MULTI-RESOLUTION APPROXIMATIONS OF GAUSSIAN PROCESSES FOR LARGE SPATIAL DATASETS

A Dissertation

by

WENLONG GONG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Matthias Katzfuss
Committee Members,	Raymond Carroll
	Huiyan Sang
	Anita Rapp
Head of Department,	Valen Johnson

December 2018

Major Subject: Statistics

ABSTRACT

Recent advances in remote-sensing techniques enabled accurate location geocoding and encouraged the collection of big spatial datasets over large domains. Data obtained in these settings are usually multivariate, with several spatial variables observed at each location. Statistical modeling for such spatial data is of ever-increasing importance in a variety of fields, including agriculture, climate science, astronomy, atmospheric science. Gaussian processes are popular and flexible models for such data, but they are computationally infeasible for large datasets.

This dissertation is focused on spatial inference and prediction for big spatial data, and in particular on the computational feasibility of the statistical methodologies. It includes a general introduction to spatial statistics including Gaussian processes, spatial prediction as well as multivariate spatial data modeling. We also introduce Gaussian-process approximations that use basis functions at multiple resolutions to achieve fast inference and that can (approximately) represent any spatial covariance structure. Finally, we extend the multi-resolution approximation from univariate to multivariate spatial data, where the computation is even more expensive, by introducing latent dimensions into covariance modeling. The last part concludes the dissertation and discusses the future work.

DEDICATION

To my parents for all their love and providing me with the best education possible. I wouldn't have been able to get here without their support.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Matthias Katzfuss, for his support and mentorship, as well as his patience and the time he spend on me for the passed three years. Without his guidance and persistent help this dissertation would not have been possible.

Also, I would like to thank: my committee member, Dr. Huiyan Sang for her help on understanding Full Scale Approximation; Dr. Raymond Carroll for his valuable advices on PhD study and conducting research in general; Dr. Anita Rapp for her comments on my dissertation proposal; Dr. Tim Travis for his suggestions on sparse matrix computation. I'm also very thankful to Dr. Shiyuan He, Dr. Kejun He for their help on some of the theoretical proofs; and anonymous referees for their helpful reviews of manuscript version of Chapter 3. Beside that, I would like to thank the Department of Statistics for the student assistantship suppoort over two years. Special thanks goes to associate department head, Dr. Michael Longnecker, who is always willing to help and give his best suggestions during all my years in the department.

Last, I would like to express my special gratitude to my family and friends for their unconditional and endless support. They have been always with me through ups and downs during my PhD study.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Matthias Katzfuss, Huiyan Sang, and Raymond Carroll of the Department of Statistics, and Professor Anita Rapp of the Department of Atmospheric Sciences.

The analyses depicted in Chapter 3 were conducted in part by Professor Matthias Katzfuss and the manuscript was submitted to arXiv in 2017. All other work conducted for the dissertation was completed by the student independently.

Funding Sources

The research was partially supported by NSF DMS-1521676 and NASA ESTO AIST-14 and graduate study was partially supported by Texas Public Education Grant.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Spatial data can be defined as location-referenced measurements taken over a spatial domain, arising in many disciplines like agriculture, geoscience, astronomy, meteorology and ecology. Depending on the measuring procedure and domain properties, spatial data could be generated from very different processes: continuous and discrete spatial processes, or point processes with random measurement locations. Spatial statistics deals with data that are spatially referenced and follows the principle that data points close in space are typically more similar than those who are far apart. For an overview of spatial statistics, see Cressie (1993).

In the recent times, the ubiquity of automated remote-sensing instruments on satellites and aircraft have led to an explosion in the amount of environmental data being collected in all fields of science. For example, the NASA Earth Observing System generates everyday terabytes of data about the land surface, the biosphere, the atmosphere, and the oceans. This results in an increasing need for analyzing big datasets with large numbers of variables and massive amounts of observations. Analyzing these datasets can give us new insight on a variety of problems, such as environmental pollution, atmospheric weather forecasting and climate change. With Gaussian processes, the spatial statistics field has a rich toolkit for data inference, including parameters estimation, predicting the unobserved spatial field, and the associated uncertainty quantification (Cressie and Wikle, 2011).

However, large data sets have posed tremendous challenges to spatial data modeling because of the notorious curse of dimensionality. In particular, for data of size n with one response variable, spatial modeling and prediction both involve inversion of an $n \times n$ covariance matrix. Traditional statistical methods require $\mathcal{O}(n^2)$ memory complexity and $\mathcal{O}(n^3)$ time complexity, which is computationally prohibative for very large n. For multivariate spatial data, say p-variate with n observations, the cross-covariance would be of size $np \times np$, and statistical inference would be even more computationally demanding. Therefore, scalable statistical methods are needed to extract information from these big spatial datasets, especially for multivariate data.

This dissertation is focused on spatial inference and prediction for big spatial data, and the particular concern here is with computational feasibility of the statistical methodology. Chapter 2 reviews some basics of spatial statistics. Chapter 3 introduces the multi-resolution approximation method for big spatial data with two examples and an application to satellite data. Chapter 4 extends the multi-resolution approximation from univariate to multivariate spatial data. Chapter 5 concludes the dissertation.

2. BACKGROUND

2.1 Gaussian process models for spatial data

A Gaussian process is a collection of random variables such that every finite subset of these random variables has a multivariate normal distribution. Gaussian process models have been widely used in spatial statistics, because the normal distribution has many well-known and attractive properties. Gaussian processes are flexible and allow for natural uncertainty quantification. In spatial statistics, the random variables that make up the Gaussian process are indexed by spatial locations.

Spatial data can usually be modeled as a Gaussian process as follows. Let $y(\cdot) \sim GP(\mu, C)$ be a Gaussian process with mean μ, and a covariance function C, which has to be positive definite. Then for any finite set of locations $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\} \subset \mathcal{D} \in \mathbf{R}^d, d \in \mathbb{Z}^+$, we have

$$\mathbf{y}(\mathcal{S}) \coloneqq (y(\mathbf{s}_1), y(\mathbf{s}_2), \ldots, y(\mathbf{s}_n))' \sim \mathcal{N}_n(\boldsymbol{\mu}, \mathbf{C}),$$

where

$$\begin{aligned}\boldsymbol{\mu} =& \boldsymbol{\mu}(\mathcal{S}) = (\mu(\mathbf{s}_1), \mu(\mathbf{s}_2), \ldots, \mu(\mathbf{s}_n))';\\ \mathbf{C} =& \mathbf{C}(\mathcal{S}, \mathcal{S}) = (C(\mathbf{s}_i, \mathbf{s}_j))_{i,j=1,\ldots,n}.\end{aligned}$$

The covariance function C describes the dependence between measurements across locations and quantifies the variability of the Gaussian process. It usually involves some unknown parameters θ.

Geospatial settings typically assume that at each $\mathbf{s} \in \mathcal{S}$, that the observed data, z, composed of the underlying spatial process y, capturing the spatial association, and an independent process ϵ, which is often called the nugget:

$$z(\mathbf{s}) = y(\mathbf{s}) + \epsilon(\mathbf{s}). \tag{2.1}$$

If we assume y is a Gaussian process with covariance function C, and $\epsilon(\mathbf{s}) \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, independent of $y(\cdot)$, So the covariance of $\mathbf{z} = (z(\mathbf{s}_1), z(\mathbf{s}_2), \ldots, z(\mathbf{s}_n))$ is $\boldsymbol{\Sigma} = \mathbf{C} + \sigma^2\mathbf{I}$.

2.1.1 Stationarity and isotropy

While the general definition of a Gaussian process provides a very broad and flexible class of models, applications often require some additional assumptions.Two properties are commonly imposed on $y(\cdot)$: stationarity and isotropy.

A stochastic process $y(\cdot)$ is strictly stationary (or strongly stationary) if all finite-dimensional distributions are transformation or shift invariant, i.e.,

$$P(y(\mathbf{s}_1) \leq y_1, \ldots, y(\mathbf{s}_n) \leq y_n) = P(y(\mathbf{s}_1 + \mathbf{h}) \leq y_1, \ldots, y(\mathbf{s}_n + \mathbf{h}) \leq y_n),$$

for any vector $\mathbf{h} \in \mathbb{R}^d$ and any choices of spatial locations $\mathbf{s} \in \mathcal{D}$.

A stochastic process $y(\cdot)$ is weakly stationary or second-order stationary if the mean is spatially constant, which means that,

$$E(y(\mathbf{s})) = \mu(\mathbf{s}) = \mu(\mathbf{s} + \mathbf{h})$$

and the covariance is a function of only the vector $\mathbf{h}$,

$$\text{cov}(y(\mathbf{s}), y(\mathbf{s} + \mathbf{h})) = \mathbf{C}(\mathbf{s}, \mathbf{s} + \mathbf{h}) = \mathbf{C}(\mathbf{h}),$$

for any $\mathbf{s} \in \mathcal{D}$ and $\mathbf{h} \in \mathbb{R}^d$.

Throughout the rest of dissertation we will use the term stationary to mean weakly stationary. A special subclass of stationary processes are isotropic processes.

A stochastic process $y(\cdot)$ is isotropic if it is weakly stationary and the covariance is a function of only distance:

$$\text{cov}(y(\mathbf{s}), y(\mathbf{s} + \mathbf{h})) = \mathbf{C}(\mathbf{s}, \mathbf{s} + \mathbf{h}) = \mathbf{C}(\|\mathbf{h}\|)$$

for any $\mathbf{s} \in \mathcal{D}, \mathbf{h} \in \mathbb{R}^d$. Isotropy means that the process is rotation invariant, and note that $\mu(\cdot)$ is spatially constant for isotropic processes.

Suppose we have a single spatial variable, and our univariate data generated by Gaussian pro-

cess $y(\cdot)$, which we assume to be second-order stationary with mean zero. According to the definition above, the covariance function is isotropic if $\mathbf{C}(\mathbf{h}_1) = \mathbf{C}(\mathbf{h}_2)$ whenever $\|\mathbf{h}_1\| = \|\mathbf{h}_2\|$, where$\|\cdot\|$ is the Euclidean norm. A class of isotropic covariance function that has received great attention is the Matérn family, which has the form of

$$\mathcal{M}(r|\sigma^2, \lambda, \nu) = \sigma^2(\frac{r}{\lambda})^\nu \mathcal{K}_\nu(\frac{r}{\lambda}), \tag{2.2}$$

where σ^2 is the sill parameter, $\lambda > 0$ is the spatial range and $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function of order $\nu > 0$ with ν being called the smoothness parameter, which captures the smoothness of the random field, with larger values of ν corresponding to smoother fields. The range parameter λ measures how fast the correlation of the random field decays with increasing distance r, with larger λ indicating a faster decay(while keeping ν fixed). The commonly used smoothness levels are 0.5, 1.5 and 2.5, because then $\mathcal{K}_\nu(\cdot)$ becomes a polynomial. Moreover, for $\nu = 0.5$, Matèrn covariance function with smoothness 0.5 is equivalent to the exponential covariance function

$$\mathbf{C}(r) = \sigma^2 \exp(-\frac{r}{\lambda}).$$

2.2 Spatial prediction for Gaussian process

A major goal in spatial statistical analysis is to make inference on $\mathbf{y}(\mathcal{S}^P)$, which is the true process $\mathbf{y}(\cdot)$ at n_P prediction locations $\mathcal{S}^P \coloneqq \{\mathbf{s}_1^P, \ldots, \mathbf{s}_{n^P}^P\}$. This is also referred to as Kriging. If $\mathbf{y}(\cdot)$ is observed directly and without error, let $\mathbf{y} = \mathbf{y}(\mathcal{S})$, $\mathbf{y}^P = \mathbf{y}(\mathcal{S}^P)$, and suppress the dependence on covariance parameter θ, we have

$$\begin{pmatrix} \mathbf{y}^P \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}_{n_P+n}\left(\begin{pmatrix} \boldsymbol{\mu}(\mathcal{S}^P) \\ \boldsymbol{\mu}(\mathcal{S}) \end{pmatrix}, \begin{pmatrix} \mathbf{C}(\mathcal{S}^P, \mathcal{S}^P) & \mathbf{C}(\mathcal{S}, \mathcal{S}^P) \\ \mathbf{C}(\mathcal{S}^P, \mathcal{S}) & \mathbf{C}(\mathcal{S}, \mathcal{S}) \end{pmatrix}\right). \tag{2.3}$$

For multivariate normal distribution, according to its well-known properties, we have

$$\mathbf{y}^P|\mathbf{y} \sim \mathcal{N}_{n_P}(\boldsymbol{\mu}_{\mathbf{y}^P|\mathbf{y}}, \Sigma_{\mathbf{y}^P|\mathbf{y}}), \tag{2.4}$$

where

$$\mu_{\mathbf{y}^P|\mathbf{y}} = \boldsymbol{\mu}(\mathcal{S}^P) + \mathbf{C}(\mathcal{S}^P, \mathcal{S})(\mathbf{C}(\mathcal{S}, \mathcal{S}))^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathcal{S}))$$
$$\Sigma_{\mathbf{y}^P|\mathbf{y}} = \mathbf{C}(\mathcal{S}^P, \mathcal{S}^P) - \mathbf{C}(\mathcal{S}^P, \mathcal{S})(\mathbf{C}(\mathcal{S}, \mathcal{S}))^{-1}\mathbf{C}(\mathcal{S}, \mathcal{S}^P).$$

The $\mu_{\mathbf{y}^P|\mathbf{y}}$ is the best linear unbiased predictor of $\mathbf{y}^P$ based on $\mathbf{y}$ (even without the Gaussian assumption).

However, most of the time, we observe data $\mathbf{z}(\cdot)$ with measurement error, instead of the underlying spatial process $\mathbf{y}(\cdot)$ directly. Thus it is more realistic to assume additive Gaussian noise in $\mathbf{z}(\cdot)$ as in Equation 2.1. Let $\mathbf{y}^P = \mathbf{y}(\mathcal{S}^P)$, $\mathbf{z}(\mathcal{S}) = \mathbf{z}$, then the spatial prediction for Gaussian data model becomes

$$\mathbf{y}^P|\mathbf{z} \sim \mathcal{N}_{n_P}(\boldsymbol{\mu}_{\mathbf{y}^P|\mathbf{z}}, \Sigma_{\mathbf{y}^P|\mathbf{z}}),$$

where

$$\mu_{\mathbf{y}^P|\mathbf{z}} = \boldsymbol{\mu}(\mathcal{S}^P) + \mathbf{C}(\mathcal{S}^P, \mathcal{S})\Sigma^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathcal{S}))$$
$$\Sigma_{\mathbf{y}^P|\mathbf{z}} = \mathbf{C}(\mathcal{S}^P, \mathcal{S}^P) - C(\mathcal{S}^P, \mathcal{S})\Sigma^{-1}\mathbf{C}(\mathcal{S}, \mathcal{S}^P).$$

2.3 Multivariate spatial data modeling

Multivariate spatial data is consisting of measurements of several spatially correlated variables, which are recorded at varying spatial locations. In this case, we need to model both dependence between variables at a particular location, and correlation between measurements across locations.

For a p-variate spatial process, let $\mathbf{Y}(\mathbf{s}) = (y_1(\mathbf{s}), \ldots, y_p(\mathbf{s}))$, at each location, $\mathbf{s} \in \mathcal{S}$, there are

p constituent components. The matrix cross-covariance function of $\mathbf{Y}(\mathbf{s})$ is

$$\mathbf{C}(\cdot,\cdot) = \begin{pmatrix} \mathbf{C}_{11}(\cdot,\cdot) & \dots & \mathbf{C}_{1p}(\cdot,\cdot) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{p1}(\cdot,\cdot) & \dots & \mathbf{C}_{pp}(\cdot,\cdot) \end{pmatrix}, \tag{2.5}$$

a mapping $\mathbf{C} : \mathcal{D} \times \mathcal{D} \to M_{p\times p}$, where $M_{p\times p}$ is the set of $p \times p$ matrices, such that the $np \times np$ covariance matrix

$$\mathbf{C}(\mathcal{S},\mathcal{S}) = \begin{pmatrix} \mathbf{C}(\mathbf{s}_1,\mathbf{s}_1) & \mathbf{C}(\mathbf{s}_1,\mathbf{s}_2) & \dots & \mathbf{C}(\mathbf{s}_1,\mathbf{s}_n) \\ \mathbf{C}(\mathbf{s}_2,\mathbf{s}_1) & \mathbf{C}(\mathbf{s}_2,\mathbf{s}_2) & \dots & \mathbf{C}(\mathbf{s}_2,\mathbf{s}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}(\mathbf{s}_n,\mathbf{s}_1) & \mathbf{C}(\mathbf{s}_n,\mathbf{s}_2) & \dots & \mathbf{C}(\mathbf{s}_n,\mathbf{s}_n) \end{pmatrix} \tag{2.6}$$

is (symmetric) positive definite, though covariance function $\mathbf{C}$ itself is not required to be symmetric. Each block in Equation 2.6 is of size $p \times p$, quantifying the (cross) covariance of p variables measured at certain pairs of locations.

Valid cross-covariance functions require that for any number or any choice of locations, the resulting covariance matrix has to be nonnegative definite. Thus constructing a valid cross-covariance function is even more difficult than in the univariate case. Various constructions are possible, including separable cross-covariance functions (Mardia and Goodall, 1993), linear model of coregionalization (LMC) (e.g., Wackernagel, 1995; Banerjee et al., 2004; Gelfand et al., 2004).

Separable cross-covariance functions assume $\mathbf{C}_{ij}(\mathbf{s}_1,\mathbf{s}_2) = \rho(\mathbf{s}_1,\mathbf{s}_2)R_{ij}$, where $\rho(\cdot,\cdot)$ has to be a valid correlation function and $R_{ij} = Cov(y_i, y_j)$ is the nonspatial covariance between variables i and j. Separable cross-covariance were sometimes called intrinsic co-regionalizations (Helterbrand and Cressie, 1994). This construction is simple, easily interpreted, but not flexible enough to model complex interactions between processes.

Linear combinations of independent processes provide a rich class of cross-covariances for multivariate data. Such models are the so-called linear models of coregionalization (LMC), which

can be exploited for valid dependence structures. The linear model of coregionalization defines $\mathbf{Y}(\mathbf{s}) = \mathbf{A}(\mathbf{s})\mathbf{x}(\mathbf{s}), \mathbf{s} \in \mathcal{D}$, where $\mathbf{A}$ is a deterministic, potentially spatially varying matrix. Then by construction, $\mathbf{y}(\cdot)$ is a valid p-variate Gaussian process with cross-covariance function. LMC is a rather general construction method and it is usually too restrictive to assume that $\mathbf{A}$ is constant. A spatially varying LMC, however, has many parameters, and thus introduces extra difficulties in the computation for necessary estimation and inferences.

2.3.1 Cross-covariance functions based on latent dimensions

Another approach to construct p-variate cross-covariance functions is based on univariate spatial covariance on an extended space (Apanasovich and Genton, 2010; Genton and Kleiber, 2015). The idea is to introduce additional latent dimensions that represent the locations of variables to be studied. Specifically, each component y_k is represented as a vector $\boldsymbol{\xi}_k \in \mathbb{R}^q$, i.e., $\boldsymbol{\xi}_k = (\xi_{k,1}, \ldots, \xi_{k,q})^T$, $k = 1, \ldots, p$ representing the processes on latent dimensions. Let $\boldsymbol{\xi}_{k(i)}$ denote the location in latent dimension of the observation at $\mathbf{s}_i$. Under this setting, the cross-covariance function between y_k and y_l becomes:

$$C_{kl}(\mathbf{s}_i, \mathbf{s}_j) = C((\mathbf{s}_i, \boldsymbol{\xi}_k), (\mathbf{s}_j, \boldsymbol{\xi}_l)), \qquad \mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^d, \qquad \boldsymbol{\xi}_k, \boldsymbol{\xi}_l \in \mathbb{R}^q. \tag{2.7}$$

Consequently, the resulting covariance matrix is guaranteed to be nonnegative definite if C is a valid covariance function. If C is stationary or isotropic, then so is the cross-covariance function. This construction is very flexible and capable of modeling complex multivariate data. As discussed in Section 2.1.1, the covariance function is stationary in both space domain and latent domain if $\mathbf{C}_{kl}(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{C}_{kl}(\|\mathbf{s}_i - \mathbf{s}_j\|, \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_l\|)$.

Since cross-covariance functions for multivariate spatial data must incorporate the correlation among variables in addition to the spatial dependence, and multivariate models are more computationally intensive than the univariate case, especially when both n and p are large. In Chapter 4, we utilize the latent dimension method and apply the multi-resolution approximation to facilitate the computation in the modeling and analysis of very large multivariate spatial data sets.

3. MULTI-RESOLUTION APPROXIMATIONS OF GAUSSIAN PROCESSES FOR LARGE SPATIAL DATASETS

3.1 Introduction

Gaussian processes (GPs) are highly popular models for spatial data, time series, and functions. They are flexible and allow natural uncertainty quantification, but their computational complexity is cubic in the data size. This prohibits GPs from being used directly for the analysis of many modern datasets consisting of a large number of observations, such as satellite remote-sensing data.

