere

with respect to the location, Lemma 1 still applies, but at a lower decay rate.

Appendix A: Proofs

Lemma 1:

Since $\phi(r;\boldsymbol{\theta})$ is in $C^{\tilde{f}+1}(\mathbb{R})$, then by Taylor's theorem we have that for every $\mathbf{x} \in B_{\mathbf{a}}$ $\phi(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) = \sum_{|\alpha|\leq \tilde{f}} \frac{D^{\alpha}_{\mathbf{x}}\phi(\mathbf{a},\mathbf{y};\boldsymbol{\theta})}{\alpha!}(\mathbf{x}-\mathbf{a})^{\alpha} + R_{\alpha}(\mathbf{x},\mathbf{y};\boldsymbol{\theta})$, where $(\mathbf{x}-\mathbf{a})^{\alpha} := (x_1-a_1)^{\alpha_1}\cdots(x_d-a_d)^{\alpha_d}$, $\alpha! := \alpha_1!\cdots\alpha_d!$, and $R_{\alpha}(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) := \sum_{|\alpha|=\tilde{f}+1}\frac{(\mathbf{x}-\mathbf{a})^{\alpha}}{\alpha!}D^{\alpha}_{\mathbf{x}}\phi(\mathbf{a}+s(\mathbf{x}-\mathbf{a}),\mathbf{y};\boldsymbol{\theta})$ for some $s \in [0,1]$. Now, recall that $\boldsymbol{\psi}^{i,k}_{\tilde{k}}$ is orthogonal to $\mathcal{P}^{p}(\mathbb{S})$ then $\sum_{h=1}^{n}\boldsymbol{\psi}^{i,k}_{\tilde{k}}[h]\phi(\mathbf{s}_h,\mathbf{y};\boldsymbol{\theta}) = \sum_{h=1}^{n}\boldsymbol{\psi}^{i,k}_{\tilde{k}}[h]$ $R_{\alpha}(\mathbf{s}_h,\mathbf{y};\boldsymbol{\theta})$. Since $\boldsymbol{\psi}^{j,l}_{\tilde{l}}$ is also orthogonal to $\mathcal{P}^{p}(\mathbb{S})$ then by applying Taylor's theorem centered at $\mathbf{b} \in B_{\mathbf{a}}$:

$$
\begin{aligned}
|(\boldsymbol{\psi}^{i,k}_{\tilde{k}})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}^{j,l}_{\tilde{l}}| &= \left|\sum_{h=1}^{n}\sum_{e=1}^{n}\boldsymbol{\psi}^{i,k}_{\tilde{k}}[h]\boldsymbol{\psi}^{j,l}_{\tilde{l}}[e]\phi(\mathbf{s}_h,\mathbf{s}_e;\boldsymbol{\theta})\right| = \left|\sum_{h=1}^{n}\sum_{e=1}^{n}\left(\sum_{|\alpha|=\tilde{f}+1}\frac{(\mathbf{s}_h-\mathbf{a})^{\alpha}}{\alpha!}\right.\right. \\
&\left.\left.\sum_{|\beta|=\tilde{f}+1}\frac{(\mathbf{s}_e-\mathbf{b})^{\beta}}{\beta!}D^{\alpha}_{\mathbf{x}}D^{\beta}_{\mathbf{y}}\phi(a+s(\mathbf{s}_h-\mathbf{a}),b+t(\mathbf{s}_e-\mathbf{b});\boldsymbol{\theta})\boldsymbol{\psi}^{i,k}_{\tilde{k}}[h]\boldsymbol{\psi}^{j,l}_{\tilde{l}}[e]\right)\right| \\
&\leq \sum_{|\alpha|=\tilde{f}+1}\sum_{|\beta|=\tilde{f}+1}\frac{r^{\alpha}_{\mathbf{a}}}{\alpha!}\frac{r^{\beta}_{\mathbf{b}}}{\beta!}|D^{\alpha}_{\mathbf{x}}D^{\beta}_{\mathbf{y}}\phi(\mathbf{a}+s(\mathbf{s}_h-\mathbf{a}),\mathbf{b}+t(\mathbf{s}_e-\mathbf{b});\boldsymbol{\theta})|\left|\sum_{r=1}^{n}\sum_{e=1}^{n}\boldsymbol{\psi}^{i,k}_{\tilde{k}}[r]\boldsymbol{\psi}^{j,l}_{\tilde{l}}[e]\right| \\
&\leq \sum_{|\alpha|=\tilde{f}+1}\sum_{|\beta|=\tilde{f}+1}\frac{r^{\alpha}_{\mathbf{a}}}{\alpha!}\frac{r^{\beta}_{\mathbf{b}}}{\beta!}\sup_{\mathbf{x}\in B_{\mathbf{a}},\mathbf{y}\in B_{\mathbf{b}}}|D^{\alpha}_{\mathbf{x}}D^{\beta}_{\mathbf{y}}\phi(\mathbf{x},\mathbf{y};\boldsymbol{\theta})|,
\end{aligned}
$$

