FAST INFERENCE FOR MULTI-SCALE AND GLOBAL SPATIAL PROCESSES

A Dissertation

by

JINGJIE ZHANG

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,	Anthony Filippi
	Debdeep Pati
	Lan Zhou
Head of Department,	Brani Vidakovic

December 2020

Major Subject: Statistics

ABSTRACT

Gaussian processes (GPs) are widely used in geospatial analysis, machine learning and many application areas. We propose novel scalable methods to tackle two problems in Gaussian process modeling for large spatial datasets.

In the first study, we focus on the ubiquitous multi-scale phenomena in geophysical and other applications. To model the multi-scale structure, we propose a novel multi-scale Vecchia (MSV) approximation of GPs. In the MSV method, the increasingly small scales of spatial variation can be captured by increasingly large sets of variables, and then an accurate approximation of the spatial dependence is obtained from very large to very fine scales. By decomposing the observed dataset into different scales, our MSV method can visualize each scale and provide insights for the underlying processes. We develop an algorithm for automatically choosing the tuning parameters, and explore properties of the MSV approximation. We provide comparisons to existing approaches based on simulated data and using satellite measurements of land-surface temperature.

The second is concerned with global spatial processes. Rapid developments in satellite remote-sensing technology have enabled the collection of geospatial data on a global scale, and so there is an increased need for covariance functions that can capture spatial dependence on spherical domains. We propose a general method of constructing nonstationary, locally anisotropic covariance functions on the sphere based on covariance functions for Euclidean space. We provide theorems and conditions such that the resulting correlation function is isotropic or axially symmetric, for sensible parameterizations in specific applications. For modern large datasets on the sphere, the Vecchia approximation is applied to achieve computationally feasible inference. We provide illustrations and comparisons in numerical studies.

DEDICATION

To my mother, my father, and my husband.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Matthias Katzfuss, for his tremendous support, valuable guidance, and continued patience throughout my doctoral studies.

I would also like to thank my advisory committee members, Dr. Debdeep Pati, Dr. Lan Zhou and Dr. Anthony Filippi, for their generous help and advice. Special thanks to Dr. Jing Du and Dr. Jianhua Huang, who provided me with great opportunities to work on interdisciplinary statistical research and apply my statistics expertise to engineering fields.

I am very grateful to the Department of Statistics. The fantastic faculty, responsible staff, and kindhearted friends made my time at Texas A&M University a wonderful experience.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Matthias Katzfuss, Dr. Debdeep Pati and Dr. Lan Zhou of the Department of Statistics and Dr. Anthony Filippi of the Department of Geography.

All work conducted for the dissertation was completed by the student under Dr. Matthias Katzfuss's supervision. Parts of the methodology and proofs in Chapter 3 were coauthored with Zhuoer Sun of the Department of Statistics.

Funding Sources

Graduate study was supported by a graduate assistantship from the Department of Statistics at Texas A&M University, and partially supported by National Science Foundation (NSF) under grant 1416730.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Gaussian processes (GPs) are commonly used as function priors in many application areas such as geospatial analysis and machine learning. GPs are popular because they are flexible, interpretable, and naturally result in probabilistic uncertainty quantification. However, for many modern spatial datasets of interest, there are two inference problems. The first is that direct application of GPs is too computationally expensive as the cost is cubic in the number of observations. Second, as satellites have enabled the collection of global data, there is an increased need for covariance functions that are valid on spheres. Here, we propose two projects focusing on these two problems.

In the first project, we focus on approximations for a large number of observations of a multi-scale GP, which is defined here as a GP whose covariance function is a sum of covariance functions at different scales, or equivalently, as a sum of independent GPs at different scales. Multi-scale processes are ubiquitous in many geophysical and other applications. For example, the atmosphere is affected by micro-scale systems such as clouds and thunderstorms, but also by extratropical cyclones that act on much larger scales (Cotton et al., 2010).