Consequently, many approximations or assumptions have been proposed that allow the application of GPs to large datasets. Some of these approaches are most appropriate for capturing fine-scale structure (e.g., Furrer et al., 2006; Kaufman et al., 2008), while others are more capable at capturing large-scale structure (e.g., Higdon, 1998; Mardia et al., 1998; Wikle and Cressie, 1999; Cressie and Johannesson, 2008; Katzfuss and Cressie, 2009, 2011, 2012). Lindgren et al. (2011) proposed an approximation based on viewing a GP with Matérn covariance as the solution to the corresponding stochastic partial differential equation, but this approach is only applicable to covariance functions of Matérn type. Vecchia's method and its extensions (e.g., Vecchia, 1988; Stein et al., 2004; Datta et al., 2016; Katzfuss and Guinness, 2017) are discontinuous and assume the so-called screening effect to hold, meaning that any given observation is conditionally independent from other observations given a small subset of (typically, nearby) observations.

We propose the multi-resolution approximation (M-RA) method, which allows capturing spatial structure at all scales. The M-RA is based on an orthogonal decomposition of the GP of interest into processes at multiple resolutions by iteratively applying the predictive process (Quiñonero-Candela and Rasmussen, 2005; Banerjee et al., 2008). The process at each resolution has an equivalent representation as a linear combination of basis functions. For increasing resolution, the number of functions increases while their scale decreases. Unlike other multi-resolution models

or wavelets (e.g. Chui, 1992; Johannesson et al., 2007; Cressie and Johannesson, 2008; Nychka et al., 2015), this M-RA automatically specifies the prior distributions of their weights as well as the basis functions with given covariance function of interest, without imposing any conditions.

To achieve computational feasibility within the M-RA framework, an approximation of the "remainder process" at each resolution using so-called modulating functions is necessary. We consider two special cases: For the M-RA taper, the modulating functions are taken to be tapering functions (i.e., compactly supported correlation functions). For increasing resolution, the remainder process is approximated with increasingly restrictive tapering functions, leading to increasingly sparse matrices. In contrast, the M-RA-block iteratively splits each region at each resolution into a set of subregions, with the remainder process assumed to be independent between these subregions. This can lead to discontinuities at the region boundaries. A special case of the M-RA-block (Katzfuss, 2017) performed very well in a recent comparison of different methods for large spatial data (Heaton et al., 2017). A further special case with only one resolution of the M-RA is given by the full-scale approximation (Snelson and Ghahramani, 2007; Sang et al., 2011; Sang and Huang, 2012).

The M-RA is suitable for inference based on large numbers of observations from a GP, which may be irregularly spaced. We will describe inference procedures that rely on operations on sparse matrices for computational feasibility. The M-RA-block can deal with truly massive datasets, as it amenable to parallel computations on modern distributed computing systems. It can be viewed as a Vecchia-type approximation (Katzfuss and Guinness, 2017), and the approximated covariance matrix is a so-called hierarchical off-diagonal low-rank matrix (e.g., Ambikasaran et al., 2016). The M-RA-taper leads to more general sparse matrices, and thus requires more careful algorithms to fully exploit the sparsity structure, but it has the advantage of not introducing artificial discontinuities.

This chapter is organized as follows. In Section 3.2, we first describe an exact orthogonal multiresolution decomposition of a GP, which then leads to the M-RA framework and the two special cases described above by applying the appropriate modulating functions. We also study their

theoretical properties. In Section 3.3, we discuss the algorithms necessary for statistical inference using the M-RA and provide details of the computational complexity. Numerical comparisons on simulated and real data are given in Sections 3.4 and 3.5, respectively. We conclude in Section 3.6. All proofs can be found in Appendix A.1. Additional simulation results can be found in the Appendix A.2.

3.2 Multi-resolution approximations

3.2.1 The true Gaussian process

Let $\{y_0(\mathbf{s}) : \mathbf{s} \in \mathcal{D}\}$ be the underlying spatial field on a continuous (non-gridded) domain $\mathcal{D} \subset \mathbb{R}^d$, $d \in \mathbb{N}^+$. And we assume that $y_0(\cdot) \sim GP(0, C_0)$ being a Gaussian process with mean 0 and a known covariance function C_0 which is positive-definite. For most spatial fields in reality, $y_0(\cdot)$ may not have mean 0, but we can estimate and subtract the mean easily. With observed $y_0(\cdot)$ at spatial locations $\mathcal{S}$ of size n, the main goal is to make parameters inference for $\boldsymbol{\theta}$ and predict $y_0(\cdot)$ at a set of locations $\mathcal{S}^P$. These procedures has $\mathcal{O}(n^2)$ memory complexity and $\mathcal{O}(n^3)$ time complexity, which is computationally prohibitive for $n \gg 10^4$.

3.2.2 Preliminaries

A multi-resolution approximation (M-RA) with M resolutions requires two main "ingredients": knots and modulating functions. The multi-resolutional set of knots, $\mathcal{Q} := \{\mathcal{Q}_0, \ldots, \mathcal{Q}_M\}$, is chosen such that, for all $m = 0, 1, \ldots, M$, $\mathcal{Q}_m = \{\mathbf{q}_{m,1}, \ldots, \mathbf{q}_{m,r_m}\}$, is a set of r_m knots, with $\mathbf{q}_{m,i} \in \mathcal{D}$. We assume that the number of knots increases with resolution (i.e., $r_0 < r_1 < \ldots < r_M$). An illustration of such a set of knots in a simple toy example is given in Figure 3.1.

The second ingredient is a set of "modulating functions" (Sang et al., 2011), $\mathcal{T} := \{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_M\}$, where $\mathcal{T}_m : \mathcal{D} \times \mathcal{D} \to [0, 1]$ is a symmetric, nonnegative-definite function. In Section 3.2.5 we will consider two specific examples, but for now we merely require that $\mathcal{T}_m(\mathbf{s}_1, \mathbf{s}_2)$ is equal to 1 when $\mathbf{s}_1 = \mathbf{s}_2$, and (exactly) equal to 0 when $\mathbf{s}_1$ and $\mathbf{s}_2$ are far apart. Here, the meaning of "far" depends on the resolution m, in that with increasing m, the modulating function should be equal to zero for increasingly large sets of pairs of locations in $\mathcal{D}$.

Based on these ingredients, we make two definitions:

DEFINITION 1 (Predictive process). *For a Gaussian process $x(\cdot) \sim GP(0, C)$, with $x^{(m)}(\cdot)$ being defined as the predictive-process approximation (Quiñonero-Candela and Rasmussen, 2005; Banerjee et al., 2008) of $x(\cdot)$ based on the knots $\mathcal{Q}_m$:*

$$x^{(m)}(\mathbf{s}) := E\big(x(\mathbf{s})|\mathbf{x}(\mathcal{Q}_m)\big) = \mathbf{b}(\mathbf{s})'\boldsymbol{\eta}, \quad \mathbf{s} \in \mathcal{D},$$

where $\mathbf{b}(\mathbf{s})' = C(\mathbf{s}, \mathcal{Q}_m)$ and $\boldsymbol{\eta} \sim \mathcal{N}_{r_m}(\mathbf{0}, \boldsymbol{\Lambda}^{-1})$, with $\boldsymbol{\Lambda} = C(\mathcal{Q}_m, \mathcal{Q}_m)$.

That is, the predictive process is simply a conditional expectation of $y(\cdot)$, and hence a smooth, low-rank approximation of $y(\cdot)$, which can also be written as a linear combination of basis functions (cf. Katzfuss, 2013). Further, the remainder $x(\cdot) - x^{(m)}(\cdot) \sim GP(0, C_R)$ is independent of $x(\cdot)$, with positive-definite covariance function $C_R(\mathbf{s}_1, \mathbf{s}_2) = C(\mathbf{s}_1, \mathbf{s}_2) - \mathbf{b}(\mathbf{s}_1)'\boldsymbol{\Lambda}^{-1}\mathbf{b}(\mathbf{s}_2)$ (Sang and Huang, 2012).

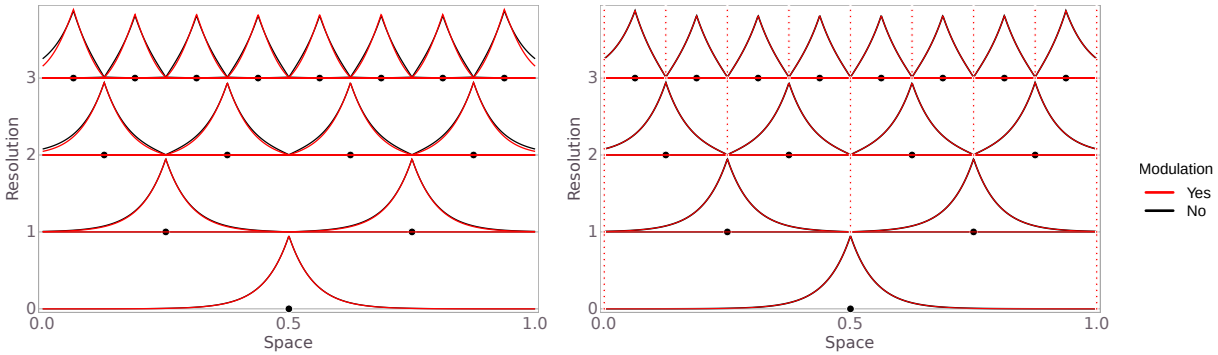
DEFINITION 2 (Modulated process). *For a Gaussian process $x(\cdot) \sim GP(0, C)$, with $[x]_{[m]}(\cdot)$ being defined as the "modulated" process corresponding to $x(\cdot)$:*

$$[x]_{[m]}(\cdot) \sim GP(0, [C]_{[m]}), \text{ where } [C]_{[m]}(\mathbf{s}_1, \mathbf{s}_2) = C(\mathbf{s}_1, \mathbf{s}_2) \cdot \mathcal{T}_m(\mathbf{s}_1, \mathbf{s}_2), \ \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D}.$$

We see that $x(\cdot)$ and $[x]_{[m]}(\cdot)$ have the same variance structure (because $\mathcal{T}_m(\mathbf{s}, \mathbf{s}) = 1$), but $[x]_{[m]}(\cdot)$ has a compactly supported covariance function that is increasingly bad approximation of C as m and the distance between $\mathbf{s}_1$ and $\mathbf{s}_2$ increase.

3.2.3 Exact multi-resolution decompositions of Gaussian processes

For any Gaussian process $y_0(\cdot) \sim GP(0, C_0)$ (as specified in Section 3.2.1), using Definition 1, we can write $y_0(\cdot) = \tau_0(\cdot) + \delta_1(\cdot)$, where $\tau_0(\cdot) := y_0^{(0)}(\cdot)$ is the predictive process of $y_0(\cdot)$ based on the knots $\mathcal{Q}_0$, and $\delta_1(\cdot) := y_0(\cdot) - \tau_0(\cdot) \sim GP(0, w_1)$ is independent from τ_0 and is itself a Gaussian process with (positive-definite) covariance function w_1. This allows us to apply again the predictive process to $\delta_1(\cdot)$ (this time based on the knots $\mathcal{Q}_1$) to obtain the decomposition



(a) M-RA-taper with $d_m = 1.75/2^m$ (b) M-RA-block

Figure 3.1: For $y_0(\cdot) \sim GP(0, C_0)$ with exponential covariance function C_0 on $\mathcal{D} = [0, 1]$, a set of multi-resolution knots (black dots) and the corresponding basis functions using the orthogonal decomposition in (3.1) (black lines) and using two versions of the M-RA (red lines) with $r_0 = 1$, $J = 2$, and $M = 3$. The M-RA-block is exact in this setting (see Proposition 6), and hence the red and black lines overlap.

$\delta_1(\cdot) = \tau_1(\cdot) + \delta_2(\cdot)$, and so forth, up to some resolution $M \in \mathbb{N}$.

This idea enables us to exactly decompose any $y_0(\cdot) \sim GP(0, C_0)$ into orthogonal components at multiple resolutions:

$$y_0(\cdot) = \tau_0(\cdot) + \ldots + \tau_{M-1}(\cdot) + \delta_M(\cdot), \tag{3.1}$$

where $\tau_m(\cdot) := \delta_m^{(m)}(\cdot)$ is the predictive process of $\delta_m(\cdot)$ based on knots $\mathcal{Q}_m$, $\delta_0(\cdot) := y_0(\cdot)$, and $\delta_m(\cdot) := \delta_{m-1}(\cdot) - \tau_{m-1}(\cdot) \sim GP(0, w_m)$ for $m = 1, \ldots, M$. Further, using the basis-function representation from Definition 1, we can write each component of the decomposition as $\tau_m(\cdot) = \mathbf{a}_m(\cdot)'\boldsymbol{\gamma}_m$, where $\boldsymbol{\gamma}_m \overset{ind.}{\sim} \mathcal{N}_{r_m}(\mathbf{0}, \boldsymbol{\Omega}^{-1})$, and starting with $w_0 = C_0$, we have for $m = 1, \ldots, M-1$:

$$\begin{aligned} \mathbf{a}_m(\mathbf{s})' &:= w_m(\mathbf{s}, \mathcal{Q}_m), \quad \mathbf{s} \in \mathcal{D} \\ \boldsymbol{\Omega}_m &:= w_m(\mathcal{Q}_m, \mathcal{Q}_m) \\ w_{m+1}(\mathbf{s}_1, \mathbf{s}_2) &:= w_m(\mathbf{s}_1, \mathbf{s}_2) - \mathbf{a}_m(\mathbf{s}_1)'\boldsymbol{\Omega}_m^{-1}\mathbf{a}_m(\mathbf{s}_2), \quad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D}. \end{aligned} \tag{3.2}$$

An important feature of this decomposition is that components $\tau_m(\cdot)$ with low resolution m capture mostly smooth, long-range dependence, whereas high-resolution components capture mostly

fine-scale, local structure. This is because the predictive process at each resolution m is an approximation to the first r_m terms in the Karhunen-Loéve (KL) expansion of $\delta_m(\cdot)$ (Sang and Huang, 2012). Figure 3.1 illustrates the resulting basis functions in our toy example.

It is straightforward to show that the decomposition of the process $y_0(\cdot) \sim GP(0, C_0)$ in (3.1) also implies an equivalent decomposition of the covariance function C_0:

$$C_0(\mathbf{s}_1, \mathbf{s}_2) = \sum_{m=0}^{M-1} w_m(\mathbf{s}_1, \mathcal{Q}_m) w_m(\mathcal{Q}_m, \mathcal{Q}_m)^{-1} w_m(\mathcal{Q}_m, \mathbf{s}_2) + w_M(\mathbf{s}_1, \mathbf{s}_2), \quad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D}. \tag{3.3}$$

3.2.4 The multi-resolution approximation

The multi-resolution approximation (M-RA) is a "modulated" version of the exact decomposition in (3.1), which at each resolution m modulates the remainder using the function $\mathcal{T}_m$ from Section 3.2.2. The key idea is that the predictive processes at low resolutions pick up the low-frequency variation in $y_0(\cdot)$. As a result, the remainder terms have smaller and smaller variability as m increases, therefore approximating the remainder with more and more restrictive modulating functions causes little approximation error.

DEFINITION 3 (Multi-resolution approximation (M-RA)). *For a given $M \in \mathbb{N}$, the M-RA of a process $y_0(\cdot) \sim GP(0, C_0)$ based on a set of knots $\mathcal{Q} = \{\mathcal{Q}_0, \ldots, \mathcal{Q}_M\}$ and a set of modulating functions $\mathcal{T} = \{\mathcal{T}_0, \ldots, \mathcal{T}_M\}$, is given by*

$$y_M(\cdot) = \sum_{m=0}^{M} \widetilde{\tau}_m(\cdot) = \sum_{m=0}^{M} \mathbf{b}_m(\mathbf{s})' \boldsymbol{\eta}_m, \tag{3.4}$$

where $\widetilde{\tau}_m(\cdot) := \widetilde{\delta}_m^{(m)}(\cdot)$ and $\boldsymbol{\eta}_m \overset{ind.}{\sim} \mathcal{N}_{r_m}(\mathbf{0}, \boldsymbol{\Lambda}_m^{-1})$ for $m = 0, \ldots, M$; $\widetilde{\delta}_0(\cdot) := [y_0]_{[0]}(\cdot) \sim GP(0, v_0)$ with $v_0 = [C_0]_{[0]}$; $\widetilde{\delta}_m(\cdot) = [\widetilde{\delta}_{m-1} - \widetilde{\tau}_{m-1}]_{[m]}(\cdot) \sim GP(0, v_m)$ for $m = 1, \ldots, M$;

and

$$\begin{aligned}
\mathbf{b}_m(\mathbf{s})' &:= v_m(\mathbf{s}, \mathcal{Q}_m), \quad \mathbf{s} \in \mathcal{D}, \quad m = 0, \ldots, M, \\
\mathbf{\Lambda}_m &:= v_m(\mathcal{Q}_m, \mathcal{Q}_m), \quad m = 0, \ldots, M, \\
v_{m+1}(\mathbf{s}_1, \mathbf{s}_2) &:= \big(v_m(\mathbf{s}_1, \mathbf{s}_2) - \mathbf{b}_m(\mathbf{s}_1)' \mathbf{\Lambda}_m^{-1} \mathbf{b}_m(\mathbf{s}_2)\big) \cdot \mathcal{T}_{m+1}(\mathbf{s}_1, \mathbf{s}_2), \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D}, m = 0, \ldots, M-1.
\end{aligned} \tag{3.5}$$

Figure 3.1 shows the M-RA basis functions in our toy example. As can be seen, the M-RA is similar to a wavelet model, in that for increasing resolution m, we have an increasing number of basis functions with increasingly compact support. However, in contrast to wavelets, the basis functions $\mathbf{b}(\cdot)$ and the precision matrix $\mathbf{\Lambda}$ of the corresponding weights in the M-RA adapt to the covariance function C_0.

For ease of notation, we often stack the basis functions as $\mathbf{b}(\cdot) := \big(\mathbf{b}_0(\cdot)', \ldots, \mathbf{b}_M(\cdot)'\big)'$ and the corresponding coefficients, $\boldsymbol{\eta} := \big(\boldsymbol{\eta}_0', \ldots, \boldsymbol{\eta}_M'\big)'$, so that

$$y_M(\cdot) = \mathbf{b}(\cdot)' \boldsymbol{\eta}, \qquad \text{where } \boldsymbol{\eta} \sim \mathcal{N}_r(\mathbf{0}, \mathbf{\Lambda}^{-1}), \tag{3.6}$$

with $\mathbf{\Lambda} := blockdiag(\mathbf{\Lambda}_0, \ldots, \mathbf{\Lambda}_M)$ and $r = \sum_{m=0}^{M} r_m$.

3.2.5 Specific examples

As described in Section 3.2.2, the M-RA requires the choice of two ingredients: knots and modulating functions. In light of the computational complexities discussed in Sections 3.3.2–3.3.3 below, we introduce a factor J, often chosen to be equal to 2 or 4. Then, starting with some (small) number of knots r_0 at resolution $m = 0$, we henceforth assume $r_m = J r_{m-1}$ for $m = 1, \ldots, M$.

Regarding the modulating functions, we will now discuss two choices that lead to two important versions of the M-RA.

3.2.5.1 M-RA-block

To define the M-RA-block, we need to partition of the spatial domain $\mathcal{D}$ recursively, with J subregions, $\mathcal{D}_1, \ldots, \mathcal{D}_J$, which is again divided into J smaller subregions at each resolution, up to level M:

$$\mathcal{D}_{j_1,\ldots,j_{m-1}} = \dot{\bigcup}_{j_m=1,\ldots,J} \mathcal{D}_{j_1,\ldots,j_m}, \quad j_1, \ldots, j_{m-1} = 1, \ldots, J; \quad m = 1, \ldots, M.$$

We then assume for each resolution m that the modulated remainder $\delta_m(\cdot)$ is independent across partitions at the mth resolution. That is, the modulating function is defined as

$$\mathcal{T}_m(\mathbf{s}_i, \mathbf{s}_j) = \begin{cases} 1, & (i_1, \ldots, i_m) = (j_1, \ldots, j_m), \\ 0, & \text{otherwise}, \end{cases} \qquad \mathbf{s}_i \in \mathcal{D}_{i_1,\ldots,i_m},\ \mathbf{s}_j \in \mathcal{D}_{j_1,\ldots,j_m}. \tag{3.7}$$

Simply speaking, we have $\mathcal{T}_m(\mathbf{s}_1, \mathbf{s}_2) = 1$ if $\mathbf{s}_1, \mathbf{s}_2$ are from the same region $\mathcal{D}_{j_1,\ldots,j_m}$, and $\mathcal{T}_m(\mathbf{s}_1, \mathbf{s}_2) = 0$ otherwise. At resolution m, $\mathcal{D}$ is split into J^m subregions. Typically, we assume that the knots at each resolution are roughly equally spread throughout the domain, so that there are roughly the same number $r_m/J^m = r_0$ of knots in every such region.

The M-RA-block and the corresponding domain partitioning are illustrated in a toy example in Figure 3.1b. A special case of the M-RA-block with $\mathcal{Q}_M = \mathcal{S}$ was first proposed in Katzfuss (2017). Another special case with $M = 1$ is the block-full-scale approximation (Snelson and Ghahramani, 2007; Sang et al., 2011).