for some $s,t \in [0,1]$. The last inequality follows since from Schwartz' inequality $\sum_{h=1}^{n}|\boldsymbol{\psi}^{i,k}_{\tilde{k}}[h]| \leq \sqrt{\sum_{h=1}^{n}(\boldsymbol{\psi}^{i,k}_{\tilde{k}}[h])^2} = 1$ and $\sum_{e=1}^{n}|\boldsymbol{\psi}^{j,l}_{\tilde{l}}[e]| \leq \sqrt{\sum_{e=1}^{n}(\boldsymbol{\psi}^{j,l}_{\tilde{l}}[e])^2} = 1$. □

Lemma 2: Since the Matérn covariance function $\phi(\mathbf{x},\mathbf{y};\boldsymbol{\theta})$ is positive definite we have that for all $\mathbf{v} \neq \mathbf{0}$:

$$
\sum_{i,j=1}^{n} v_i v_j \mathbf{C}^{i,j}(\boldsymbol{\theta}) = \sum_{i,j=1}^{n} v_i v_j \phi(\mathbf{x}_i,\mathbf{y}_j;\boldsymbol{\theta}) > 0,
$$

where $\mathbf{C}^{i,j}(\boldsymbol{\theta})$ is the (i,j) entry of the matrix $\mathbf{C}(\boldsymbol{\theta})$. The diagonal terms of $\mathbf{C}_W$ are of the form $(\boldsymbol{\psi}^{i,k}_{\tilde{k}})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}^{i,k}_{\tilde{k}}$. This implies that

$$(\boldsymbol{\psi}_{\tilde{k}}^{i,k})^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})\boldsymbol{\psi}_{\tilde{k}}^{i,k} > 0.$$

Thus, $\mathbf{D}_W$ will always be positive definite. □

Appendix B: Notation

Index for when the following are first defined, mentioned or reformulated.

$d \in \mathbb{N}$	Dimension of problem (p1)	$n \in \mathbb{N}$	Number of observations (p1)
$t \in \mathbb{N}$	Maximum MB level (p7)	$p \in \mathbb{N}$	Number of columns of $\mathbf{M}$ (p1)
$f \in \mathbb{N}$	Polynomial degree (p6)	$\tilde{p} \in \mathbb{N}$	Accuracy parameter of MB (p10)
$\tilde{f} \in \mathbb{N}$	Degree of multilevel basis. (p4,6,10)	$w \in \mathbb{N}$	Dimension of $\boldsymbol{\theta}$ (p1)
$\mathbf{M} \in \mathbb{R}^{n\times p}$	Design matrix (p1)	$\phi(\cdot)$	Covariance function (p6)
$\mathrm{S} := \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$	Locations of observations (p1)	$\boldsymbol{\theta} := (\nu, \rho) \in \mathbb{R}^2$	Param. of matern kernel (p1)
$\mathbf{C}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$	Covariance matrix (p1)	$\mathbf{Z} \in \mathbb{R}^n$	Observation values (p1)
$\mathbf{s}_0 \in \mathbb{R}^d$	Target points set (p1)	$\mathcal{P}^p(\mathrm{S})$	Span of the columns of $\mathbf{M}$ (p1)
$\boldsymbol{\beta} \in \mathbb{R}^p$	Vector of unknowns from deterministic model (p1)	$\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$, $\hat{\boldsymbol{\theta}} \in \mathbb{R}^2$	Estimates
		$\hat{\mathbf{Z}}(\mathbf{s}_0)$	kriging estimate at $\mathbf{s}_0$ (p2)
$l(\boldsymbol{\beta}, \boldsymbol{\theta}) \in \mathbb{R}$	Log-likelihood function (p1)	B_k^q	Box at level q and index k (p6)
	in B_k^q (p8)		in B_k^q (p7,9)
	at level q (p9)		at level q (p9)
$\mathbf{L}$	$\mathbb{R}^n \to \mathcal{P}^p(\mathrm{S})$ (p3,9)	$\mathbf{W}$	$\mathbb{R}^n \to (\mathcal{P}^p(\mathrm{S}))^{\perp}$ (p3,9)
$\mathbf{P}^{\mathrm{T}}$	$\begin{bmatrix} \mathbf{W}^{\mathrm{T}} & \mathbf{L}^{\mathrm{T}} \end{bmatrix}$ (p10)	$\tilde{\mathbf{Z}}_W^i$	is equal to $\tilde{\mathbf{W}}_i\mathbf{Z}$ (p15)
$\mathbf{C}_W^i$	is equal to $\mathbb{E}[\tilde{\mathbf{Z}}_W^i(\tilde{\mathbf{Z}}_W^i)^{\mathrm{T}}]$ (p15)	$\tilde{\ell}_W^i(\boldsymbol{\theta})$	Multilevel log-likelihood (p15)
$\mathcal{Q}_f^d$	Set of polynomial monomials of order f and dimension d (p6)	$\tau \in \mathbb{N}_0$	Level dependent criterion constant (p13)