Although most of the existing GP approximation methods can be applied to multi-scale GPs, by simply considering the marginal distribution of the data that implicitly collapses the processes or covariance functions at different scales into one, it can be highly advantageous to exploit the multi-scale structure explicitly and to specify a suitable approximation for each scale. We propose here a multi-scale Vecchia (MSV) approximation for multi-scale GPs observed at point level, which essentially combines suitable Vecchia approximations at each of the different scales. Roughly speaking, smooth large-scale processes can be approximated well using low-rank approaches, while non-smooth fine-scale processes often exhibit strong screening effects and thus can be approximated well by assuming conditional independence using small conditioning sets. By decomposing the observed dataset into different scales, our MSV method can visualize each scale and provide insights for the underlying processes. We develop an algorithm for automatically choosing the

tuning parameters, and explore properties of the MSV approximation. Our approach also leads to nice visualizations of the different scales, which can be highly useful in many scientific contexts.

The second project is about global spatial processes. Traditionally, geostatistical analysis relied on approximating small or regional spatial domains as flat, as subsets of $\mathbb{R}^2$. However, as satellites have enabled the collection of global data, there is an increased need for covariance functions that are valid on spheres. Thus, our focus here is on spatial data observed on the unit 2-sphere, $\mathbb{S} = \{\tilde{\mathbf{s}} \in \mathbb{R}^3 : \|\tilde{\mathbf{s}}\| = 1\}$.

For processes that reside on the surface of a sphere, two different measures of distance are commonly used. The most natural distance is great-arc distance, which measures the distance "going along the surface of the sphere". In contrast, Euclidean or chordal distance pierces through the sphere.

It is not trivial to build a great-arc-distance-based covariance function on the sphere because of the curvature of $\mathbb{S}$ (e.g., Jones, 1963). Most well-known covariance functions are valid (i.e., positive definite) only when used with Euclidean distance on $\mathbb{R}^d$, for some (or all) $d \geq 1$. They are not guaranteed to be valid when used with great-arc distance (e.g. Gneiting, 2013). Much of the work on covariance functions that are valid with great-arc distance has focused on isotropic covariance functions.

We consider here using Euclidean distance and restrict a valid covariance function in $\mathbb{R}^3$ to $\mathbb{S}$ according to Yaglom (1987). This is guaranteed to be valid, in that we know the covariance function is valid for points in $\mathbb{R}^3$, it just so "happens" that all locations now fall onto the surface of our sphere within this space. In addition, results based on chordal and great-arc distance are often indistinguishable (Guinness and Fuentes, 2016).

We develop nonstationary, locally anisotropic covariance functions on the sphere based on the locally anisotropic covariance functions for Euclidean space proposed in Paciorek and Schervish (2006). The approach is very general, and we provide theorems and conditions such that the resulting correlation function is isotropic or axially symmetric, for sensible parameterizations in specific applications. For large datasets, we can also apply the Vecchia approximation for scalable

inference.

The remainder of this dissertation is organized as follows. In Chapter 2, we derive the multi-scale Vecchia approximations of Gaussian processes, and conduct numerical comparisons to existing approaches. We provides an application of MSV to satellite measurements of land-surface temperature. In Chapter 3, we construct classes of nonstationary covariance functions on the sphere, and provide theorems and conditions for important special cases such as isotropy and axial symmetry. We simulate Gaussian process realizations using various covariance functions constructed by our method, and provide comparisons in numerical studies. Chapter 4 concludes.

2. MULTI-SCALE VECCHIA APPROXIMATIONS OF GAUSSIAN PROCESSES

2.1 Introduction

Gaussian processes (GPs) are commonly used as function priors in machine learning (e.g., Rasmussen and Williams, 2006) and geospatial modeling (e.g., Cressie and Wikle, 2011). GPs are popular because they are flexible, interpretable, and naturally result in probabilistic uncertainty quantification. However, direct application of GPs is computationally expensive for large datasets, as the cost is cubic in the number of observations. Some of the existing GP approximation methods rely on sparsity (Furrer et al., 2006; Du et al., 2009; Lindgren et al., 2011); some rely on low-rank structure (e.g., Wikle and Cressie, 1999; Quiñonero-Candela and Rasmussen, 2005; Cressie and Johannesson, 2008; Katzfuss and Cressie, 2011); and some on a combination of the two (e.g., Snelson and Ghahramani, 2007; Sang et al., 2011).