3.2.5.2 M-RA-taper

We can also specify the modulating functions to be compactly supported correlation functions, often refered to as tapering functions. For simplicity, we assume here that the modulating functions are of the form,

$$\mathcal{T}_m(\mathbf{s}_1, \mathbf{s}_2) = \mathcal{T}_*(\|\mathbf{s}_1 - \mathbf{s}_2\|/d_m),$$

with $d_{m+1} = d_m/J^{1/d}$, where d is the dimension of $\mathcal{D}$, $\|\cdot\|$ is some norm on $\mathcal{D}$, and $\mathcal{T}_*$ is a compactly supported correlation function that is scaled such that $\mathcal{T}^*(x) = 0$ for all $x \geq 1$. In all data examples in this manuscript, we will use Kanter's function (Kanter, 1997):

$$\mathcal{T}_*(x) := \begin{cases} 1, & x = 0, \\ (1-x)\frac{\sin(2\pi x)}{2\pi x} + \frac{1-\cos(2\pi x)}{2\pi^2 x}, & x \in (0,1), \\ 0, & x \geq 1. \end{cases}$$

For other possible choices of tapering functions, see Gneiting (2002). The taper-M-RA is illustrated in Figure 3.1a. A special case of the M-RA-taper with $M = 1$ is the taper-full-scale approximation (Sang and Huang, 2012; Katzfuss, 2013).

3.2.6 Properties of the M-RA process

Throughout this subsection, let $y_M(\cdot)$ be the M-RA (from Definition 3) of $y_0(\cdot) \sim GP(0, C_0)$ on domain $\mathcal{D}$ based on knots $\mathcal{Q} = \{\mathcal{Q}_0, \ldots, \mathcal{Q}_M\}$ and modulating functions $\mathcal{T} = \{\mathcal{T}_0, \ldots, \mathcal{T}_M\}$.

PROPOSITION 1 (Distribution of the M-RA). *The M-RA is a Gaussian process, $y_M(\cdot) \sim GP(0, C_M)$, with covariance function*

$$C_M(\mathbf{s}_1, \mathbf{s}_2) = \sum_{m=0}^{M} v_m(\mathbf{s}_1, \mathcal{Q}_m) v_m(\mathcal{Q}_m, \mathcal{Q}_m)^{-1} v_m(\mathcal{Q}_m, \mathbf{s}_2), \quad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D},$$

where v_m is defined in (3.5). *We call C_M the M-RA of the covariance function C_0.*

PROPOSITION 2 (Duplication of knots). *If $\mathbf{q} \in \mathcal{Q}_m$, then $v_{m+l}(\mathbf{q}, \mathbf{s}) = 0$ for any $\mathbf{s} \in \mathcal{D}$ and $l \geq 1$.*

This proposition implies that there is no benefit to designate the same locations as knots at multiple resolutions; that is, all knot locations in $\mathcal{Q}$ should be unique.

PROPOSITION 3 (Exact variance). *If $\mathbf{s} \in \mathcal{Q}$, then the M-RA variance at location $\mathbf{s}$ is exact; that is, $C_M(\mathbf{s}, \mathbf{s}) = C_0(\mathbf{s}, \mathbf{s})$.*

This proposition implies that, in contrast to other recent basis-function approaches (e.g., Lind-

gren et al., 2011; Nychka et al., 2015), no variance or "edge" correction is needed for the M-RA if we place a knot location at each observed location.

PROPOSITION 4 (Smoothness). *If realizations (i.e., sample paths) of $\mathbf{y}_0(\cdot)$ are exactly p times differentiable at $\mathbf{s} \in \mathcal{Q}$, then realizations of $\mathbf{y}_M(\cdot)$ are also exactly p times differentiable at $\mathbf{s}$, provided that $C_0(\cdot, \mathbf{q})$ and $\mathcal{T}_m(\cdot, \mathbf{q})$ are at least $2p$ times differentiable at $\mathbf{s}$, for any $\mathbf{q} \in \mathcal{Q}$ and $m = 1, \ldots, M$.*

Many commonly used covariance functions (e.g., Matérn) are infinitely differentiable away from the origin. If C_0 is such a covariance function, the M-RA-block thus has the same smoothness as the original process $\mathbf{y}_0(\cdot)$ at any $\mathbf{s}$ that is not located on the boundary between subregions at any resolution (cf. Katzfuss, 2017). Tapering functions are often smooth away from the origin, except at the distance at which they become exactly zero. Thus, the M-RA-taper will typically have the same smoothness at $\mathbf{s}$ as $\mathbf{y}_0(\cdot)$ if $\mathcal{T}$ is at least $2p$ times differentiable at the origin and $\mathbf{s}$ is not exactly at distance d_m from any $\mathbf{q} \in \mathcal{Q}_m$, for all $m = 1, \ldots, M$. Note that this result does not require the smoothness of y_0 to be the same at all locations $\mathbf{s}$; if the smoothness (or other local characteristics) of the covariance function C_0 varies over space, the M-RA will automatically adapt to this nonstationarity and vary over space accordingly.

There is, however, an issue with the continuity of the M-RA-block process at the region boundaries:

PROPOSITION 5 (Continuity). *Assume that C_0 is a continuous function. Then, for the M-RA-taper, realizations of the corresponding process $y_M(\cdot)$ and the posterior mean (i.e., kriging prediction) surface $\mu_M(\mathbf{s}) := E(y_M(\mathbf{s})|\mathbf{z})$ based on observations $\mathbf{z}$ as in* (3.8) *are both continuous, assuming that $\mathcal{T}_m$ is continuous for all $m = 0, 1, \ldots, M$. In contrast, for the M-RA-block, $y_M(\cdot)$ and $\mu_M(\cdot)$ are both discontinuous in general at any $\mathbf{s}$ on the boundary between any two subregions.*

PROPOSITION 6 (Exactness of M-RA-block). *Let C_0 be a (stationary) exponential covariance function on the real line, $\mathcal{D} = \mathbb{R}$. Further, let C_M be the covariance function of the corresponding M-RA-block (see Section 3.2.5.1) with $r_m = (J-1)J^m$ knots for $m = 0, \ldots, M-1$, which are*

placed such that at each resolution m, a knot is located on each boundary between two subregions at resolution $m+1$. Then, the M-RA is exact at every knot location; that is, $C_M(\mathbf{s}_1, \mathbf{s}_2) = C_0(\mathbf{s}_1, \mathbf{s}_2)$ for any $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{Q}$.

This proposition is illustrated in Figure 3.1b. As we will see in Section 3.3.2, this result allows us to exactly decompose a $n \times n$ exponential covariance matrix in terms of a sparse matrix with n rows but only about $\log_2 n$ nonzero elements per row with $r_0 = 1$ and $J = 2$. This leads to tremendous computational savings (e.g., $\log_2(n) < 30$ for $n = 1$ billion).

While the exact result in Proposition 6 relies on the Markov property and the exact screening effect of the exponential covariance function (which is a Matérn covariance with smoothness parameter $\nu = 0.5$), similar but approximate results are expected to hold for larger smoothness parameters in one dimension. Specifically, Stein (2011) shows that an asymptotic screening effect holds for $\nu = 1.5$ when using conditioning sets of size 2, and he conjectures that an asymptotic screening effect holds for any ν when using conditioning sets of size greater than ν. This conjecture is also explored numerically in Katzfuss and Guinness (2017). To exploit this screening effect using the M-RA-block, we can simply place $c > \nu$ knots near every subregion boundary (i.e., $r_0 = c(J-1)$).

3.3 Inference

In this section, we discuss the inference for the M-RA, based on a set of measurements at locations $\mathcal{S}$ of size n. We assume additive, independent measurement error, such that

$$\mathbf{z} = \mathbf{y}_M(\mathcal{S}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}_n(\mathbf{0}, \mathbf{V}_\epsilon), \tag{3.8}$$

where $\mathbf{V}_\epsilon$ is a diagonal matrix that might depend on the parameter vector, $\boldsymbol{\theta}$. Throughout this section, we assume that $\boldsymbol{\theta}$ is fixed at a particular value, unless noted otherwise. For the sparsity and complexity calculations, we assume $r_m = r_0 J^m$ and $n = \mathcal{O}(r_M)$.

3.3.1 General inference results

3.3.1.1 Prior matrices

For a given set of parameters, the covariance function C_0, the basis functions $\mathbf{b}(\cdot)$ and the precision matrix $\mathbf{\Lambda}$ in (3.6) are fixed. The first step for inference is to calculate the prior matrices $\mathbf{\Lambda}$ and $\mathbf{B} := [\mathbf{B}_0, \ldots, \mathbf{B}_M] := [\mathbf{b}_0(\mathcal{S}), \ldots, \mathbf{b}_M(\mathcal{S})]$. Define $\mathbf{W}^k_{m,l} := v_k(\mathcal{Q}_m, \mathcal{Q}_l)$ and $\mathbf{W}^k_{\mathcal{S},m} := v_k(\mathcal{S}, \mathcal{Q}_m)$, so that $\mathbf{\Lambda}_m = \mathbf{W}^m_{m,m}$ and $\mathbf{B}_m = \mathbf{W}^m_{\mathcal{S},m}$. For $m = 0, \ldots, M$, starting with $\mathbf{W}^0_{m,l} = v_0(\mathcal{Q}_m, \mathcal{Q}_l)$ and $\mathbf{W}^0_{\mathcal{S},m} = v_0(\mathcal{S}, \mathcal{Q}_m)$, it is straightforward to verify that

$$\mathbf{W}^{k+1}_{m,l} = \left(\mathbf{W}^k_{m,l} - \mathbf{W}^k_{m,k}\mathbf{\Lambda}_k^{-1}\mathbf{W}^k_{l,k}{}'\right) \circ \mathcal{T}_{k+1}(\mathcal{Q}_m, \mathcal{Q}_l), \qquad k = 0, \ldots, l-1; \quad l = 0, \ldots, m; \tag{3.9}$$

and

$$\mathbf{W}^{k+1}_{\mathcal{S},m} = \left(\mathbf{W}^k_{\mathcal{S},m} - \mathbf{W}^k_{\mathcal{S},k}\mathbf{\Lambda}_k^{-1}\mathbf{W}^k_{m,k}{}'\right) \circ \mathcal{T}_{k+1}(\mathcal{S}, \mathcal{Q}_m), \qquad k = 0, \ldots, m-1. \tag{3.10}$$

Here, $\circ$ denotes the Hadamard or element-wise product. Note that $\mathbf{\Lambda}_m$ and $\mathbf{B}_m$ both grow in dimension and become increasingly sparse with increasing resolution m. We have $(\mathbf{\Lambda}_m)_{i,j} = 0$ if $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{q}_{m,j}) = 0$, and $(\mathbf{B}_m)_{i,j} = 0$ if $\mathcal{T}_m(\mathbf{s}_i, \mathbf{q}_{m,j}) = 0$.

3.3.1.2 Posterior inference

Once $\mathbf{\Lambda}$ and $\mathbf{B}$ have been obtained, the posterior distribution of the unknown weight vector, $\boldsymbol{\eta}$, is given by well-known formulas for conjugate normal-normal Bayesian models:

$$\boldsymbol{\eta} \,|\, \mathbf{z} \sim \mathcal{N}_r(\widetilde{\boldsymbol{\nu}}, \widetilde{\mathbf{\Lambda}}^{-1}), \tag{3.11}$$

where $\widetilde{\mathbf{\Lambda}} = \mathbf{\Lambda} + \mathbf{B}'\mathbf{V}_\epsilon^{-1}\mathbf{B}$, $\widetilde{\boldsymbol{\nu}} = \widetilde{\mathbf{\Lambda}}^{-1}\widetilde{\mathbf{z}}$, and $\widetilde{\mathbf{z}} = \mathbf{B}'\mathbf{V}_\epsilon^{-1}\mathbf{z}$.

Based on this posterior distribution of $\boldsymbol{\eta}$, the likelihood can be written as (e.g., Katzfuss and

Hammerling, 2017):

$$-2\log L(\boldsymbol{\theta}) = -\log|\boldsymbol{\Lambda}| + \log|\widetilde{\boldsymbol{\Lambda}}| + \log|\mathbf{V}_\epsilon| + \mathbf{z}'\mathbf{V}_\epsilon^{-1}\mathbf{z} - \widetilde{\mathbf{z}}'\widetilde{\boldsymbol{\Lambda}}^{-1}\widetilde{\mathbf{z}}. \tag{3.12}$$

Using this expression, the likelihood can be evaluated quickly for any given value of the parameter vector $\boldsymbol{\theta}$. This allows us to carry out likelihood-based inference (e.g., maximum likelihood or Metropolis-Hastings) on the parameters in C_0 and $\mathbf{V}_\epsilon$, by computing the quantities in (3.9)–(3.12) for each parameter value.

To obtain spatial predictions for fixed parameters $\boldsymbol{\theta}$, note that $\mathbf{y}_M(\mathcal{S}^P) = \mathbf{B}^P\boldsymbol{\eta}$, where $\mathbf{B}^P := \mathbf{b}(\mathcal{S}^P)$. Defining $\mathbf{W}^k_{\mathcal{S}^P,l} := v_k(\mathcal{S}^P, \mathcal{Q}_l)$, $\mathbf{B}^P = [\mathbf{B}^P_0, \ldots, \mathbf{B}^P_M]$ can be obtained based on the quantities from Section 3.3.1.1 by calculating $\mathbf{W}^0_{\mathcal{S}^P,m} = v_0(\mathcal{S}^P, \mathcal{Q}_m)$ and

$$\mathbf{W}^{k+1}_{\mathcal{S}^P,m} = \left(\mathbf{W}^k_{\mathcal{S}^P,m} - \mathbf{W}^k_{\mathcal{S}^P,k}\boldsymbol{\Lambda}_k^{-1}\mathbf{W}^{k\,\prime}_{m,k}\right) \circ \mathcal{T}_{k+1}(\mathcal{S}^P, \mathcal{Q}_m), \qquad k = 0, \ldots, m-1,$$

and setting $\mathbf{B}^P_m = \mathbf{W}^m_{\mathcal{S}^P,m}$, for $m = 0, \ldots, M$. The posterior predictive distribution is given by,

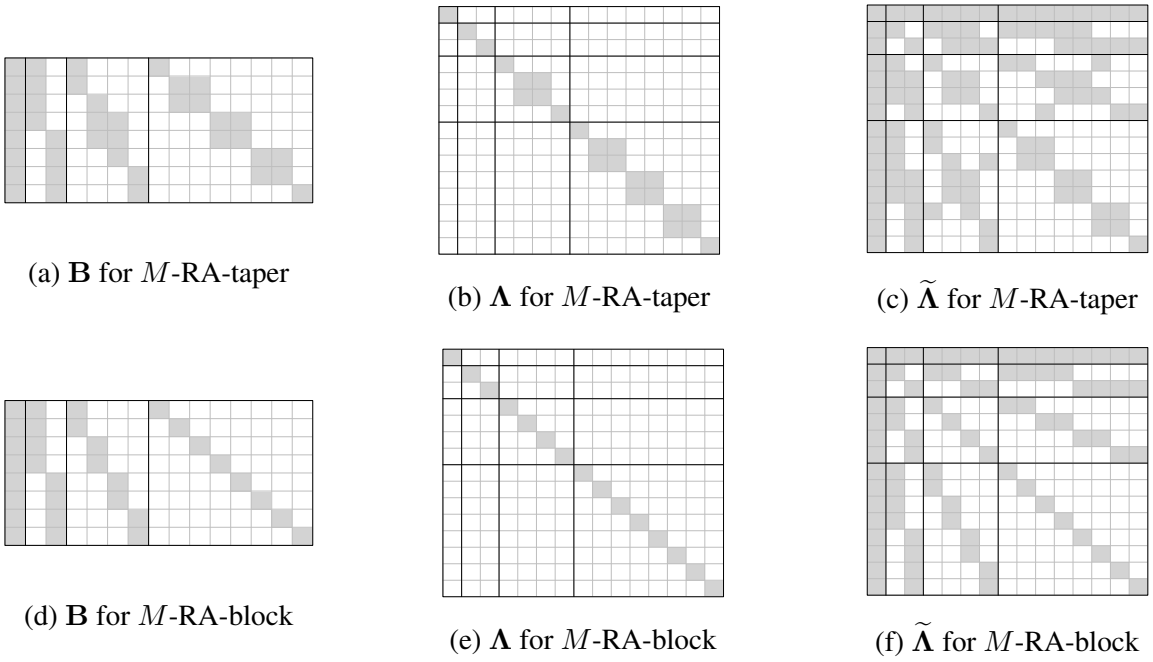
$$\mathbf{y}_M(\mathcal{S}^P) \,|\, \mathbf{z} \sim \mathcal{N}_{n_P}(\mathbf{B}^P\widetilde{\boldsymbol{\nu}}, \mathbf{B}^P\widetilde{\boldsymbol{\Lambda}}^{-1}\mathbf{B}^{P\prime}). \tag{3.13}$$

Hence, the main computational effort required for inference is the Cholesky decomposition of $\widetilde{\boldsymbol{\Lambda}}$, the posterior precision matrix of the basis-function weights in (3.11). As $\boldsymbol{\Lambda}$ and $\mathbf{B}$ are both sparse, $\widetilde{\boldsymbol{\Lambda}}$ is a sparse matrix that can be decomposed quickly. Specifically, $\widetilde{\boldsymbol{\Lambda}}$ has the block structure $\widetilde{\boldsymbol{\Lambda}} = (\widetilde{\boldsymbol{\Lambda}}_{m,l})_{m,l=0,\ldots,M}$, where $\widetilde{\boldsymbol{\Lambda}}_{m,l} = \boldsymbol{\Lambda}_m \mathbb{1}_{\{m=l\}} + \mathbf{B}'_m\mathbf{V}_\epsilon^{-1}\mathbf{B}_l$ is an $r_m \times r_l$ matrix whose (i,j)th element is 0 if $\nexists \mathbf{s} \in \mathcal{D}$ such that $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{s}) \neq 0$ and $\mathcal{T}_l(\mathbf{q}_{l,j}, \mathbf{s}) \neq 0$. Figure 3.2 shows the sparsity structures of $\mathbf{B}$, $\boldsymbol{\Lambda}$, and $\widetilde{\boldsymbol{\Lambda}}$ corresponding to the toy example in Figure 3.1.

3.3.1.3 Inference in the absence of measurement error

If there is no measurement error (i.e., $\mathbf{V}_\epsilon = \mathbf{0}$), we have

$$\mathbf{z} = \mathbf{y} \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma}).$$



(a) $\mathbf{B}$ for M-RA-taper

(b) $\mathbf{\Lambda}$ for M-RA-taper

(c) $\widetilde{\mathbf{\Lambda}}$ for M-RA-taper

(d) $\mathbf{B}$ for M-RA-block

(e) $\mathbf{\Lambda}$ for M-RA-block

(f) $\widetilde{\mathbf{\Lambda}}$ for M-RA-block

Figure 3.2: Illustration of the sparsity in the matrices $\mathbf{B}$, $\mathbf{\Lambda}$, and $\widetilde{\mathbf{\Lambda}}$ for the toy example in Figure 3.1. Resolutions are separated by solid black lines. Top row: M-RA-taper. Bottom row: M-RA-block.

where $\mathbf{\Sigma} = \mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{B}'$. To ensure that $\mathbf{B}$ (and hence $\mathbf{\Sigma}$) has full rank, we assume for this case that $\mathcal{S} = \mathcal{Q}$ (and thus $n = r$) and (in light of Proposition 2) that the knots are unique. The likelihood can then be calculated as $-2\log L(\boldsymbol{\theta}) = -\log|\mathbf{\Sigma}| - \mathbf{y}'\mathbf{\Sigma}^{-1}\mathbf{y}$, where $\log|\mathbf{\Sigma}| = \log|\mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{B}'| = \log|\mathbf{B}|^2 - \log|\mathbf{\Lambda}|$, and $\mathbf{y}'\mathbf{\Sigma}^{-1}\mathbf{y} = \widetilde{\mathbf{y}}'\mathbf{\Lambda}\widetilde{\mathbf{y}}$ with $\widetilde{\mathbf{y}} = \mathbf{B}^{-1}\mathbf{y}$.

3.3.2 Inference details for the M-RA-block

For the M-RA-block from Section 3.2.5.1, $\mathbf{B}$, $\mathbf{\Lambda}$, and $\widetilde{\mathbf{\Lambda}}$ are block-sparse matrices, with each block roughly of size $r_0 \times r_0$ and corresponding to (the knots at) a pair of regions.

As noted in Section 3.3.1.1, we have $(\mathbf{\Lambda}_m)_{i,j} = 0$ if $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{q}_{m,j}) = 0$, and so $\mathbf{\Lambda}_m$ is a block-diagonal matrix with diagonal blocks $\{v_m(\mathcal{Q}^{j_1,\ldots,j_m}, \mathcal{Q}^{j_1,\ldots,j_m}) : j_1,\ldots,j_m = 1,\ldots,J\}$, where $\mathcal{Q}^{j_1,\ldots,j_m} = \{\mathbf{q}_{m,i} : \mathbf{q}_{m,i} \in \mathcal{Q}_m \cap \mathcal{D}_{j_1,\ldots,j_m}\}$ is the set of roughly r_0 knots at resolution m that lie in $\mathcal{D}_{j_1,\ldots,j_m}$. It is well known that the inverse $\mathbf{\Lambda}_k^{-1}$ of a block-diagonal matrix $\mathbf{\Lambda}_k$ has the same block-diagonal structure as $\mathbf{\Lambda}_k$, and so the prior calculations in Section 3.3.1.1 involving $\mathbf{\Lambda}_k^{-1}$ can be carried out at low computational cost.