Appendix C

Using the approach developed in this paper we can compute the MSE at the target point $\mathbf{s}_0$.

Now, since $\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}$ we have that $\mathbf{M}_f^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{M}_f = \mathbf{M}_f^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{M}_f = \mathbf{M}_f^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{K}_W(\boldsymbol{\theta})^{-1}\mathbf{P}\mathbf{M}_f$, $\mathbf{K}_W(\boldsymbol{\theta}) := \mathbf{P}\mathbf{K}(\boldsymbol{\theta})\mathbf{P}^{\mathrm{T}}$, $\mathbf{K}_W(\boldsymbol{\theta}) := \begin{bmatrix} \mathbf{C}_W(\boldsymbol{\theta}) & \mathbf{a}_W \\ \mathbf{a}_W^{\mathrm{T}} & \mathbf{S}_W(\boldsymbol{\theta}) \end{bmatrix}$ where $\mathbf{S}_W(\boldsymbol{\theta}) := \mathbf{L}\mathbf{C}(\boldsymbol{\theta})\mathbf{L}^{\mathrm{T}} \in \mathbb{R}^{p\times p}$ and $\mathbf{a}_W(\boldsymbol{\theta}) := \mathbf{W}\mathbf{C}(\boldsymbol{\theta})\mathbf{L}^{\mathrm{T}} \in \mathbb{R}^{(n-p)\times p}$. Let $\tilde{\mathbf{S}}_W(\boldsymbol{\theta}) := (\mathbf{S}_W(\boldsymbol{\theta}) - \mathbf{a}_W^{\mathrm{T}}\mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{a}_W)^{-1}$, then

$$\mathbf{K}_W(\boldsymbol{\theta})^{-1} := \begin{bmatrix} \mathbf{C}_W(\boldsymbol{\theta})^{-1} + \mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{a}_W\tilde{\mathbf{S}}_W(\boldsymbol{\theta})\mathbf{a}_W^{\mathrm{T}}\mathbf{C}_W(\boldsymbol{\theta})^{-1} & -\mathbf{C}_W(\boldsymbol{\theta})^{-1}\mathbf{a}_W\tilde{\mathbf{S}}_W(\boldsymbol{\theta}) \\ -\tilde{\mathbf{S}}_W(\boldsymbol{\theta})\mathbf{a}_W^{\mathrm{T}}\mathbf{C}_W(\boldsymbol{\theta})^{-1} & \tilde{\mathbf{S}}_W(\boldsymbol{\theta}) \end{bmatrix}.$$

Given that $\mathbf{W}\mathbf{M}_f = \mathbf{0}$ then $\mathbf{M}_f^{\mathrm{T}}\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{M}_f = (\mathbf{L}\mathbf{M}_f)^T\tilde{\mathbf{S}}_W(\boldsymbol{\theta})(\mathbf{L}\mathbf{M}_f)$. Following a similar argument we have that equation (14) becomes

$$1 + \tilde{\mathbf{u}}^{\mathrm{T}}\mathbf{M}_f^{\mathrm{T}}(\mathbf{L}\mathbf{M}_f)^{\mathrm{T}}\tilde{\mathbf{S}}_W(\boldsymbol{\theta})(\mathbf{L}\mathbf{M}_f)\tilde{\mathbf{u}} - \mathbf{c}_W^{\mathrm{T}}\mathbf{K}_W^{-1}(\boldsymbol{\theta})\mathbf{c}_W, \tag{16}$$

where $\mathbf{c}_W := \mathbf{P}\mathbf{c}$. By using matrix-vector products with the PCG method $\tilde{\mathbf{S}}_W(\boldsymbol{\theta})$ can be computed in $\mathcal{O}((k(t+2)+1)np + p^d)$. Thus the term $(\mathbf{L}\mathbf{M}_f)^{\mathrm{T}}\tilde{\mathbf{S}}_W(\boldsymbol{\theta})(\mathbf{L}\mathbf{M}_f)$ can be computed in $\mathcal{O}((k(t+2)+1)np + p^d + (t+2)pn)$. Now, by also using matrix-vector products the term $\mathbf{K}_W^{-1}(\boldsymbol{\theta})\mathbf{c}_W$ can be computed in $\mathcal{O}(kn(t+2) + p^d)$. Thus the total cost for computing equation (16) is $\mathcal{O}((k(t+2)+1)np + p^d + (t+2)pn)$.