We focus on approximations for a large number of observations of a multi-scale GP, which is defined here as a GP whose covariance function is a sum of covariance functions at different scales, or equivalently, as a sum of independent GPs at different scales. Multi-scale processes are ubiquitous in many geophysical and other applications. For example, environmental processes are often subject to diurnal, seasonal, and multi-year cycles over time (Kim et al., 2007); the atmosphere is affected by micro-scale systems such as clouds and thunderstorms, but also by extratropical cyclones that act on much larger scales (Cotton et al., 2010); and for soil moisture, short-range dependence is governed by surface characteristics such as soil texture, vegetation, and topography, while long-range dependence is due to precipitation (Skøien et al., 2003). GPs whose covariance functions are sums of kernels at different scales are also often used in GP emulation (e.g., Ba and Joseph, 2012), astronomy (e.g., Sobolewska et al., 2014), and machine learning (e.g., Rasmussen and Williams, 2006; Wilson and Adams, 2013; Wilson et al., 2014). Further applications can be found in Ferreira and Lee (2007), for example.

Most of the GP approximations described above can be applied to multi-scale GPs, by sim-

ply considering the marginal distribution of the data, which implicitly collapses the processes or covariance functions at different scales into one. While its name might imply differently, this marginal approach is also the one considered in the multi-resolution approximation (Katzfuss, 2017; Katzfuss and Gong, 2020). In contrast, we will show here that it can be highly advantageous to exploit the multi-scale structure explicitly and to specify a suitable approximation for each scale.

Multi-scale approaches from engineering often do not result in consistent joint statistical models, and they usually work on developing coarser representations of the intersed phenomenon to achieve computationally fast algorithms (e.g., Saquib et al., 1996; Comer and Delp, 1999). In statistics, most existing multi-scale approaches use tree-structured models, and work on data collected at different scales. For example, Zhu et al. (2004) develop a spatial model for soil data collected at varying resolutions and accuracies, and they define the neighborhood structure by a parent-child relationship presented in a multi-level tree. Huang et al. (2002) propose a multi-resolution autoregressive tree-structured model for fast and resolution-consistent statistical prediction for satellite data measured at different resolutions. Similar tree-structured approaches can be found in Gotway and Young (2002) and Tzeng et al. (2005). There also exist some literature on multi-scale time-series models (Ferreira et al., 2006), which couple standard linear models at various time scales via stochastic links across scales. Ferreira and Lee (2007) give an overview of these multiscale models.

We propose here a multi-scale Vecchia (MSV) approximation for multi-scale GPs observed at point level, which essentially combines suitable Vecchia approximations at each of the different scales. The Vecchia approximation, originally proposed for the data vector directly (i.e., for a single level) in Vecchia (1988), replaces the high-dimensional joint distribution of the entire data vector with a product of univariate conditional distributions. The conditioning set for each univariate conditional distribution consists of a small subset of previously ordered variables. This can lead to tremendous computational savings if each conditioning set is small, which can be assumed if the so-called screening effect holds. However, the screening effect is relatively weak when observations include a nugget or noise term. Katzfuss and Guinness (2017) and Katzfuss et al. (2020a)

proposed a general Vecchia approach that treated the noise term separately from the continuous covariance component, and thus showed that the screening effect can largely be restored and the conditioning sets can be small.

Our MSV approach essentially extends the general-Vecchia idea to multiple levels: at each level, a suitable Vecchia approximation is found for the process acting at the corresponding scale. Roughly speaking, smooth large-scale processes can be approximated well using low-rank approaches, while non-smooth fine-scale processes often exhibit strong screening effects and thus can be approximated well using small conditioning sets. In this context, a nugget or noise term is the ultimate fine-scale process, which is independent over space and thus does not require any conditioning. As shown by Katzfuss and Guinness (2017), all of these different approximations are merely special cases of the Vecchia approach, and thus can be combined into the MSV here.

We describe how to efficiently conduct inference using the MSV, and we provide an algorithm for automatic choice of the number of knot variables and the conditioning set size at each level. This algorithm is also applicable and useful for one-level (Vecchia, 1988) or two-level (Katzfuss and Guinness, 2017) Vecchia approximations. Our approach also leads to nice decompositions and visualizations of the different scales, which can be highly useful in many scientific contexts. We generally assume the covariance functions at the different levels to be known (e.g., from expert knowledge, or by using existing algorithms), and focus on accurate approximation of the resulting spatial dependence; however, we also provide a computationally cheap approximation to the integrated likelihood, which can be employed for parameter inference.