For the posterior covariance matrix, we have from Section 3.3.1.2 that $(\widetilde{\mathbf{\Lambda}}_{m,l})_{i,j} = 0$ if $\nexists \mathbf{s} \in \mathcal{D}$ such that $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{s}) \neq 0$ and $\mathcal{T}_l(\mathbf{q}_{l,j}, \mathbf{s}) \neq 0$, and so the block in $\widetilde{\mathbf{\Lambda}}$ corresponding to regions $\mathcal{D}_{i_1,\ldots,i_m}$ and $\mathcal{D}_{j_1,\ldots,j_m}$ is zero if the regions do not overlap (i.e., if $\mathcal{D}_{i_1,\ldots,i_m} \cap \mathcal{D}_{j_1,\ldots,j_m} = \emptyset$). The Cholesky factor of a (appropriately reordered) matrix with this particular block-sparse structure has zero fill-in, and can thus be carried out very rapidly.

Katzfuss (2017) describe an algorithm for inference in a special case of the M-RA-block that can be straightforwardly extended to the more general M-RA-block considered here. This algorithm is well suited for parallel and distributed computations for massive datasets, and it leads to efficient storage of the full posterior predictive distribution in (3.13). The time and memory complexity are shown to be $\mathcal{O}(nM^2r_0^2)$ and $\mathcal{O}(nMr_0)$, respectively.

3.3.3 Inference details for the M-RA-taper

The case of the M-RA-taper from Section 3.2.5.2 results in sparse matrices, but care must be taken to ensure computational feasibility. A crucial observation for the computational results below is that for any location $\mathbf{s} \in \mathcal{D}$ and any resolution m, only $\mathcal{O}(r_0)$ knots from $\mathcal{Q}_m$ are within a distance of d_m from $\mathbf{s}$ (i.e., all sets of the form $\{\mathbf{q}_{m,i} \in \mathcal{Q}_m : \|\mathbf{s} - \mathbf{q}_{m,i}\| \leq d_m\}$ contain only $\mathcal{O}(r_0)$ elements), because we assumed that the $r_m = r_0 J^m$ knots at resolution m are roughly equally spread over the domain $\mathcal{D}$, and $d_m = d_0/J^{m/d}$.

First, consider calculation of the prior matrices as described in Section 3.3.1.1. The matrices $\mathbf{\Lambda}$ and $\mathbf{B}$ have $\mathcal{O}(nr_0)$ and $\mathcal{O}(nMr_0)$ nonzero elements, respectively, because $(\mathbf{\Lambda}_m)_{i,j} = 0$ if $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{q}_{m,j}) = 0$, and $(\mathbf{B}_m)_{i,j} = 0$ if $\mathcal{T}_m(\mathbf{s}_i, \mathbf{q}_{m,j}) = 0$. Before carrying out the actual inference procedures, it is helpful to pre-calculate $\mathcal{I}_{m,l} := \{(i,j) : \mathcal{T}_l(\mathbf{q}_{m,i}, \mathbf{q}_{l,j}) \neq 0\}$, the set of nonzero indices of the matrix $\mathbf{W}_{m,l}^l$, for $l = 0, \ldots, m$ and $m = 0, \ldots, M$. This can typically be done in $\mathcal{O}(n \log n)$ time (e.g. Vaidya, 1989). In the actual inference procedure, we then only need to calculate the $\mathcal{I}_{m,l}$-elements of the matrices $\mathbf{W}_{m,l}^k$ in (3.9). The main difficulty herein is that while $\mathbf{\Lambda}_k$ is sparse, its inverse $\mathbf{\Lambda}_k^{-1}$ is not. However, we only need to compute certain elements of $\mathbf{\Lambda}_k^{-1}$:

PROPOSITION 7. *For $l = 0, \ldots, m$ and $m = 0, \ldots, M$, the matrix $\mathbf{W}_{m,l}^l$ can be obtained by*

computing

$$\mathbf{W}_{m,l}^{k+1} = \left(\mathbf{W}_{m,l}^{k} - \mathbf{W}_{m,k}^{k}\mathbf{S}_k\mathbf{W}_{l,k}^{k\,\prime}\right) \circ \mathcal{T}_{k+1}(\mathcal{Q}_m, \mathcal{Q}_l), \qquad k = 0, \ldots, l-1, \tag{3.14}$$

where $\mathbf{S}_k = \mathbf{\Lambda}_k^{-1} \circ \mathbf{G}_k$ *and* $(\mathbf{G}_k)_{i,j} = \mathbb{1}_{\{\|\mathbf{q}_{m,i} - \mathbf{q}_{m,j}\| < (2+2/J)d_m\}}$. *Thus, the* (i,j) *element of* $\mathbf{\Lambda}_m^{-1}$ *is not required for calculating the prior matrices in* (3.9) *if* $\|\mathbf{q}_{m,i} - \mathbf{q}_{m,j}\| \geq (2 + 2/J)\, d_m$.
The total time complexity for computing all prior matrices in (3.9) *is* $\mathcal{O}(nM^2r_0^3)$*, ignoring the cost of computing the* $\mathbf{S}_k$ *from the* $\mathbf{\Lambda}_k$.

To calculate $\mathbf{S}_k$ from $\mathbf{\Lambda}_k$, we use a selected inversion algorithm (Erisman and Tinney, 1975; Li et al., 2008; Lin et al., 2011) in which we regard element (i,j) as a structural zero only if $\|\mathbf{q}_{k,i} - \mathbf{q}_{k,j}\| \geq (2 + 2/J)d_m$. This algorithm the same computational complexity as the Cholesky decomposition of the same matrix. For one-dimensional domains ($d = 1$), $\mathbf{\Lambda}_k$ is a banded matrix with bandwidth $\mathcal{O}(r_0)$, and so the time complexity to compute its Cholesky decomposition (and selected inverse) is $\mathcal{O}(r_k r_0^2)$ (e.g., Gelfand et al., 2010, p. 187). For $d \geq 2$, the rows and columns of $\mathbf{\Lambda}$ should be ordered such that the Cholesky decomposition leads to a (near) minimal fill-in and hence fast computations. Functions for this reordering are readily available in most statistical or linear-algebra software. The discussion in Furrer et al. (2006) indicates that the resulting time complexity for the Cholesky decomposition is roughly linear in the matrix dimension for $d = 2$. Moreover, our numerical experiments showed that the selected inversions only account for a small fraction of the total time required to compute the prior matrices, and so the total computation time for computing the prior matrices scales roughly as $\mathcal{O}(nM^2r_0^3)$.

Once the prior matrices including $\mathbf{B}$ and $\mathbf{\Lambda}$ have been obtained, posterior inference requires computing and decomposing the posterior precision matrix $\widetilde{\mathbf{\Lambda}} = \mathbf{\Lambda} + \mathbf{B}'\mathbf{V}_\epsilon^{-1}\mathbf{B}$ in (3.11), with (m,l)th block $\widetilde{\mathbf{\Lambda}}_{m,l} = \mathbf{\Lambda}_m \mathbb{1}_{\{m=l\}} + \mathbf{B}_m'\mathbf{V}_\epsilon^{-1}\mathbf{B}_l$. The (j,k)th element of this block is

$$(\widetilde{\mathbf{\Lambda}}_{m,l})_{j,k} = (\mathbf{\Lambda}_m)_{j,k}\mathbb{1}_{\{m=l\}} + \textstyle\sum_{i=1}^{n} v_m(\mathbf{s}_i, \mathbf{q}_{m,j}) v_l(\mathbf{s}_i, \mathbf{q}_{l,k})(\mathbf{V}_\epsilon)_{i,i}^{-1}.$$

As each of the n $\mathbf{s}_i$ is within distances of d_m and d_l of $\mathcal{O}(r_0)$ elements of $\mathcal{Q}_m$ and $\mathcal{Q}_l$, respectively,

the time complexity to compute $(\mathbf{B}'\mathbf{B})_{m,l}$ is $\mathcal{O}(nr_0^2)$, and hence computing $\widetilde{\mathbf{\Lambda}}$ requires $\mathcal{O}(nM^2r_0^2)$ time.

PROPOSITION 8. *The number of nonzero elements in $\widetilde{\mathbf{\Lambda}}$ is $\mathcal{O}(nMr_0)$.*

The time complexity for obtaining the Cholesky decomposition of $\widetilde{\mathbf{\Lambda}}$ is difficult to quantify, as it depends on its sparsity structure and the chosen ordering, but again our numerical experiments showed that the contribution of the Cholesky decomposition to the overall computation time is relatively small when appropriate reordering algorithms are used.

For prediction, the posterior covariance $\mathbf{B}^P\widetilde{\mathbf{\Lambda}}^{-1}\mathbf{B}^{P\prime}$ in (3.13) is dense and hence cannot be obtained explicitly for a large number of prediction locations. But the posterior covariance matrix of a moderate number of linear combinations $\mathbf{L}\mathbf{y}(\mathcal{S}^P)$ can be obtained as $(\mathbf{L}\mathbf{B}^P)\widetilde{\mathbf{\Lambda}}^{-1}(\mathbf{L}\mathbf{B}^P)'$, also based on a Cholesky decomposition of $\widetilde{\mathbf{\Lambda}}$.

In summary, the time and memory complexity of the M-RA-taper are $\mathcal{O}(nM^2r_0^3)$ and $\mathcal{O}(nMr_0)$, respectively, plus the cost of computing the Cholesky decompositions of $\mathbf{\Lambda}$ and $\widetilde{\mathbf{\Lambda}}$. These decompositions only accounted for a relatively small amount of the overall computation time in our numerical experiments. Thus, the time complexity of the M-RA-taper is roughly cubic in r_0 while it is square in r_0 for the M-RA-block. However, the actual computational cost for the M-RA-taper can be reduced when the covariance function C_0 has a small effective range relative to the size of $\mathcal{D}$, because then C_0 can be tapered at resolution 0 without causing a large approximation error; in contrast, for the M-RA-block, we always have $\mathcal{T}_0(\mathbf{s}_1, \mathbf{s}_2) \equiv 1$. As explained in Katzfuss (2017), it is often appropriate to expect a good approximation for $M = \mathcal{O}(\log n)$ (and hence $r_0 = \mathcal{O}(1)$), which results in quasilinear complexity as a function of n for the M-RA.

3.4 Simulation study

For this section, we used data simulated from a true Gaussian process to compare the M-RA-block and M-RA-taper to full-scale approximations, FSA-block (Sang et al., 2011) and FSA-taper (Sang and Huang, 2012), which correspond to the 1-RA-block and 1-RA-taper, respectively. An implementation of the simulations in Julia (http://julialang.org) version 0.4.5 was run on a 16-core

machine with 64G RAM.

The true Gaussian process was assumed to have mean zero and an exponential covariance function,

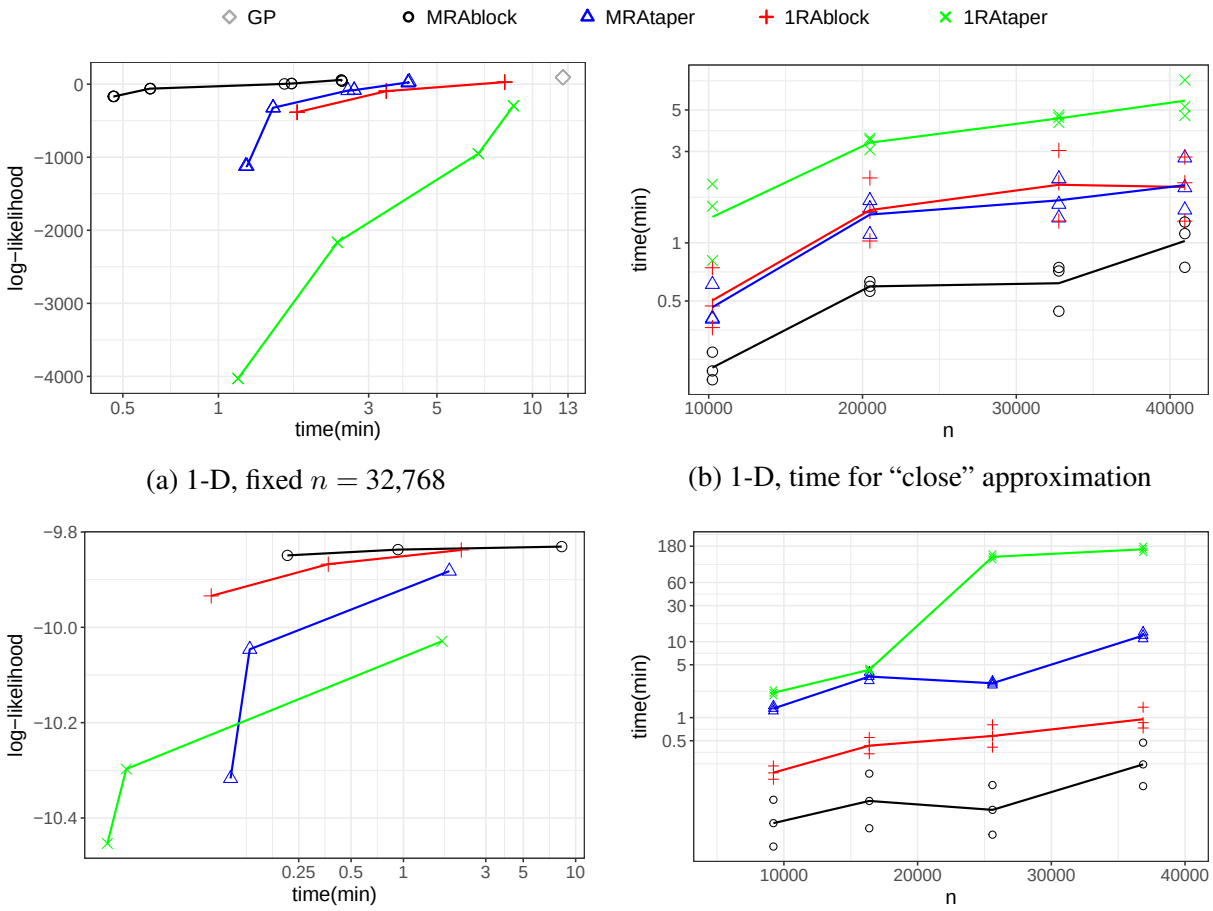
$$C_0(\mathbf{s}_1, \mathbf{s}_2) = \sigma^2 \exp(-\|\mathbf{s}_1 - \mathbf{s}_2\|/\kappa), \qquad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D}, \tag{3.15}$$

with $\sigma^2 = 0.95$ and $\kappa = 0.05$ on a one-dimensional ($\mathcal{D} = [0, 1]$) or two-dimensional ($\mathcal{D} = [0, 1]^2$) domain. We assumed a nugget or measurement-error variance of $\tau^2 = 0.05$ (i.e., $\mathbf{V}_\epsilon = 0.05\,\mathbf{I}$). Results for Matérn covariances with different range, smoothness, and variance parameters showed similar patterns as those presented below and can be found in the Supplementary Material.

All comparisons were carried out based on the log-score (i.e., the log-likelihood at the true parameter values), which is a strictly score that is uniquely maximized in expectation under the true model (e.g., Gneiting and Katzfuss, 2014). All results were averaged over five replications.

For M-RA-taper, some experimentation showed that there are general guidelines to follow in order to get a close approximation to true GP. For a true covariance function C_0 with effective range ρ, we recommend setting the M-RA taper range at resolution 0 to $d_0 = 2\rho$, and the distance between two adjacent knots at resolution 0 to be at most $\frac{2}{3}$ of ρ. For example, the covariance in (3.15) has an effective range of $\rho \approx 0.15$, and so we set $d_0 = 0.3$ and the distance between adjacent knots at resolution 0 to 0.1.

First, we simulated datasets of different sizes on an equidistant grid in one-dimensional space with $\mathcal{D} = [0, 1]$, which allowed fast simulation with the Davies-Harte algorithm and permitted the evaluation of the exact likelihood using the Durbin-Levinson algorithm for comparison (McLeod et al., 2007). For each dataset, we recorded the computation times as well as the log-scores for varying configurations of the M-RA (i.e., with different r_0, J, and M). We also considered the computation times to achieve particular levels of approximation accuracy, specifically the time required to obtain an average log-score within a difference of $0.003n$, $0.005n$, and $0.007n$ of the log-score of the true model. We then repeated the simulation study in two dimensions, $\mathcal{D} = [0, 1]^2$. As it was infeasible to compute the true log-likelihood for large n, we use the best approximation (i.e., the largest approximated log-likelihood) as the base to compare the relative performance of



(a) 1-D, fixed $n = 32{,}768$

(b) 1-D, time for "close" approximation

(c) 2-D, fixed $n = 36{,}864$

(d) 2-D, time for "close" approximation

Figure 3.3: Summary of results from the simulation study. Top row: $\mathcal{D} = [0, 1]$. Bottom row: $\mathcal{D} = [0, 1]^2$. Left column: Log-score versus computation time for different versions of the M-RA for fixed n. Right column: Computation time required to get a "close" approximation to the truth (or best approximation) for different n; lines connect the means of the three times for each model and each n. Note that the axes of time is on a log-transformed scale. Additional results can be found in the Supplementary Material.

different methods, with cut-off values of $0.008n$, $0.01n$, and $0.012n$.

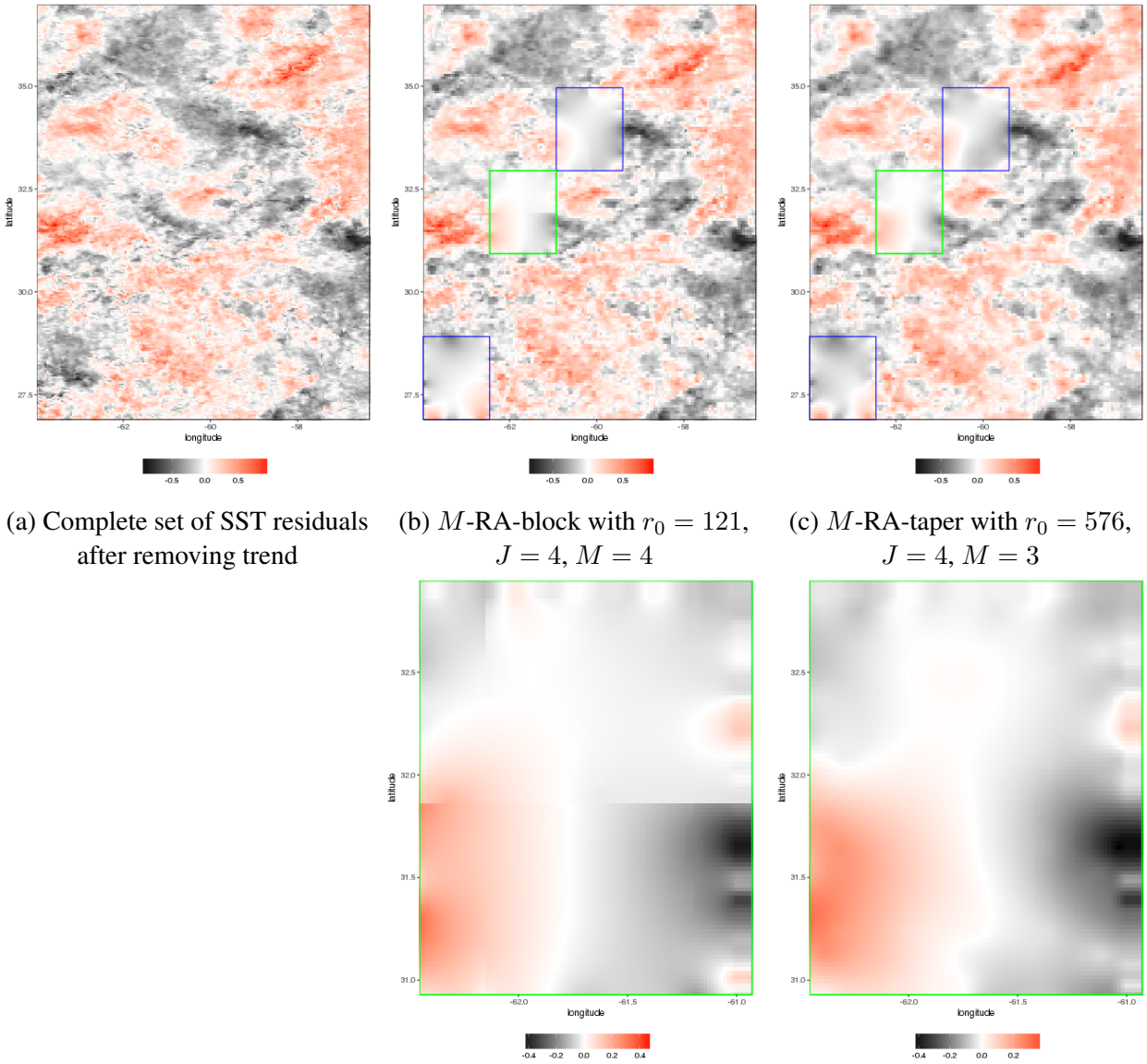
The results are summarized in Figure 3.3. The computation times scaled roughly as expected. The M-RA-block was consistently better than the other methods, while M-RA-taper and 1-RA-block performed similarly. The 1-RA-taper was not competitive.

3.5 Application

In this section, we applied the four methods from Section 3.4 to a real satellite dataset. We considered $n = 44{,}711$ Level-3 daytime sea surface temperature (SST) data from August 2016 over a region in the North Atlantic Ocean, as measured by the Moderate Resolution Imaging Spectroradiometer on board the Terra satellite. The data are freely available at `https://giovanni.gsfc.nasa.gov`. More specifically, the data (shown in Figure 3.4a) were taken to be the residuals of the SST data after removing a longitudinal and latitudinal trend. The exploratory analysis showed that an exponential covariance fit well for the data, and then all methods used were approximating the covariance in (3.15). We assumed a constant noise variance τ^2 (i.e., $\mathbf{V}_\epsilon = \tau^2\mathbf{I}$).