Acknowledgements: We appreciate the help and advice from Jun Li and Lisandro Dalcin in getting the C++ code working properly, and to Stefano Castruccio for giving us feedback on our manuscript.

References

Anitescu, M., Chen, J., and Wang, L. (2012), "A Matrix-Free Approach for Solving the Parametric Gaussian Process Maximum Likelihood Problem," *SIAM Journal on Scientific Computing*, 34, 240–262.

Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H. (2013a), "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory.

Balay, S., Brown, J., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H. (2013b), "PETSc Web Page," Http://www.mcs.anl.gov/petsc.

Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997), "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," in *Modern Software Tools in Scientific Computing*, eds. Arge, E., Bruaset, A. M., and Langtangen, H. P., Birkhäuser Press, pp. 163–202.

Beatson, R. and Greengard, L. (1997), "A Short Course on Fast Multipole Methods," in *Wavelets, Multilevel Methods and Elliptic PDEs*, Oxford University Press, pp. 1–37.

Castrillón-Candás, J., Li, J., and Eijkhout, V. (2013), "A Discrete Adapted Hierarchical Basis Solver for Radial Basis Function Interpolation," *BIT Numerical Mathematics*, 53, 57–86.

Chen, Y., Davis, T. A., Hager, W. W., and Rajamanickam, S. (2008), "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate," *ACM Trans. Math. Softw.*, 35, 22:1–22:14.

Davis, T. and Hager, W. (1999), "Modifying a Sparse Cholesky Factorization," *SIAM Journal on Matrix Analysis and Applications*, 20, 606–627.

— (2001), "Multiple-Rank Modifications of a Sparse Cholesky Factorization," *SIAM Journal on Matrix Analysis and Applications*, 22, 997–1013.

— (2005), "Row Modifications of a Sparse Cholesky Factorization," *SIAM Journal on Matrix Analysis and Applications*, 26, 621–639.

Davis, T. A. and Hager, W. W. (2009), "Dynamic Supernodes in Sparse Cholesky Update/Downdate and Triangular Solves," *ACM Trans. Math. Softw.*, 35, 27:1–27:23.

Elden, L., Wittmeyer-Koch, L., and Nielsen, H. (2004), *Introduction to Numerical Computation - analysis and Matlab illustrations*, Studentlitteratur.

Furrer, R. and Genton, M. G. (2011), "Aggregation-Cokriging for Highly-Multivariate Spatial Data," *Biometrika*, 98, 615–631.

Furrer, R., Genton, M. G., and Nychka, D. (2006), "Covariance Tapering for Interpolation of Large Spatial Datasets," *Journal of Computational and Graphical Statistics*, 15, 502–523.

George, A. (1973), "Nested Dissection of a Regular Finite Element Mesh," *SIAM Journal on Numerical Analysis*, 10, 345–363.

Gilbert, J. R. and Tarjan, R. E. (1987), "The Analysis of a Nested Dissection Algorithm," *Numerische Mathematik*, 50, 377–404.

Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008), "Covariance Tapering for Likelihood-Based Estimation in Large Spatial Datasets," *Journal of the American Statistical Association*, 103, 1545–1555.

Stein, M. L., Chen, J., and Anitescu, M. (2012), "Difference Filter Preconditioning for Large Covariance Matrices," *SIAM Journal on Matrix Analysis and Applications*, 33, 52–72.

— (2013), "Stochastic Approximation of Score Functions for Gaussian Processes," *Annals of Applied Statistics*, 7, 1162–1191.

Stein, M. L., Chi, Z., and Welty, L. J. (2004), "Approximating Likelihoods for Large Spatial Data Sets," *Journal of the Royal Statistical Society, Series B*, 66, 275–296.

Sun, Y., Li, B., and Genton, M. G. (2012), "Geostatistics for Large Datasets," in *Space-Time Processes and Challenges Related to Environmental Problems*, eds. Porcu, M., Montero, J. M., and Schlather, M., Springer, pp. 55–77.

Sun, Y. and Stein, M. L. (2015), "Statistically and Computationally Efficient Estimating Equations for Large Spatial Datasets," *Journal of Computational and Graphical Statistics*, in press.

Ying, L., Biros, G., and Zorin, D. (2004), "A Kernel-Independent Adaptive Fast Multipole Method in Two and Three Dimensions," *Journal of Computational Physics*, 196, 591–626.