To compare the different approximation methods, we created five different datasets by randomly splitting the complete dataset of residuals into training data, areal test data, and random test data, each containing 78%, 12% and 10%, respectively, of the values in the full data set. The split of the complete data into training and test sets was designed to mimic the typical setting of Level-2 satellite data, with unobserved areas over which the satellite did not fly in a particular time period, and observed areas with some missing values (e.g., due to clouds). Specifically, the areal test locations were obtained by splitting the domain into $5 \times 5 = 25$ equal-area rectangles and then removing three of these rectangles at random. The remaining test locations were obtained by simple random sampling of the remaining locations.

Based on each of the five training sets and for a range of settings for each of the four approximation methods, we carried out maximum-likelihood estimation of the unknown parameters σ^2, κ, and τ^2, and obtained posterior predictive distributions at the held-out test locations. We compared the pointwise (i.e., marginal) posterior distributions obtained by the methods to the held-out test data in terms of the root mean squared prediction error (RMSPE) and the continuous ranked



(a) Complete set of SST residuals after removing trend

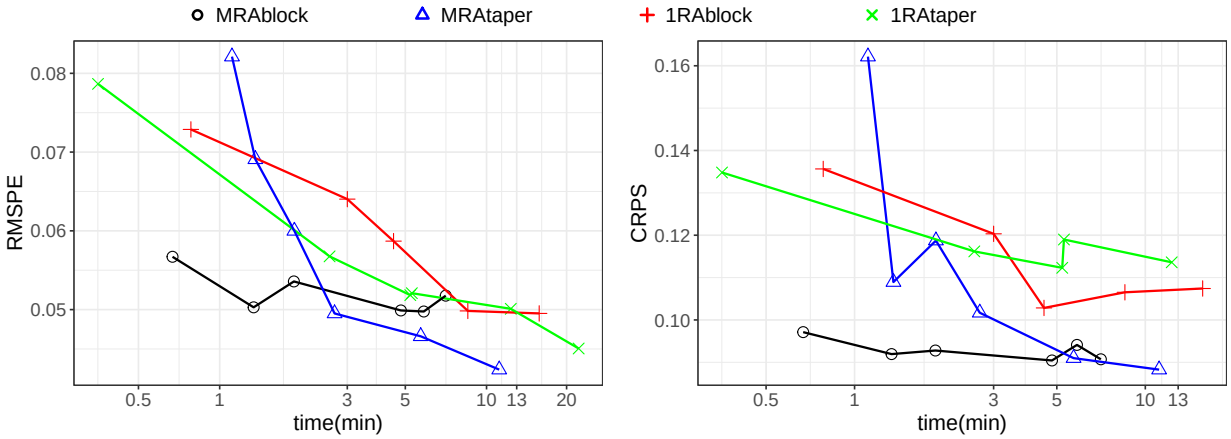
(b) M-RA-block with $r_0 = 121$, $J = 4$, $M = 4$

(c) M-RA-taper with $r_0 = 576$, $J = 4$, $M = 3$

(d) Missing-area prediction for M-RA-block

(e) Missing-area prediction for M-RA-taper

Figure 3.4: Top row: Complete dataset of sea-surface temperature, along with posterior predictive means for M-RA-taper and M-RA-block based on removing three areal test regions and additional randomly selected values. Bottom row: Zoomed-in view of the green rectangle in the upper prediction plots. Color scales are in units of degrees Celsius.



(a) RMSPE (b) CRPS

Figure 3.5: For the satellite SST data, comparison of scores (lower is better) for predictions of areal test data for different settings of the M-RA.

probability score (CRPS), which is a proper score to quantify the goodness of fit for the predictive distribution to the data (e.g., Gneiting and Katzfuss, 2014). The scores for the random test data were almost zero for all methods. The scores for the areal test data are shown in Figure 3.5 (averaged over the five datasets). In general, the scores for M-RA-taper and M-RA-block were better than those for the full-scale approximations. M-RA-taper produced some RMSPEs that were even lower than those for M-RA-block.

Maybe more important than the differences in prediction scores are the differences in the prediction plots. Figure 3.4 shows an example of the posterior means as obtained by M-RA-taper and M-RA-block, for versions of the two methods that took a similar time to run (5 to 7 minutes) and resulted in similar RMSPEs in Figure 3.5a. Despite the good approximation accuracy and low RMSPE of M-RA-block, we can see in Figure 3.4d that there are clearly visible artifacts due to discontinuities of the M-RA-block at the region boundaries (see Proposition 5), which do not appear for the continuous M-RA-taper in Figure 3.4e. Avoiding these kinds of "non-physical" artifacts is often of paramount importance to domain scientists.

3.6 Conclusions

We have proposed and studied a general approach for obtaining multi-resolution approximations of Gaussian processes (GPs) based on an orthogonal decomposition of the GP of interest into processes at multiple resolutions. We considered two specific cases of this approach: The M-RA-taper achieves sparsity and computational feasibility by applying increasingly compact tapering functions as the resolution increases, while the M-RA-block is based on a recursive splitting of the spatial domain, and assumes conditional independence between the spatial subregions at each resolution. We have provided algorithms for inference, along with computational complexity of the methods.

We have shown theoretically and numerically that both M-RA versions have useful properties and can outperform related existing approaches. The M-RA-block achieves more accurate approximations to a given covariance function for a given computation time, and its block-sparse structure allows it to deal with truly massive datasets on modern distributed computing systems. However, the M-RA-block process is discontinuous at the subregion boundaries. The M-RA-taper can be useful for real-world applications in which the true covariance function is unknown anyway, and hence it might be more important to have a "smooth" model that avoids the potential artifacts and discontinuities inherent to the M-RA-block due to its domain partitioning. The M-RA-taper's prediction accuracy can be highly competitive, especially when the effective range of the covariance model is small relative to the domain size. Also note that posterior inference involving the M-RA-taper only requires general sparse matrices, which would allow for relatively straightforward treatment of areal-averaged measurements (e.g., satellite footprints).

Next chapter will consider multivariate extensions of the methodology. Also of interest is more precise quantification of the approximation error, and a further investigation of how to choose the number of resolutions and the knots depending on the covariance to be approximated.

4. MULTI-RESOLUTION APPROXIMATION FOR MULTIVARIATE SPATIAL DATA

4.1 Introduction

Advances in remote-sensing techniques have enabled the collection of scientific data from multiple processes accurately over large spatial domain, which led to an explosion in the amount of data in all field of science, like agriculture, geology, oceanography, astronomy and meteorology. The data are mostly multivariate, with several variables observed over space. The interest of researchers is to understand the spatial dependence within and across variables. Gaussian processes are popular models for these settings, because of their flexibility and natural uncertainty quantification (e.g., Banerjee et al., 2004; Rasmussen and Williams, 2006; Cressie and Wikle, 2011). However, large data sets pose substantial challenges to spatial modeling because of the notorious curse of dimensionality. In particular, spatial modeling and prediction involve inversion of an $n \times n$ covariance matrix for data of size n with one response variable. For multivariate spatial data with p processes and n observations for each, the cross-covariance would be of size $np \times np$ and the statistical inference would be even more computationally demanding. So scalable statistical methods are needed to extract information from big spatial datasets, especially for multivariate data.

Consequently, many approximation methods have been developed to achieve computational feasibility in order to apply Gaussian processes to large datasets. Low-rank models (e.g., Higdon, 1998; Mardia et al., 1998; Wikle and Cressie, 1999; Banerjee et al., 2008; Cressie and Johannesson, 2008; Katzfuss and Cressie, 2009, 2011, 2012; Nguyen et al., 2014) approximate the spatial process based on a few basis functions, which may result in over-smoothing and fail to capture fine-scale variation (Finley et al., 2009; Stein, 2014). Sparse approximation techniques shrink the covariance of spatial locations to zero if locations are far apart, and yield a sparse covariance matrix in order to reduce the computation (Furrer et al., 2006; Kaufman et al., 2008). Vecchia's method and its extensions (e.g., Vecchia, 1988; Stein et al., 2004; Datta et al., 2016; Guinness, 2016; Katzfuss and Guinness, 2017; Katzfuss et al., 2018) induce sparsity into the precision matrix instead by

assuming conditional independence, but these methods require a careful choice of conditional sets. Katzfuss (2017) and Katzfuss and Gong (2017) introduce a multi-resolution approximation (MRA) framework, as a mixture of low-rank models and sparse approximation techniques, which allows capturing spatial structure at all scales. A special case of the MRA is given by the full-scale approximation (Snelson and Ghahramani, 2007; Sang et al., 2011; Sang and Huang, 2012). Among these approximation methods, only a small part of work focused on multivariate cases. Banerjee et al. (2008), Finley et al. (2009) and Sang et al. (2011) adopted the linear model of coregionalization (LMC) method (Wackernagel, 2003; Gelfand et al., 2004) to model multivariate dependence structure by specifying a principal component transformation matrix in a Bayesian setting. However, LMC does not account for asymmetric covariance in general, and using spatially varying weight matrices would involve too many parameters in the model. The univariate MRA can also be applied in the LMC setting to model multivariate data, but the intrinsic drawbacks of LMC, like over-parameterization, may not be easy to overcome. Apanasovich and Genton (2010) proposed modeling cross-covariances using existing covariance functions on an extended space that includes latent dimensions, which extended the univariate models to a broader scope. This leads to a more flexible way to extend the univariate MRA to the multivariate setting.

In this chapter, we introduce a multivariate MRA of Gaussian processes on an extended space, which facilitates efficient computation with a large number of spatial observations on multiple processes. It fits into the framework of general multi-scale methods (e.g. Chui, 1992; Johannesson et al., 2007; Cressie and Johannesson, 2008; Nychka et al., 2015), which have great flexibility in capturing spatial dependence while being computational feasible. Comparing to other multi-resolution models, the MRA automatically specifies the spatial basis functions, and generates the prior distributions of their weights once a covariance function is given. In this chapter, we use the latent-dimension approach with univariate covariance functions to model the cross-covariance of multivariate processes (e.g. Apanasovich and Genton, 2010; Genton and Kleiber, 2015), which fits naturally into the MRA framework. The multi-resolution structure provides great flexibility for the modeling of multivariate random fields in capturing the marginal and cross-covariance among

processes at different scales, while keeping the computation within budget.

4.2 Multivariate Gaussian process and latent dimensions

4.2.1 Basic definition

Consider a p-dimensional random process $\mathbf{Y}(\cdot) = \{y_1(\mathbf{s}), \ldots, y_p(\mathbf{s}))^T : \mathbf{s} \in \mathcal{D}\}$, on a continuous (non-gridded) domain $\mathcal{D} \subset \mathbb{R}^d$, $d \in \mathbb{N}^+$, where $y_k(\mathbf{s})$ is the k-th process at location $\mathbf{s}$. We assume that $\mathbf{Y}(\cdot) \sim GP(0, C)$ is a multivariate Gaussian process with cross-covariance function C. So the marginal distribution of each process $y_k(\cdot)$ follows a univariate Gaussian distribution. And the cross-covariance of all processes, $C(\mathbf{s}_1, \mathbf{s}_2) = cov\{\mathbf{Y}(\mathbf{s}_1), \mathbf{Y}(\mathbf{s}_2)\} = \{C_{k,l}(\mathbf{s}_1, \mathbf{s}_2)\}_{k,l=1}^p$ is composed of functions

$$C_{k,l}(\mathbf{s}_1, \mathbf{s}_2) = \text{cov}(y_k(\mathbf{s}_1), y_l(\mathbf{s}_2)), \qquad \mathbf{s}_1, \mathbf{s}_2 \in \mathbb{R}^d, \tag{4.1}$$

for $k, l = 1, \ldots, p$. Once the cross-covariance function is specified, the main goal is to make inference on the unknown parameters $\boldsymbol{\theta}$ in covariance function based on the observed data, and predict $\mathbf{Y}(\cdot)$ at a set of unobserved locations $\mathcal{S}^P$ (i.e., to get the posterior mean of $\mathbf{Y}(\mathcal{S}^P)$). For data of length n, with $n = \sum_{k=1}^p n_k$, traditional methods using the Cholesky decomposition of the resulting covariance matrix has $\mathcal{O}(n^2)$ memory complexity and $\mathcal{O}(n^3)$ time complexity , which is computationally prohibitive for $n \gg 10^4$.

Each process $y_k(\cdot)$ is associated with a vector $\boldsymbol{\xi}_k \in \mathbb{R}^q$, i.e., $\boldsymbol{\xi}_k = (\xi_{k,1}, \ldots, \xi_{k,q})^T$, $k = 1, \ldots, p$, which representing the process in the q latent dimensions. Then the observation location $\mathbf{s}$ will be denoted as $\tilde{\mathbf{s}} = (\mathbf{s}, \boldsymbol{\xi}_k) \in \mathbb{R}^{d+q}$ if process y_k is observed at location $\mathbf{s}$. Let $\boldsymbol{\xi}_{k(i)}$ denote the location in the latent dimension of the observation at $\mathbf{s}_i$. The observed location set is $\tilde{\mathcal{S}} = \{(\mathbf{s}_i, \boldsymbol{\xi}_{k(i)}), i = 1, \ldots, n\}$. And we have $\boldsymbol{\xi}_{k(i)} = \boldsymbol{\xi}_k$ if process y_k is observed at location $\mathbf{s}_i$. Under this setting, (4.1) is modeled as a covariance function with arguments in $\mathbb{R}^{d+q}$:

$$C_{kl}(\mathbf{s}_i, \mathbf{s}_j) = C((\mathbf{s}_i, \boldsymbol{\xi}_k), (\mathbf{s}_j, \boldsymbol{\xi}_l)), \qquad \mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^d, \qquad \boldsymbol{\xi}_k, \boldsymbol{\xi}_l \in \mathbb{R}^q. \tag{4.2}$$

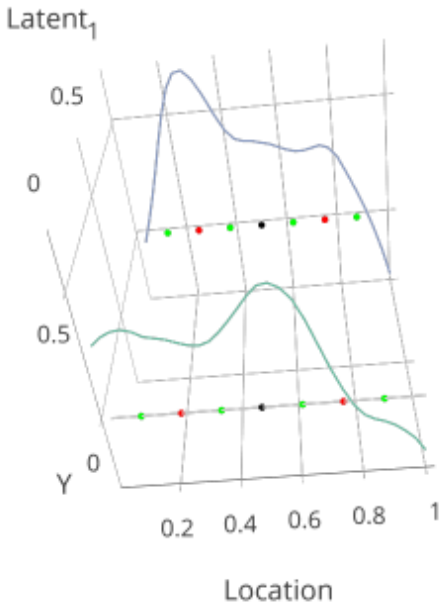


Figure 4.1: Illustration of a bivariate process in one-dimensional latent space (i.e., $d = 1, p = 1$) and knot allocation with $r_0 = 1, J = 2, M = 2$.

Consequently, the resulting covariance matrix is guaranteed to be positive definite if C is a valid covariance function. The cross-correlation between processes will be determined by the distances between the latent locations, $\boldsymbol{\delta}_{kl} = \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_l\|$. With locations $\mathbf{s}_i$ and $\mathbf{s}_j$ fixed, larger $\boldsymbol{\delta}_{ij}$s are translated to smaller cross-correlation between the k-th and l-th process.

The multivariate MRA is proposed based on the univariate MRA structure and implemented on $d + q$ dimensions, where the additional latent dimension represents the various processes to be modeled. Figure 4.1 is an illustration of a bivariate process in one-dimensional space with one latent dimension.

4.2.2 Latent locations

Generally, instead of specifying the $\boldsymbol{\xi}_k$'s, we can also treat them as parameters and estimate then with observed data. In addition, the cross-covariance with latent dimension allows modeling non-stationarity by varying parameters for different processes (e.g., for non-stationary Matérn, using $\sigma_k, \nu_k, \lambda_k$ for process y_k).

As discussed in Section 2.3.1, the parameterization of latent locations could be accomplished

using either $\boldsymbol{\xi}_k \in \mathbb{R}^q$ or the distances $\boldsymbol{\delta}_{kl} = \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_l\| \in \mathbb{R}$, or even by pre-specifying the values of $\boldsymbol{\xi}_k, k = 1, \ldots, p$. Choice of the latent locations of each process is subjective, but there are still some general guidelines to follow. For example, for covariance functions with a fixed range parameter, the relative distance of processes on latent dimension determines the cross-correlation between processes, with shorter distance indicating strong cross-correlation and large distances implying less correlated processes. Thus for bivariate processes that are known to be strongly correlated. If we fix the range parameter in its covariance function and set $\boldsymbol{\xi}_1$ to 0, we can place the other process near 0, say at 0.5 in the latent dimension. On the other hand, if all the $\boldsymbol{\xi}_k$'s are specified and fixed, we could select a larger range parameter to model highly dependent processes.

Thus both the distances between $\boldsymbol{\xi}_k$'s and the range parameter in the covariance function can be used to adjust the cross-correlation between different processes. If $\boldsymbol{\xi}_k$'s are treated as model parameters, there might be an identifiablility problem when trying to estimate $\boldsymbol{\xi}_k$ and range parameters at the same time. In this case, we could restrict the latent locations within a certain range, for instance, fixing $\boldsymbol{\xi}_1 = 0$ and $\boldsymbol{\xi}_2 = 1$, and for $k > 2$, $\boldsymbol{\xi}_k \in [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2]$. In this way the distances of different processes in the latent dimensions do not exceed 1.

4.2.3 Nonstationary covariance

The disadvantage of using a stationary univariate covariance function is that we have only one set of parameters to control the dependence for multiple processes, including smoothness, range and sill. For example, it is not possible to distinguish between the smoothing effect across the latent dimension and the spatial dimension if we have a single smoothness parameter for the covariance. In practice, spatial fields often have different ranges, and thus a nonstationary covariance is more appropriate for such cases.

According to Paciorek and Schervish (2006), isotropic correlation functions in $\mathbb{R}^d, d \in \mathbb{Z}^+$ can be extended to nonstationary by using the so-called Mahalanobis distance

$$q(\mathbf{s}_i, \mathbf{s}_j) = (\mathbf{h}'\mathbf{A}^{-1}\mathbf{h})^{1/2},$$

where $\mathbf{h} = \|\mathbf{s}_i - \mathbf{s}_j\|$ with $\mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^d$ and $\mathbf{A} = \mathbf{A}(\mathbf{s}_i) + \mathbf{A}(\mathbf{s}_j)$, and $\mathbf{A}(\mathbf{s})$ for each $\mathbf{s}$ is a positive definite $d \times d$ matrix that changes smoothly over space. For any isotropic correlation function ρ that is positive definite on $\mathbb{R}^d$ for all $d = 1, 2, \ldots$,

$$\rho_{NS}(\mathbf{s}_i, \mathbf{s}_j) = c(\mathbf{s}_i, \mathbf{s}_j)\rho(q(\mathbf{s}_i, \mathbf{s}_j)) \tag{4.3}$$

is a valid nonstationary correlation function, where

$$c(\mathbf{s}_i, \mathbf{s}_j) = |\mathbf{A}(\mathbf{s}_i)|^{1/4}|\mathbf{A}(\mathbf{s}_j)|^{1/4}|(\mathbf{A}(\mathbf{s}_i) + \mathbf{A}(\mathbf{s}_j))/2|^{-1/2}$$

is the normalization term. The isotropic Matérn covariance function introduced in Section 2.1.1 can be made nonstationary in a similar fashion. And we can even let the smoothness parameter (Stein, 2005) as well as the variance vary over space. A nonstationary Matérn covariance function corresponding to (2.2) is then given by

$$\mathcal{M}_{NS}(\mathbf{s}_i, \mathbf{s}_j) = \sigma(\mathbf{s}_i)\sigma(\mathbf{s}_j)c(\mathbf{s}_i, \mathbf{s}_j)\mathcal{M}_{(\nu(\mathbf{s}_i)+\nu(\mathbf{s}_j))/2}(q(\mathbf{s}_i, \mathbf{s}_j)), \tag{4.4}$$

where $\sigma(\mathbf{s})$ and $\nu(\mathbf{s})$ denotes the spatially varying standard deviation and smoothness parameters.

The setting of multiple processes with latent dimension is well suited to modeling nonstationarity with parameters varying over space. In particular, when the parameters depend on the latent dimension, the covariance function allows much more flexibility for modeling different processes. The Matérn covariance function in (4.4) can be then redefined as

$$\mathcal{M}_{NS}(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j) = \sigma(\boldsymbol{\xi}_k)\sigma(\boldsymbol{\xi}_l)c(\boldsymbol{\xi}_k, \boldsymbol{\xi}_l)\mathcal{M}_{(\nu(\boldsymbol{\xi}_k)+\nu(\boldsymbol{\xi}_l))/2}(q(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j)),$$

where $\boldsymbol{\xi}_k, \boldsymbol{\xi}_l$ are the latent location for process $\mathbf{y}_k$ and $\mathbf{y}_l$ observed at location $\tilde{\mathbf{s}}_i$ and $\tilde{\mathbf{s}}_j$, i.e., $\tilde{\mathbf{s}}_i = \{\mathbf{s}_i, \boldsymbol{\xi}_k\}$ and $\tilde{\mathbf{s}}_j = \{\mathbf{s}_j, \boldsymbol{\xi}_l\}$:

$$q(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j) = ((\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j)'\mathbf{A}^{-1}(\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j))^{1/2},$$

where at $\boldsymbol{\xi}_k$, $\mathbf{A}$ can be defined as a diagonal matrix with each diagonal element equal to the range parameters for different dimensions, i.e. $\{\lambda_1, \lambda_2, \ldots, \lambda_d, \lambda_k\}$ with $\lambda_1 = \lambda_2 = \ldots = \lambda_d = \lambda$. Then, for $k = l$, which means the process k is observed at both locations,

$$\text{cov}(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j) = \sigma(\boldsymbol{\xi}_k)^2 \rho\left(\frac{\|\mathbf{s}_i - \mathbf{s}_j\|}{\lambda_k}\right).$$

For the case where $\mathbf{s}_i = \mathbf{s}_j$, which means two processes $\mathbf{y}_l$ and $\mathbf{y}_k$ are observed at the same location $\mathbf{s}_i$,

$$\text{cov}(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j) = \sigma(\boldsymbol{\xi}_k)\sigma(\boldsymbol{\xi}_l) \rho\left(\frac{\|\boldsymbol{\xi}_k - \boldsymbol{\xi}_l\|}{\lambda}\right).$$

4.3 Multivariate MRA

4.3.1 Domain partitioning and knot allocation

With latent dimensions for multiple processes, the extended domain $\tilde{\mathcal{D}} \in \mathbb{R}^{d+q}$ will be the cross-product between the original spatial domain $\mathcal{D} \in \mathbb{R}^d$ and the latent process domain $\mathbb{R}^q$ as indicated in (4.2). To define MRA, we need to partition the domain $\tilde{\mathcal{D}}$. For the spatial domain $\mathcal{D}$, partitioning is done recursively at each resolution as in univariate MRA. As discussed in Chapter 3, to de fine the M-RA, we need partition the domain $\mathcal{D}$ recursively, where each of the J regions, $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_J$ is partitioned into J smaller subregions again, up to M-th resolution:

$$\mathcal{D}_{j_1,\ldots,j_{m-1}} = \cup_{j_m=1,\ldots,J} \mathcal{D}_{j_1,\ldots,j_m}, \qquad j_1, \ldots, j_{m-1} = 1, \ldots, J; m = 1, \ldots, M.$$

To achieve computational efficiency, the subregions are assumed to be independent after the partition. If the observed locations are uniformly distributed over the spatial domain $\mathcal{D}$, the partitions can be done by recursively dividing each region into J subregions equally. For irregularly spaced locations, we need more complicated partitioning in order to get proper inference within computation budget. For latent space, the grouping of processes has to be carefully designed to account for the cross dependence.

We need to choose at which resolution that partitioning happens. If the processes are split

at lower resolution, at resolution 1 for example, then cross-covariance will only get captured for large-scale, smooth dependence at resolution 0. Once we split, the two processes will be modeled independently with a regular MRA within each of their spatial domains, which will greatly reduce the computation. Therefore, choosing the resolution at which the splitting happens is a tradeoff between approximation quality of cross dependence and the overall computation efficiency. For processes with higher cross-correlation, it is better to postpone the partition to a later stage to retain quality of approximation on the dependence across processes. At the same time, because of the flexibility of MRA, we could always adjust the number of knots within each process to reduce computation load to achieve fast inference.

To model multiple processes especially when $p > 2$, the placement of processes on the latent dimension also matters to determine the split and grouping of processes. A general guidance is to put closely correlated process near each other according to domain knowledge of the processes. As indicated in Section 4.2, the distance δ_{kl} and the range parameter of covariance function determine the cross-correlation between processes y_k and y_l, and therefore, highly correlated processes are suggested to be placed close to each other. As mentioned, domain knowledge on real applications will be helpful to determine the placement and partitioning of processes as well. Suppose the processes $y_1, \ldots, y_p$ are ordered in an one-dimensional latent space i.e., q=1 (ordering of process location for q >1 is tricky) according to prior knowledge, we can partition them into G groups $g_1, \ldots, g_G$ in latent space along with the recursive partitioning in spatial domain. The structure of MRA allows grouping and partition of multiple resolutions, which provides great flexibility on the tradeoff between approximation accuracy and efficiency. Again, partitioning of processes at lower resolutions facilitates the computation but leads to ignoring of small-scale cross-covariance, whereas later splitting will model the cross-covariance better at the cost of more computation.

Along with the partitioning of the latent dimension, we need a multi-resolution set of knots in the spatial domain. Similar schemes as in Katzfuss (2017) or Katzfuss and Gong (2017) can be applied here, by assuming the number of knots increases with resolution by a factor of J. The set of knots, $\mathcal{Q}_m = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_M\}$ is chosen such that, $\mathcal{Q}_m = \{\mathbf{q}_{m,1}, \ldots, \mathbf{q}_{m,r_m}\}$, is a set of r_m knots,

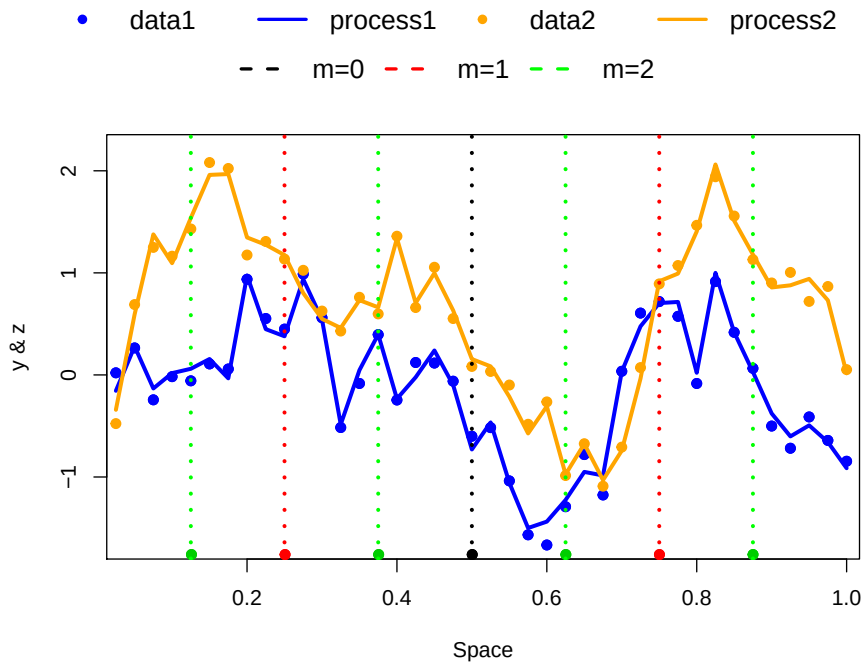


Figure 4.2: Illustration of domain partitioning for a bivariate process with $\mathcal{D} = [0, 1]$, $r_0 = 1$, $J = 2$, $M = 2$.

with $\mathbf{q}_{m,i} \in \mathbb{R}^{d+q}$. To achieve good approximations, it is recommended to choose small M and J and large r_0 as long as the computational resources allow, under the constraint that $r_0 J^M \geq n$.

To decide the allocation of knots, an exploratory analysis shows that, under the same computation budget, assigning knots between processes in latent dimension does not show obvious advantage in the approximation but potentially causes problem. For example, it is not clear how to specify parameters like the smoothness for knots in-between processes in the latent dimensions. For the sake of simplicity, in the latent dimension, knots are only placed exactly on each process' latent position $\boldsymbol{\xi}_k$, which also makes the specification of cross-covariance function on knots locations more straightforward. An illustration of bivariate process with such a set of knots is given in Figure 4.2. In this plot, it shows a bivariate Gaussian process $\mathbf{Y} = (y_1, y_2)$ along with observed data $\mathbf{Z} = (z_1, z_2)$. We used the same set of knots iteratively for both process with $r_0 = 1$, $J = 2$, $M = 2$ and the same partition on spatial domain $\mathcal{D} = [0, 1]$ at each latent location of the processes. Note that knots for different processes can be also be different in spatial dimension.

4.3.2 Definition of multivariate MRA

Similarly as in Chapter 3, we can define a multivariate MRA. For a given $M \in \mathbb{N}$, the M-RA of a multivariate process $\mathbf{Y}_0(\cdot) \sim GP(0, C_0)$ based on a set of knots $\mathcal{Q} = \{\mathcal{Q}_0, \ldots, \mathcal{Q}_M\}$ and a set of modulating functions $\mathcal{T} = \{\mathcal{T}_0, \ldots, \mathcal{T}_M\}$, is given by

$$\mathbf{Y}_M(\cdot) = \sum_{m=0}^{M} \widetilde{\tau}_m(\cdot) = \sum_{m=0}^{M} \mathbf{b}_m(\mathbf{s})' \boldsymbol{\eta}_m, \tag{4.5}$$

where $\widetilde{\tau}_m(\cdot) := \widetilde{\delta}_m^{(m)}(\cdot)$ and $\boldsymbol{\eta}_m \overset{ind.}{\sim} \mathcal{N}_{r_m}(\mathbf{0}, \boldsymbol{\Lambda}_m^{-1})$ for $m = 0, \ldots, M$; $\widetilde{\delta}_0(\cdot) := [\mathbf{Y}_0]_{[0]}(\cdot) \sim GP(0, v_0)$ with $[\mathbf{Y}_0]_{[0]}(\cdot)$ being the modulated $\mathbf{Y}_0(\cdot) = \mathbf{Y}(\cdot)$ at 0-th resolution, $v_0 = [C_0]_{[0]}$, which is the corresponding modulated covariance C_0; Similarly, the modulated residuals at m-th resolution, $\widetilde{\delta}_m(\cdot) = [\widetilde{\delta}_{m-1} - \widetilde{\tau}_{m-1}]_{[m]}(\cdot) \sim GP(0, v_m)$ for $m = 1, \ldots, M$; and

$$\begin{aligned}
\mathbf{b}_m(\mathbf{s})' &:= v_m(\mathbf{s}, \mathcal{Q}_m), \quad \mathbf{s} \in \mathcal{D}, \quad m = 0, \ldots, M, \\
\boldsymbol{\Lambda}_m &:= v_m(\mathcal{Q}_m, \mathcal{Q}_m), \quad m = 0, \ldots, M, \\
v_{m+1}(\mathbf{s}_1, \mathbf{s}_2) &:= \big(v_m(\mathbf{s}_1, \mathbf{s}_2) - \mathbf{b}_m(\mathbf{s}_1)' \boldsymbol{\Lambda}_m^{-1} \mathbf{b}_m(\mathbf{s}_2)\big) \cdot \mathcal{T}_{m+1}(\mathbf{s}_1, \mathbf{s}_2), \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D}, m = 0, \ldots, M-1.
\end{aligned} \tag{4.6}$$

For ease of notation, we often stack the basis functions as $\mathbf{b}(\cdot) := \big(\mathbf{b}_0(\cdot)', \ldots, \mathbf{b}_M(\cdot)'\big)'$ and the corresponding coefficients, $\boldsymbol{\eta} := \big(\boldsymbol{\eta}_0', \ldots, \boldsymbol{\eta}_M'\big)'$, so that

$$\mathbf{Y}_M(\cdot) = \mathbf{b}(\cdot)' \boldsymbol{\eta}, \qquad \text{where } \boldsymbol{\eta} \sim \mathcal{N}_r(\mathbf{0}, \boldsymbol{\Lambda}^{-1}), \tag{4.7}$$

with $\boldsymbol{\Lambda} := blockdiag(\boldsymbol{\Lambda}_0, \ldots, \boldsymbol{\Lambda}_M)$ and $r = \sum_{m=0}^{M} r_m$. And the covariance of $\mathbf{Y}_M(\cdot) \sim GP(0, C_M)$ is given by

$$C_M(\mathbf{s}_1, \mathbf{s}_2) = \sum_{m=0}^{M} v_m(\mathbf{s}_1, \mathcal{Q}_m) v_m(\mathcal{Q}_m, \mathcal{Q}_m)^{-1} v_m(\mathcal{Q}_m, \mathbf{s}_2), \quad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D},$$

where v_m is defined in (4.6).

4.3.3 Computational complexity

The computational complexity of univariate MRA applies to the multivariate cases. Because the multivariate MRA is essentially applying MRA on an extended space, with latent dimensions added to spatial dimensions. The increase in dimensions does not affect its computational complexity directly, except that we assign knots for each process so that the total number of knots at each resolution are increased by a factor of p, the total number of processes.

As discussed in Section 3.3.2, we have $(\mathbf{\Lambda}_m)_{i,j} = 0$ if $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{q}_{m,j}) = 0$, i.e., $\mathbf{q}_{m,i}$ and $\mathbf{q}_{m,j}$ are from different subregions of resolution m. And thus $\mathbf{\Lambda}_m$ is a block-diagonal matrix with diagonal blocks of roughly $r_0 \times r_0$. It is well known that for block-diagonal matrix $\mathbf{\Lambda}_k$, the inverse $\mathbf{\Lambda}_k^{-1}$ of it has the same block-diagonal structure as $\mathbf{\Lambda}_k$, and so the prior calculations involving $\mathbf{\Lambda}_k^{-1}$ can be carried out at $\mathcal{O}({r_0}^3)$. Iterating over $k = 1, \ldots, m$ and $l = 1, \ldots, k$ will result in a $\mathcal{O}(M^2{r_0}^3)$. Note that r_0 here is the number of knots for all processes.

For the posterior covariance matrix, we have from Section 3.3.1.2 that $(\widetilde{\mathbf{\Lambda}}_{m,l})_{i,j} = 0$ if $\nexists \mathbf{s} \in \mathcal{D}$ such that $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{s}) \neq 0$ and $\mathcal{T}_l(\mathbf{q}_{l,j}, \mathbf{s}) \neq 0$, and so the block in $\widetilde{\mathbf{\Lambda}}$ corresponding to regions $\mathcal{D}_{i_1,\ldots,i_m}$ and $\mathcal{D}_{j_1,\ldots,j_m}$ is zero if the regions do not overlap (i.e., the subregions within one resolution are exclusive to each other as illustrated in Figure 3.1b and Figure 4.2). The Cholesky factor of this particular block-sparse structure has zero fill-in, and can thus be carried out very rapidly. Following the domain partition scheme discussed in Section 4.3.1, there are $\sum_{m=1}^{M} J^m$ subregions in total, with each having a computation complexity of $\mathcal{O}(M^2{r_0}^3)$. As calculating the $\widetilde{\mathbf{\Lambda}}$ matrices dominating the computation, the final computation complexity of MRA is boiled down to $\mathcal{O}(J^M M^2{r_0}^3) = \mathcal{O}(nM{r_0}^2)$ with $n = r_0 J^M$. A comparison of the complexity between direct Gaussian process inference, FSA and MRA is summarized in Table 4.1.

4.4 Simulation study

In this section, we simulated data from a Gaussian process and compared the performance of the multivariate MRA with the full-scale approximation (FSA), which is equivalent to the MRA with only one resolution. The true underlying Gaussian process was assumed to have mean zero

Table 4.1: Comparison of computational complexity

	Time	**Memory**
GPs	n^3	n^2
FSA	$nr_0{}^2$	nr_0
MRA	$n(Mr_0)^2$	nMr_0

and Matérn covariance function with a nugget on $\mathbb{R}^{d+q}$,

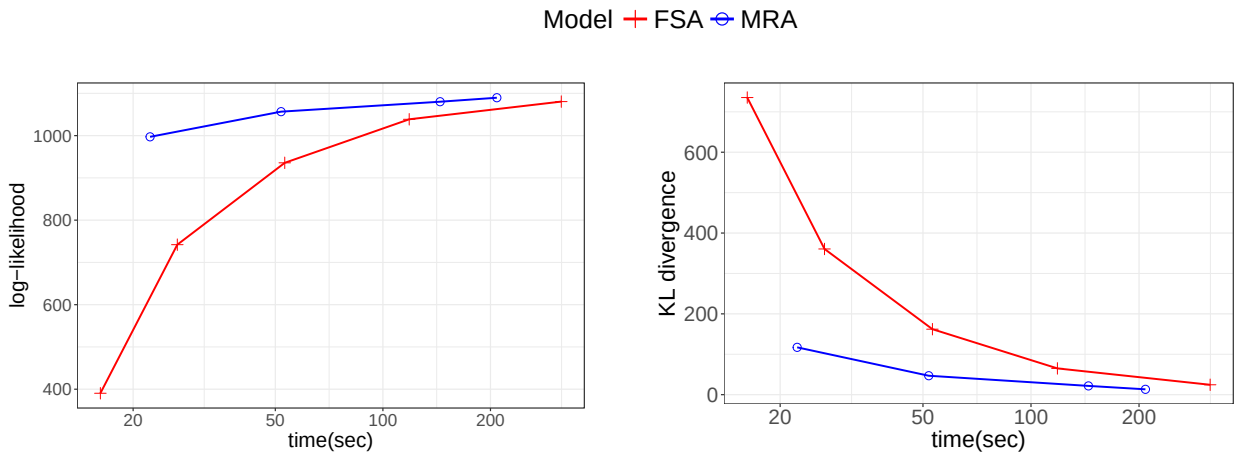
$$C_0(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j) = 0.95M_\nu(|\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j|/\lambda) + 0.05I(\tilde{\mathbf{s}}_i = \tilde{\mathbf{s}}_j), \qquad \tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j \in \mathbb{R}^{d+q}, \tag{4.8}$$

where λ is the range parameter, ν is the smoothness parameter of Matérn covariance function and $I(\cdot)$ is the indicator function. The data was simulated on an one-dimensional equidistant grid for each process. For simplicity, we use $q = 1$ and $\boldsymbol{\xi}_1 = 0, \boldsymbol{\xi}_2 = 1$. Under this setting, we simulated bivariate data with different parameter settings and recorded the log-likelihood as well as its computation time for MRA and FSA with different r_0, J and M (for FSA, $M = 1$). All results are averaged over five replications. In Figure 4.3b and Figure 4.3d the Kullback-Leibler (KL) divergence is calculated for a dataset of size $n = 16384$ as shown for model performance comparison, with KL divergence given by:

$$\text{KL}(\mathcal{L}_{FSA}, \mathcal{L}_{MRA}) = \frac{1}{2}(\log(\det({\Sigma_{FSA}}^{-1}\Sigma_{MRA})) + \text{tr}({\Sigma_{FSA}}^{-1}\Sigma_{MRA}) - n),$$

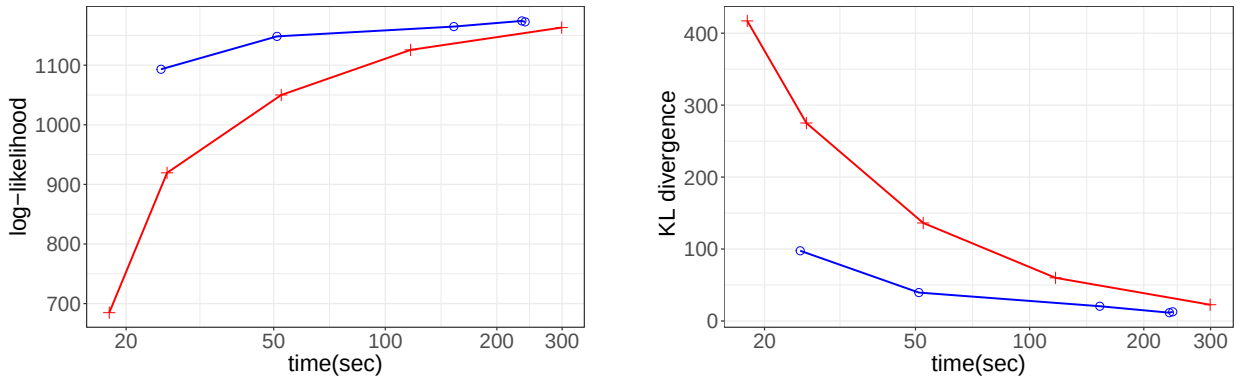
where Σ_{FSA} and Σ_{MSA} indicate the approximated covariance matrix from FSA and MRA respectively.

The computation times scaled roughly as expected. The multivariate MRA consistently performs better than FSA in terms of the log-likelihood approximation as well as the KL divergence. We also compared different levels of cross-dependence by changing the range parameter of the Matérn covariance function. Because there is an interaction between smoothness parameter and range parameters, we compared the scenario of different range parameters while having smooth-



(a) Log-likelihood approximation of bivariate process with $\nu = 1.5, \lambda = 0.2$

(b) KL divergence approximation of bivariate process with $\nu = 1.5, \lambda = 0.2$



(c) Log-likelihood approximation of bivariate process with $\nu = 1.5, \lambda = 0.5$

(d) KL divergence approximation of bivariate process with $\nu = 1.5, \lambda = 0.5$

ness fixed at 1.5. It shows no obvious difference in the comparison pattern of FSA and MRA.

5. CONCLUSIONS

Gaussian processes have been widely used in spatial statistics but face tremendous computation challenges for big datasets. This dissertation studied methods of approximating Gaussian processes for modeling big spatial data, with a particular focus on computational efficiency.

In Chapter 3, we proposed and studied a general approach for obtaining multi-resolution approximations of Gaussian processes based on an orthogonal decomposition of the GP of interest into processes at multiple resolutions. We considered two specific cases of this approach: The MRA-taper achieves sparsity and computational feasibility by applying increasingly compact tapering functions as the resolution increases. The MRA-block is based on a recursive splitting of the spatial domain, and assumes conditional independence between the spatial subregions at each resolution. The MRA-block produces more accurate approximations to a given covariance function within a given time budget, and its block-sparse structure allows it to deal with truly massive datasets on modern distributed computing systems. However, the M-RA-block process is discontinuous at the subregion boundaries. This is why the MRA-taper can be useful for real-world applications in which the true covariance function is unknown anyway, and hence it might be more important to have a 'smooth' model that avoids the potential artifacts and discontinuities inherent to the MRA-block due its domain partitioning.

In Chapter 4, the MRA is extended to multivariate spatial fields consisting of several different variables observed over space. The particular focus of that setting is to take the cross-dependence into account and make the computation feasible when the number of observations is very large. In this extension, we use latent dimensions with univariate covariance functions to model the cross-covariance of multivariate process, which fits naturally into the MRA framework. Basically, we introduced a multivariate MRA of Gaussian processes on latent dimension that could facilitate efficient computation for large multivariate spatial data. Within the framework of multi-scale structure, it allows modeling non-stationary covariance in a natural way with spatial by varying parameters for different processes. The multi-resolution structure provides great flexibility for the modeling

of multivariate random fields, to capture the marginal and cross-covariance among processes at different scales.

Future research on the multivariate MRA will include a real application and more extensive simulations with larger data sizes. We plan to explore nonstationary covariance structure with multivariate MRA, where we could let the location on latent dimension vary as a parameter as well. Also of interest is a more precise quantification of the approximation error, and a further investigation of how to choose the number of resolutions and the knots depending on the covariance to be approximated.

REFERENCES

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., and O'Neil, M. (2016). Fast direct methods for Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):252–265.

Apanasovich, T. V. and Genton, M. G. (2010). Cross-covariance functions for multivariate random fields based on latent dimensions. *Biometrika*, 97(1):15–30.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2004). *Hierarchical modeling and analysis for spatial data*. Chapman & Hall.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(4):825–848.

Chui, C. (1992). *An Introduction to Wavelets*. Academic Press.

Cressie, N. (1993). *Statistics for Spatial Data, revised edition*. John Wiley & Sons, New York, NY.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(1):209–226.

Cressie, N. and Wikle, C. K. (2011). *Statistics for Spatio-Temporal Data*. Wiley, Hoboken, NJ.

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets. *Journal of the American Statistical Association*, 111(514):800–812.

Erisman, A. M. and Tinney, W. F. (1975). On computing certain elements of the inverse of a sparse matrix. *Communications of the ACM*, 18(3):177–179.

Finley, A. O., Sang, H., Banerjee, S., and Gelfand, A. E. (2009). Improving the performance of predictive process modeling for large datasets. *Computational Statistics & Data Analysis*, 53(8):2873–2884.

Furrer, R., Genton, M. G., and Nychka, D. W. (2006). Covariance tapering for interpolation of

large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523.

Gelfand, A., Diggle, P., Guttorp, P., and Fuentes, M., editors (2010). *Handbook of Spatial Statistics*. CRC Press.

Gelfand, A. E., Schmidt, A. M., Banerjee, S., and Sirmans, C. F. (2004). Nonstationary Multivariate Process Modeling through Spatially Varying Coregionalization. *Test*, 13(2):263–312.

Genton, M. G. and Kleiber, W. (2015). Cross-Covariance Functions for Multivariate Geostatistics. *Statistical Science*, 30(2):147–163.

Gneiting, T. (2002). Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83(2):493–508.

Gneiting, T. and Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151.

Guinness, J. (2016). Permutation and grouping methods for sharpening Gaussian process approximation. *arXiv:1609.05372*.

Heaton, M. J., Datta, A., Finley, A., Furrer, R., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2017). Methods for analyzing large spatial data: A review and comparison. *arXiv:1710.05013*.

Helterbrand, J. D. and Cressie, N. (1994). Universal cokriging under intrinsic coregionalization. *Mathematical Geology*.

Higdon, D. (1998). A process-convolution approach to modelling temperatures in the North Atlantic Ocean. *Environmental and Ecological Statistics*, 5(2):173–190.

Johannesson, G., Cressie, N., and Huang, H.-C. (2007). Dynamic multi-resolution spatial models. *Environmental and Ecological Statistics*, 14(1):5–25.

Kanter, M. (1997). Unimodal spectral windows. *Statistics & Probability Letters*, 34(4):403–411.

Katzfuss, M. (2013). Bayesian nonstationary spatial modeling for very large datasets. *Environmetrics*, 24(3):189–200.

Katzfuss, M. (2017). A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association*, 112(517):201–214.

Katzfuss, M. and Cressie, N. (2009). Maximum likelihood estimation of covariance parameters in the spatial-random-effects model. In *Proceedings of the Joint Statistical Meetings*, pages 3378–3390, Alexandria, VA. American Statistical Association.

Katzfuss, M. and Cressie, N. (2011). Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets. *Journal of Time Series Analysis*, 32(4):430–446.

Katzfuss, M. and Cressie, N. (2012). Bayesian hierarchical spatio-temporal smoothing for very large datasets. *Environmetrics*, 23(1):94–107.

Katzfuss, M. and Gong, W. (2017). Multi-resolution approximations of Gaussian processes for large spatial datasets. *arXiv:1710.08976*, pages 1–21.

Katzfuss, M. and Guinness, J. (2017). A general framework for Vecchia approximations of Gaussian processes. *arXiv:1708.06302*.

Katzfuss, M., Guinness, J., and Gong, W. (2018). Vecchia approximations of Gaussian-process predictions. pages 1–22.

Katzfuss, M. and Hammerling, D. (2017). Parallel inference for massive distributed spatial data using low-rank models. *Statistics and Computing*, 27(2):363–375.

Kaufman, C. G., Schervish, M., and Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555.

Li, S., Ahmed, S., Klimeck, G., and Darve, E. (2008). Computing entries of the inverse of a sparse matrix using the FIND algorithm. *Journal of Computational Physics*, 227(22):9408–9427.

Lin, L., Yang, C., Meza, J., Lu, J., Ying, L., and Weinan, E. (2011). SelInv - An algorithm for selected inversion of a sparse symmetric matrix. *ACM Transactions on Mathematical Software*, 37(4):40.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society, Series B*, 73(4):423–498.

Mardia, K. and Goodall, C. (1993). Spatial-temporal analysis of multivariate environmental mon-

itoring data. *Multivariate environmental statistics*, 86(347).

Mardia, K., Goodall, C., Redfern, E., and Alonso, F. (1998). The kriged Kalman filter. *Test*, 7(2):217–282.

McLeod, A. I., Yu, H., and Krougly, Z. (2007). Algorithms for linear time series analysis: With R package. *Journal of Statistical Software*, 23(5).

Nguyen, H., Katzfuss, M., Cressie, N., and Braverman, A. (2014). Spatio-temporal data fusion for very large remote sensing datasets. *Technometrics*, 56(2):174–185.

Nychka, D. W., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. R. (2015). A multi-resolution Gaussian process model for the analysis of large spatial data sets. *Journal of Computational and Graphical Statistics*, 24(2):579–599.

Paciorek, C. and Schervish, M. (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics*, 17(5):483–506.

Quiñonero-Candela, J. and Rasmussen, C. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Sang, H. and Huang, J. Z. (2012). A full scale approximation of covariance functions. *Journal of the Royal Statistical Society, Series B*, 74(1):111–132.

Sang, H., Jun, M., and Huang, J. Z. (2011). Covariance approximation for large multivariate spatial datasets with an application to multiple climate model errors. *Annals of Applied Statistics*, 5(4):2519–2548.

Snelson, E. and Ghahramani, Z. (2007). Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics 11 (AISTATS)*.

Stein, M. L. (2005). Nonstationary spatial covariance functions. *Technical Report No. 21, University of Chicago*.

Stein, M. L. (2011). When does the screening effect hold? *The Annals of Statistics*, 39(6):2795–2819.

Stein, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19.

Stein, M. L., Chi, Z., and Welty, L. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B*, 66(2):275–296.

Vaidya, P. M. (1989). An O(n log n) algorithm for the all-nearest-neighbors problem. *Discrete & Computational Geometry*, 4(2):101–115.

Vecchia, A. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society, Series B*, 50(2):297–312.

Wackernagel, H. (1995). *Multivariate geostatistics: an introduction with applications*.

Wackernagel, H. (2003). *Multivariate geostatistics: an introduction with applications*. Springer, New York, NY.

Wikle, C. K. and Cressie, N. (1999). A dimension-reduced approach to space-time Kalman filtering. *Biometrika*, 86(4):815–829.

APPENDIX A

SUPPLEMENTAL MATERIAL FOR CHAPTER 3

A.1 Proofs

In this section, we provide proofs for the propositions stated throughout the manuscript. We also state and prove three lemmas that are used in the proofs of the propositions.

Proof of Proposition 1. From (3.6), we have $y_M(\cdot) = \mathbf{b}(\cdot)'\boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}_r(\mathbf{0}, \boldsymbol{\Lambda}^{-1})$ and $\mathbf{b}(\cdot)$ is a vector of deterministic functions (for given C_0, $\mathcal{Q}$, and $\mathcal{T}$). Hence, it is trivial to show that $y_M(\cdot)$ is a Gaussian process with mean zero. The covariance function is derived by combining the expression for $y_M(\cdot)$ on the right-hand side of (3.4) with the equations in (3.5). □

LEMMA 1 (Exact predictive process). *The predictive process is exact at any knot location; that is, if $x^{(m)}(\cdot)$ is the predictive process of $x(\cdot) \sim GP(0, C)$ based on knots $\mathcal{Q}_m$ (see Definition 1), and $\mathbf{s}_1 \in \mathcal{Q}_m$ (or $\mathbf{s}_2 \in \mathcal{Q}_m$), then*

$$cov\big(x^{(m)}(\mathbf{s}_1), x^{(m)}(\mathbf{s}_2)\big) = C(\mathbf{s}_1, \mathbf{s}_2).$$

Proof of Lemma 1. By the law of total covariance, we have

$$\begin{aligned} cov\big(x^{(m)}(\mathbf{s}_1), x^{(m)}(\mathbf{s}_2)\big) &= cov\big(E(x(\mathbf{s}_1)|\mathbf{x}(\mathcal{Q}_m), E(x(\mathbf{s}_2)|\mathbf{x}(\mathcal{Q}_m)\big) \\ &= cov\big(x(\mathbf{s}_1), x(\mathbf{s}_2)\big) - E\big(cov(x(\mathbf{s}_1), x(\mathbf{s}_2)|\mathbf{x}(\mathcal{Q}_m))\big) = C(\mathbf{s}_1, \mathbf{s}_2), \end{aligned}$$

because $cov(x(\mathbf{s}_1), x(\mathbf{s}_2)|\mathbf{x}(\mathcal{Q}_m)) = 0$ if $\mathbf{s}_1 \in \mathcal{Q}_m$ (or $\mathbf{s}_2 \in \mathcal{Q}_m$). □

Proof of Proposition 2. The proof will be carried out by induction. For $l = 1$, we have $v_{m+1}(\mathbf{q}, \mathbf{s}) = \big(v_m(\mathbf{q}, \mathbf{s}) - cov(\widetilde{\tau}(\mathbf{q}), \widetilde{\tau}(\mathbf{s}))\big)\mathcal{T}_{m+1}(\mathbf{q}, \mathbf{s}) = 0$, because using Lemma 1, we can see that $cov\big(\widetilde{\tau}(\mathbf{q}), \widetilde{\tau}(\mathbf{s})\big) = cov\big(\widetilde{\delta}^{(m)}(\mathbf{q}), \widetilde{\delta}^{(m)}(\mathbf{s})\big) = cov\big(\widetilde{\delta}(\mathbf{q}), \widetilde{\delta}(\mathbf{s})\big) = v_m(\mathbf{q}, \mathbf{s})$. For $l > 1$, assuming

that $v_{m+l-1}(\mathbf{q},\mathbf{s}) = 0$, we have

$$v_{m+l}(\mathbf{q},\mathbf{s}) = \big(v_{m+l-1}(\mathbf{q},\mathbf{s}) - \mathbf{b}_{m+l-1}(\mathbf{q})'\mathbf{\Lambda}_{m+l-1}^{-1}\mathbf{b}_{m+l-1}(\mathbf{s})\big)\cdot\mathcal{T}_{m+l}(\mathbf{q},\mathbf{s}) = 0,$$

because $\mathbf{b}_{m+l-1}(\mathbf{q}) = v_{m+l-1}(\mathbf{q},\mathcal{Q}_{m+l-1}) = \mathbf{0}$. □

LEMMA 2 (M-RA covariance at knot location $\mathbf{s}$). *If* $\mathbf{s}_1 \in \mathcal{Q}$, *then*

$$C_M(\mathbf{s}_1,\mathbf{s}_2) = \textstyle\sum_{m=0}^{M-1} v_m(\mathbf{s}_1,\mathcal{Q}_m)v_m(\mathcal{Q}_m,\mathcal{Q}_m)^{-1}v_m(\mathcal{Q}_m,\mathbf{s}_2) + v_M(\mathbf{s}_1,\mathbf{s}_2), \quad \mathbf{s}_2 \in \mathcal{D}.$$

Proof of Lemma 2. In the expression for C_M in Proposition 1, we have $v_M(\mathbf{s}_1,\mathcal{Q}_M)v_M(\mathcal{Q}_M,\mathcal{Q}_M)^{-1}v_M(\mathcal{Q}_M,\mathbf{s}_2) = v_M(\mathbf{s}_1,\mathbf{s}_2)$ for $\mathbf{s}_1 \in \mathcal{Q}$. This follows from Lemma 1 if $\mathbf{s}_1 \in \mathcal{Q}_M$, and from Proposition 2 for $\mathbf{s}_1 \in \mathcal{Q}_m$ for $m < M$ (because then both sides of the equation are zero). □

Proof of Proposition 3. Because $\mathcal{T}_m(\mathbf{s},\mathbf{s}) = 1$ for all $m = 0,1,\ldots,M$, we have $v_0(\mathbf{s},\mathbf{s}) = C_0(\mathbf{s},\mathbf{s})$ and $v_{m+1}(\mathbf{s},\mathbf{s}) = v_m(\mathbf{s},\mathbf{s}) - \mathbf{b}_m(\mathbf{s})'\mathbf{\Lambda}_m^{-1}\mathbf{b}_m(\mathbf{s})$ for $m = 1,\ldots,M$. Thus, we can write $v_M(\mathbf{s},\mathbf{s}) = C_0(\mathbf{s},\mathbf{s}) - \sum_{m=0}^{M-1}\mathbf{b}_m(\mathbf{s})'\mathbf{\Lambda}_m^{-1}\mathbf{b}_m(\mathbf{s})$, and using Lemma 2, we have

$$C_M(\mathbf{s},\mathbf{s}) = \textstyle\sum_{m=0}^{M-1}\mathbf{b}_m(\mathbf{s})'\mathbf{\Lambda}_m^{-1}\mathbf{b}_m(\mathbf{s}) + C_0(\mathbf{s},\mathbf{s}) - \sum_{m=0}^{M-1}\mathbf{b}_m(\mathbf{s})'\mathbf{\Lambda}_m^{-1}\mathbf{b}_m(\mathbf{s}) = C_0(\mathbf{s},\mathbf{s}).$$

□

Proof of Proposition 4. First, note that realizations of $y_0(\cdot)$ are p times differentiable at $\mathbf{s}$ if and only if $C_{0,\mathbf{s}}(\mathbf{h}) := C_0(\mathbf{s},\mathbf{s}+\mathbf{h})$ is $2p$ times differentiable at the origin ($2p$DO).

By Lemma 2, we have $C_{M,\mathbf{s}}(\mathbf{h}) := C_M(\mathbf{s},\mathbf{s}+\mathbf{h}) = \sum_{m=0}^{M-1} f_m(\mathbf{s},\mathbf{s}+\mathbf{h}) + v_M(\mathbf{s},\mathbf{s}+\mathbf{h})$, where $f_m(\mathbf{s}_1,\mathbf{s}_2) := \sum_{j=1}^{r_m} a_{m,j}(\mathbf{s}_1)v_m(\mathbf{q}_{m,j},\mathbf{s}_2)$, and $a_{m,j}(\mathbf{s})$ is the j-th element of the vector $\mathbf{a}_m(\mathbf{s}) = v_m(\mathcal{Q}_m,\mathcal{Q}_m)^{-1}v_m(\mathcal{Q}_m,\mathbf{s})$. We now show by induction for $m = 0,\ldots,M-1$ that

$$\begin{aligned}&v_{m,\mathbf{q},\mathbf{s}}(\mathbf{h}) := v_m(\mathbf{q},\mathbf{s}+\mathbf{h}) \text{ (for any } \mathbf{q}\in\mathcal{Q}) \text{ and } f_{m,\mathbf{s}}(\mathbf{h}) := f_m(\mathbf{s},\mathbf{s}+\mathbf{h}) \text{ are at least}\\ &2p\text{DO, and } v_{m,\mathbf{s},\mathbf{s}}(\mathbf{h}) \text{ is exactly } 2p\text{DO.}\end{aligned} \tag{A.1}$$

For $m = 0$, $v_{0,\mathbf{q},\mathbf{s}}(\mathbf{h}) = C_0(\mathbf{q}, \mathbf{s}+\mathbf{h}) \cdot \mathcal{T}_0(\mathbf{q}, \mathbf{s}+\mathbf{h})$ is at least $2p$DO by assumption and hence so is $f_{0,\mathbf{s}}(\mathbf{h}) = \sum_{j=1}^{r_0} a_{0,j}(\mathbf{s}) v_0(\mathbf{q}_{0,j}, \mathbf{s}+\mathbf{h})$. Further, $v_{0,\mathbf{s},\mathbf{s}}(\mathbf{h})$ is exactly $2p$DO. Now assume that (A.1) holds for m. Then, using Equation 3.2, $v_{m+1,\mathbf{q},\mathbf{s}}(\mathbf{h}) = \big(v_{m,q,\mathbf{s}}(\mathbf{h}) - f_m(\mathbf{q}, \mathbf{s}+\mathbf{h})\big) \cdot \mathcal{T}_{m+1}(\mathbf{q}, \mathbf{s}+\mathbf{h})$, which is at least $2p$DO, and so is $f_{m+1,\mathbf{s}}(\mathbf{h}) = \sum_{j=1}^{r_{m+1}} a_{m+1,j}(\mathbf{s}) v_{m+1,\mathbf{q}_{m,j},\mathbf{s}}(\mathbf{h})$. Also, $v_{m+1,\mathbf{s},\mathbf{s}}(\mathbf{h})$ is exactly $2p$DO. This proves (A.1) for $m = 1, \ldots, M$.

In summary, we have $C_{M,\mathbf{s}}(\mathbf{h}) = \sum_{m=0}^{M-1} f_{m,\mathbf{s}}(\mathbf{h}) + (v_{M-1,\mathbf{s},\mathbf{s}}(\mathbf{h}) - f_{M-1,\mathbf{s}}(\mathbf{h})) \cdot \mathcal{T}_M(\mathbf{s}, \mathbf{s}+\mathbf{h})$, where $\mathcal{T}_{M,\mathbf{s}}(\mathbf{h}) = \mathcal{T}_M(\mathbf{s}, \mathbf{s}+\mathbf{h})$ and $f_{m,\mathbf{s}}(\mathbf{h})$, $m = 0, \ldots, M-1$, are all at least $2p$DO, and $v_{M-1,\mathbf{s},\mathbf{s}}(\mathbf{h})$ is exactly $2p$DO.

Thus, $C_{M,\mathbf{s}}(\mathbf{h}) = C_M(\mathbf{s}, \mathbf{s}+\mathbf{h})$ is $2p$DO, and so realizations of the corresponding M-RA process $y_M(\cdot) \sim GP(0, C_M)$ are p times differentiable at $\mathbf{s}$. □

Proof of Proposition 5. First, note that realizations are (mean-square) continuous at $\mathbf{s} \in \mathcal{D}$, if $\lim_{\mathbf{h}\to\mathbf{0}} C_M(\mathbf{s}, \mathbf{s}+\mathbf{h}) = C_M(\mathbf{s}, \mathbf{s})$. Further, we have $\mu_M(\mathbf{s}) = E(y_M(\mathbf{s})|\mathbf{z}) = \mathbf{z}' cov(\mathbf{z})^{-1} C_M(\mathcal{S}, \mathbf{s})$. From the proof of Proposition 4, we have that $C_M(\mathbf{s}_0, \mathbf{s}+\mathbf{h}) = \sum_{m=0}^{M} \sum_{j=1}^{r_m} a_{m,j}(\mathbf{s}_0) v_m(\mathbf{q}_{m,j}, \mathbf{s}+\mathbf{h})$. It is straightforward to show using a proof by induction very similar to that for Proposition 4, that $\lim_{\mathbf{h}\to\mathbf{0}} v_m(\mathbf{q}_{m,j}, \mathbf{s}+\mathbf{h}) = v_m(\mathbf{q}_{m,j}, \mathbf{s})$ if $\lim_{\mathbf{h}\to\mathbf{0}} \mathcal{T}_m(\mathbf{q}_{m,j}, \mathbf{s}+\mathbf{h}) = \mathcal{T}_m(\mathbf{q}_{m,j}, \mathbf{s})$ for all m. In contrast, if $\mathbf{s}$ is on a region boundary, at least one $\mathcal{T}_m(\mathbf{q}_{m,j}, \mathbf{s}+\mathbf{h})$ will be discontinuous as a function of $\mathbf{h}$, and so will $C_M(\mathbf{s}_0, \mathbf{s}+\mathbf{h})$ (unless $v_m(\mathbf{s}, \mathbf{s}+\mathbf{h}) = w_m(\mathbf{s}, \mathbf{s}+\mathbf{h})$ and hence the M-RA-block is exact — see Proposition 6). □

LEMMA 3 (Sum of predictive processes). *For the decomposition in* (3.1)*, the sum of predictive processes up to resolution m is equal in distribution to the predictive process based on the union of the knots up to resolution m, for any $m = 0, 1, \ldots, M$; that is, $\sum_{l=0}^{m} \tau_l(\cdot) \stackrel{d}{=} E(y_0(\cdot)|y_0(\cup_{l=0}^{m} \mathcal{Q}_l))$.*

Proof of Lemma 3. For $m = 1$, $\delta_1(\mathbf{s}) \perp\!\!\!\perp y_0(\mathcal{Q}_0)$, for any $\mathbf{s} \in \mathcal{D}$, because $E\big(\delta_1(\mathbf{s}) y_0(\mathcal{Q}_0)\big) = E\Big(\big(y_0(\mathbf{s}) - E(y_0(\mathbf{s})|y_0(\mathcal{Q}_0))\big) y_0(\mathcal{Q}_0)\Big) = E\big(y_0(\mathcal{Q}_0)\big) E(\delta_1(\mathbf{s})) = 0$, and $y_0(\mathcal{Q}_0)$, $\delta_1(\mathbf{s})$ are jointly Gaussian. And we have $E(y_0(\cdot)|\delta_1(\mathcal{Q}_1), y_0(\mathcal{Q}_0)) = E(y_0(\cdot)|y_0(\mathcal{Q}_1), y_0(\mathcal{Q}_0))$, because for the σ-

algebras

$$\sigma(\delta_1(\mathcal{Q}_1), y_0(\mathcal{Q}_0)) = \sigma\big(y_0(\mathcal{Q}_1) - E\big(y_0(\mathcal{Q}_1)|y_0(\mathcal{Q}_0)\big), y_0(\mathcal{Q}_0)\big) = \sigma\big(y_0(\mathcal{Q}_1), y_0(\mathcal{Q}_0)\big),$$

since $\sigma\big(y_0(\mathcal{Q}_1) - E\big(y_0(\mathcal{Q}_1)|y_0(\mathcal{Q}_0)\big), y_0(\mathcal{Q}_0)\big) = \sigma\big(y_0(\mathcal{Q}_1) - f\big(y_0(\mathcal{Q}_0)\big)\big), y_0(\mathcal{Q}_0)\big) \subset \sigma\big(y_0(\mathcal{Q}_1), y_0(\mathcal{Q}_0)\big)$, and the opposite also holds. Therefore,

$$\begin{aligned} E\big(\delta_1(\mathbf{s})|\delta_1(\mathcal{Q}_1)\big) &= E\big(\delta_1(\mathbf{s})|\delta_1(\mathcal{Q}_1), y_0(\mathcal{Q}_0)\big) \\ &= E\big(y_0(\mathbf{s})|\delta_1(\mathcal{Q}_1), y_0(\mathcal{Q}_0)\big) - E\big(E\big(y_0(\mathbf{s})|y_0(\mathcal{Q}_0)\big)|\delta_1(\mathcal{Q}_1), y_0(\mathcal{Q}_0)\big) \\ &= E\big(y_0(\mathbf{s})|y_0(\mathcal{Q}_1), y_0(\mathcal{Q}_0)\big) - E\big(y_0(\mathbf{s})|y_0(\mathcal{Q}_0)\big), \end{aligned}$$

And so,

$$\tau_0(\mathbf{s}) + \tau_1(\mathbf{s}) = E\big(y_0(\mathbf{s})|y_0(\mathcal{Q}_0)\big) + E\big(\delta_1(\mathbf{s})|\delta_1(\mathcal{Q}_1)\big) = E(y_0(\mathbf{s})|y_0(\mathcal{Q}_1), y_0(\mathcal{Q}_0)).$$

Then, $\delta_2(\mathbf{s}) = y_0(\mathbf{s}) - E\big(y_0(\mathbf{s})|y_0(\mathcal{Q}_0 \cup \mathcal{Q}_1)\big)$, which implies $y_0(\mathcal{Q}_0 \cup \mathcal{Q}_1) \perp\!\!\!\perp \delta_2(\mathbf{s})$. Iteratively repeat this argument to obtain $\sum_{l=0}^{m} \tau_l(\mathbf{s}) = E\big(y_0(\mathbf{s})|y_0(\cup_{l=0}^{m}\mathcal{Q}_l)\big)$. □

LEMMA 4 (Block-independence for exponential covariance). *Assume $y_0(\cdot) \sim GP(0, C_0)$, where C_0 is an exponential covariance function on the real line, $\mathcal{D} = \mathbb{R}$, and consider a domain partitioning as in* (3.7) *with $r_m = (J-1)J^m$ knots for $m = 0, \ldots, M-1$, which are placed such that at each resolution m a knot is located on each boundary between two subregions at resolution $m+1$. Then, for any $m = 1, \ldots, M$, if $\mathbf{s}_i \in \mathcal{D}_{i_1,\ldots,i_m}$ and $\mathbf{s}_j \in \mathcal{D}_{j_1,\ldots,j_m}$, we have $w_m(\mathbf{s}_i, \mathbf{s}_j) = 0$ (defined in* (3.2)*) if $(i_1, \ldots, i_m) \neq (j_1, \ldots, j_m)$.*

Proof of Lemma 4. For any $m = 1, \ldots, M$, using Lemma 3, we have

$$w_m(\mathbf{s}_i, \mathbf{s}_j) = C_0(\mathbf{s}_i, \mathbf{s}_j) - C_0(\mathbf{s}_i, \mathcal{Q}^{m-1})C_0(\mathcal{Q}^{m-1}, \mathcal{Q}^{m-1})^{-1}C_0(\mathcal{Q}^{m-1}, \mathbf{s}_j),$$

where $\mathcal{Q}^{m-1} := \cup_{l=0}^{m-1} \mathcal{Q}_l$. By the law of total covariance,

$$\begin{aligned} w_m(\mathbf{s}_i, \mathbf{s}_j) &= C_0(\mathbf{s}_i, \mathbf{s}_j) - Cov\big(E\big(y_0(\mathbf{s}_i)|y_0(\mathcal{Q}^{m-1})\big), E\big(y_0(\mathbf{s}_j)|y_0(\mathcal{Q}^{m-1})\big)\big) \\ &= E\big(Cov\big(y_0(\mathbf{s}_i), y_0(\mathbf{s}_j)|y_0(\mathcal{Q}^{m-1})\big)\big). \end{aligned}$$

Because $(i_1, i_2, \ldots, i_{m-1}) \neq (j_1, j_2, \ldots, j_{m-1})$, there is a $\mathbf{q} \in \mathcal{Q}^{m-1}$ that lies between $\mathbf{s}_i$ and $\mathbf{s}_j$. As $y_0(\cdot)$ is a Markov process (e.g., Rasmussen and Williams, 2006, Ch. 6), $E\big(Cov\big(y_0(\mathbf{s}_i), y_0(\mathbf{s}_j)|y_0(\mathcal{Q}^{m-1})\big)\big) = E\big(Cov\big(y_0(\mathbf{s}_i), y_0(\mathbf{s}_j)|y_0(\mathbf{q})\big)\big) = w_m(\mathbf{s}_i, \mathbf{s}_j) = 0$. □

Proof of Proposition 6. Comparing the expression for C_M in Lemma 2 to the expression for C_0 in (3.3), it is clear that $C_M(\mathbf{s}_1, \mathbf{s}_2) = C_0(\mathbf{s}_1, \mathbf{s}_2)$ if

$$v_m(\mathbf{s}_i, \mathbf{s}_j) = w_m(\mathbf{s}_i, \mathbf{s}_j), \quad \text{for } m = 0, \ldots, M \text{ and any } \mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}. \tag{A.2}$$

We now prove (A.2) by induction. For $m = 0$, we have $v_0(\mathbf{s}_i, \mathbf{s}_j) = C_0(\mathbf{s}_i, \mathbf{s}_j)\mathcal{T}_0(\mathbf{s}_i, \mathbf{s}_j) = C_0(\mathbf{s}_i, \mathbf{s}_j)$, because $\mathcal{T}_0(\mathbf{s}_i, \mathbf{s}_j) \equiv 1$ for the M-RA-block. For $m > 0$, assume that $v_{m-1}(\mathbf{s}_i, \mathbf{s}_j) = w_{m-1}(\mathbf{s}_i, \mathbf{s}_j)$. Then, we can write

$$v_m(\mathbf{s}_i, \mathbf{s}_j) = w_m(\mathbf{s}_i, \mathbf{s}_j)\mathcal{T}_m(\mathbf{s}_i, \mathbf{s}_j). \tag{A.3}$$

Assume that $\mathbf{s}_i \in \mathcal{D}_{i_1,\ldots,i_m}$ and $\mathbf{s}_j \in \mathcal{D}_{j_1,\ldots,j_m}$. Then, if $(i_1, \ldots, i_m) = (j_1, \ldots, j_m)$, (A.3) holds because $\mathcal{T}_m(\mathbf{s}_i, \mathbf{s}_j) = 1$. If $(i_1, \ldots, i_m) \neq (j_1, \ldots, j_m)$, we have $\mathcal{T}_m(\mathbf{s}_i, \mathbf{s}_j) = 0$ but also $w_m(\mathbf{s}_i, \mathbf{s}_j) = 0$ by Lemma 4. This proves (A.3), which proves (A.2), which in turns proves Proposition 6. □

Proof of Proposition 7. From (3.9), we have $\mathbf{W}_{m,l}^{k+1} = (\mathbf{W}_{m,l}^{k} - \mathbf{X}_{m,l}^{k}) \circ \mathcal{T}_{k+1}(\mathcal{Q}_m, \mathcal{Q}_l)$, where $\mathbf{X}_{m,l}^{k} := \mathbf{W}_{m,k}^{k}\mathbf{\Lambda}_k^{-1}\mathbf{W}_{l,k}^{k\prime}$. The (i,j)th element of this matrix is

$$(\mathbf{X}_{m,l}^{k})_{i,j} = \textstyle\sum_{a,b=1}^{r_k} v_k(\mathbf{q}_{m,i}, \mathbf{q}_{k,a}) v_l(\mathbf{q}_{l,j}, \mathbf{q}_{k,b}) (\mathbf{\Lambda}_k^{-1})_{a,b}, \tag{A.4}$$

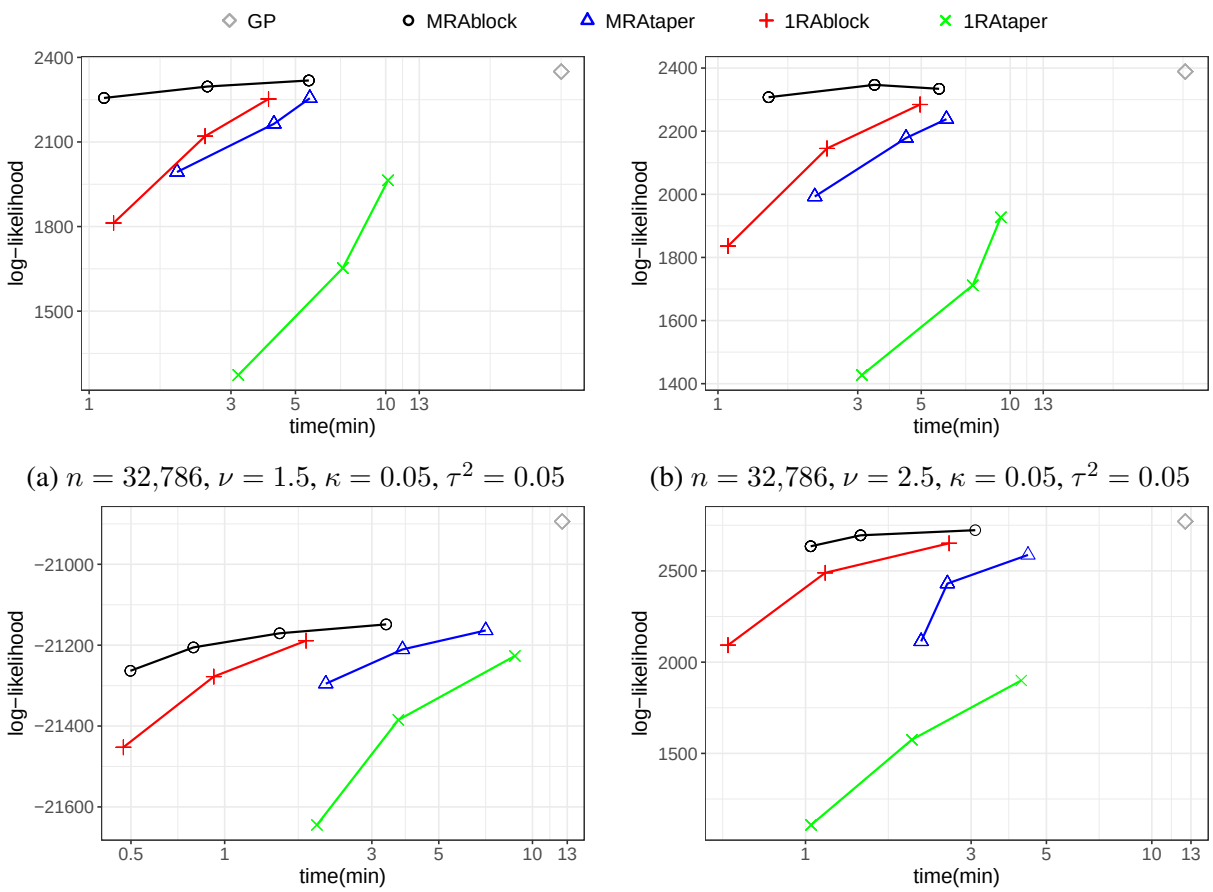
where $v_k(\mathbf{q}_{m,i}, \mathbf{q}_{k,a}) = 0$ if $\|\mathbf{q}_{m,i} - \mathbf{q}_{k,a}\| \geq d_k$, and $v_l(\mathbf{q}_{l,j}, \mathbf{q}_{k,b}) = 0$ if $\|\mathbf{q}_{l,j} - \mathbf{q}_{k,b}\| \geq d_k$. Further, we only need the (i,j)th element of $\mathbf{W}_{m,l}^{k+1}$ (and thus of $\mathbf{X}_{m,l}^{k}$) if $(i,j) \in \mathcal{I}_{m,l}$, because $(\mathbf{W}_{m,l}^{l})_{i,j} = 0$ if $\|\mathbf{q}_{m,i} - \mathbf{q}_{l,j}\| \geq d_l$. Hence, we only need $(\mathbf{\Lambda}_k^{-1})_{a,b}$ if $\|\mathbf{q}_{m,i} - \mathbf{q}_{l,j}\| < d_l$, $\|\mathbf{q}_{m,i} - \mathbf{q}_{k,a}\| < d_k$, *and* $\|\mathbf{q}_{l,j} - \mathbf{q}_{k,b}\| < d_k$, for some $m, l \in \{k+1, \ldots, M\}$. As $d_{k+1} = d_k/J > d_{k+2} > \ldots > d_M$, this means that do not need to calculate $(\mathbf{\Lambda}_k^{-1})_{a,b}$ if $\|\mathbf{q}_{k,a} - \mathbf{q}_{k,b}\| \geq 2d_k + 2d_{k+1} = (2 + 2/J)d_k$, and so we can replace $\mathbf{\Lambda}_k^{-1}$ in $\mathbf{X}_{m,l}^{k}$ by $\mathbf{S}_k = \widetilde{\mathbf{\Lambda}}_k^{-1} \circ \mathbf{G}_k$.

Further, for each $(i,j) \in \mathcal{I}_{m,l}$, the time to compute (A.4) is $\mathcal{O}(r_0^2)$, because for any $\mathbf{s} \in \mathcal{D}$, the size of the set $\{\mathbf{q} \in \mathcal{Q}_k : v_k(\mathbf{s}, \mathbf{q}) \neq 0\}$ is $\mathcal{O}(r_0)$. As $\mathcal{I}_{m,l}$ is a set of size $\mathcal{O}(r_m r_0)$, the cost of computing $\mathbf{W}_{m,l}^{k}$ for each m, l, k is $\mathcal{O}(r_m r_0^3)$. Thus, the total computation time for $k = 0, \ldots, l-1$, $l = 0, \ldots, m$, and $m = 0, \ldots, M$ is $\mathcal{O}(\sum_{m=0}^{M} \sum_{l=0}^{m} \sum_{k=0}^{l-1} r_m r_0^3) = \mathcal{O}(r_0^3 \sum_{m=0}^{M} r_m m^2) = \mathcal{O}(r_0^4 \sum_{m=0}^{M} J^m m^2) = \mathcal{O}(r_0^4 M^2 J^M) = \mathcal{O}(nM^2 r_0^3)$, because $n = \mathcal{O}(r_0 J^M)$ and $\sum_{m=0}^{M} m^2 J^m \leq 2M^2 J^M = \mathcal{O}(MJ^M)$. □

Proof of Proposition 8. We have $(\widetilde{\mathbf{\Lambda}}_{m,l})_{i,j} = 0$ if $\nexists \mathbf{s} \in \mathcal{D}$ such that $\mathcal{T}_m(\mathbf{q}_{m,i}, \mathbf{s}) \neq 0$ and $\mathcal{T}_l(\mathbf{q}_{l,j}, \mathbf{s}) \neq 0$, or equivalently, if $\|\mathbf{q}_{m,i} - \mathbf{q}_{l,j}\| \geq d_m + d_l$. As $d_l = d_m J^{(l-m)/d}$, the ith row $(\widetilde{\mathbf{\Lambda}}_{m,l})_{i,\cdot}$ has $\mathcal{O}(r_0 J^{(l-m)_+})$ nonzero elements, where $(x)_+ = x \mathbb{1}_{\{x \geq 0\}}$. The entire row of the matrix $\widetilde{\mathbf{\Lambda}}$ corresponding to $\mathbf{q}_{m,i}$ thus has $\mathcal{O}(r_0 \sum_{l=0}^{M} J^{(l-m)_+}) = \mathcal{O}(r_0(m + J^{M-m}))$ nonzero elements. As there are $\mathcal{O}(r_0 J^m)$ rows corresponding to resolution m, the total number of nonzero elements in $\widetilde{\mathbf{\Lambda}}$ is $\mathcal{O}(\sum_{m=0}^{M} r_0 J^m \cdot r_0(m + J^{M-m})) = \mathcal{O}(r_0^2(MJ^M + \sum_{m=0}^{M} mJ^m)) = \mathcal{O}(nMr_0)$, because $\sum_{m=0}^{M} mJ^m \leq 2MJ^M = \mathcal{O}(MJ^M)$ and $n = \mathcal{O}(r_0 J^M)$. □

A.2 Additional simulation plots

We provide here additional settings for the simulation study described in Section 4 of this chapter. We consider various settings for the Matérn covariance function with smoothness parameter ν, range parameter κ, and noise or nugget variance τ^2.



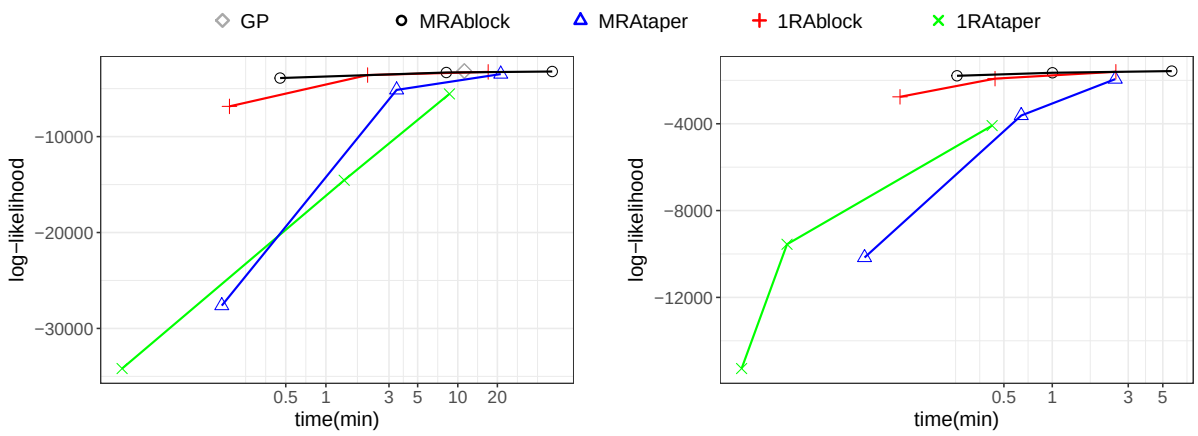
(a) $n = 32{,}786$, $\nu = 1.5$, $\kappa = 0.05$, $\tau^2 = 0.05$

(b) $n = 32{,}786$, $\nu = 2.5$, $\kappa = 0.05$, $\tau^2 = 0.05$

(c) $n = 32{,}786$, $\nu = 1.5$, $\kappa = 0.15$, $\tau^2 = 0.2$

(d) $n = 40{,}960$, $\nu = 1.5$, $\kappa = 0.05$, $\tau^2 = 0.05$

Figure A.1: Comparison of approximation accuracy for different sample sizes in one-dimensional space.



(a) $n = 25{,}600$, $\nu = 1.5$, $\kappa = 0.05$, $\tau^2 = 0.05$ (b) $n = 12{,}544$, $\nu = 2.5$, $\kappa = 0.05$, $\tau^2 = 0.05$

Figure A.2: Comparison of approximation accuracy for different sample sizes in two-dimensional space.