The remainder of this chapter is organized as follows. Section 2.2 derives the multi-scale Vecchia approximation, and develops an algorithm for automatically choosing the tuning parameters. Section 2.3 contains further derivations and proofs. In Section 2.4, we perform numerical studies and comparisons to existing methods. Section 2.5 provides an application of MSV to satellite measurements of land-surface temperature. We conclude in Section 2.6.

2.2 Methodology

2.2.1 A multi-scale Gaussian process

Consider a Gaussian process (GP) $z(\cdot) \sim GP(0, C)$ on a domain or spatial region $\mathcal{D} \subset \mathbb{R}^d$ with covariance function C. Assume that $z(\cdot)$ is a multi-scale process in the sense that $z(\cdot) = \sum_{\ell=1}^{L} y^{(\ell)}(\cdot)$, where the processes at the individual levels, $y^{(\ell)}(\cdot) \overset{ind.}{\sim} GP(0, C^{(\ell)})$, $\ell = 1, \ldots, L$, are ordered from large scales to fine scales. We think of the scale of a process here in terms of the effective range of its covariance function (i.e., the distance beyond which the correlation drops below a small threshold, such as 0.05), and we assume throughout that $y^{(L)}(\cdot)$ is Gaussian white noise. For example, in spatial statistics, $z(\cdot)$ is often modeled as the sum of a large-scale, fine-scale, and nugget or noise component. A simple toy example is shown in Figure 2.1.

Due to the independence assumption of the different processes, the covariance of $z(\cdot)$ is

$$C(\mathbf{s}_i, \mathbf{s}_j) = \sum_{\ell=1}^{L} C^{(\ell)}(\mathbf{s}_i, \mathbf{s}_j), \qquad \mathbf{s}_i, \mathbf{s}_j \in \mathcal{D}. \tag{2.1}$$

Assume we have n observations $\mathbf{z} = (z_1, \ldots, z_n)^\top$ of $z(\cdot)$, such that $z_i = z(\mathbf{s}_i)$. In general, inference involving n observations of a GP requires $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ time. This is computationally infeasible when n is in the tens of thousands or more, and so for many datasets of interest, GP approximations are necessary.

2.2.2 Multi-scale Vecchia approximation

To obtain a fast approximation of the GP $z(\cdot)$, one could simply apply an existing GP approximation to $z(\cdot)$ directly, using the "collapsed" covariance function C in (2.1). However, the main idea of our multi-scale Vecchia (MSV) approximation is that it can often be highly beneficial to consider each covariance $C^{(\ell)}$ separately, and tailor an approximation specifically to each of the L levels. Simply speaking, smooth large-scale components can often be approximated well by a low-rank process relying on a small number of anchoring points or knots, and non-smooth fine-scale components often exhibit strong screening or conditional-independence properties (see Section

2.2.3 below for more details), while neither approximation might work well for the sum of the two components.

To specify the MSV, define $\mathbf{y}^{(\ell)} = (y^{(\ell)}(\mathbf{s}_1), \ldots, y^{(\ell)}(\mathbf{s}_n))^\top = (y_1^{(\ell)}, \ldots, y_n^{(\ell)})^\top$ for each level $\ell = 1, \ldots, L-1$. Denote the vector of anchoring points or knots at level ℓ as $\mathbf{y}_\ell = (y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)})^\top$. Stack all variables into a vector $\mathbf{x} = (\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_{L-1}^\top, \mathbf{z}^\top)^\top$.

The exact distribution of the observation vector $\mathbf{z}$ is given by $f(\mathbf{z}) = \int f(\mathbf{x}) d\mathbf{y}_{1:L-1}$, where

$$f(\mathbf{x}) = \big(\prod_{\ell=1}^{L-1} \prod_{i=1}^{n_\ell} f(y_i^{(\ell)} | y_1^{(\ell)}, \ldots, y_{i-1}^{(\ell)})\big)\big(\prod_{i=1}^{n} f(z_i | \mathbf{y}_1, \ldots, \mathbf{y}_{L-1}, z_1, \ldots, z_{i-1})\big).$$

Our MSV is essentially a Vecchia approximation applied to the distribution $f(\mathbf{x})$,

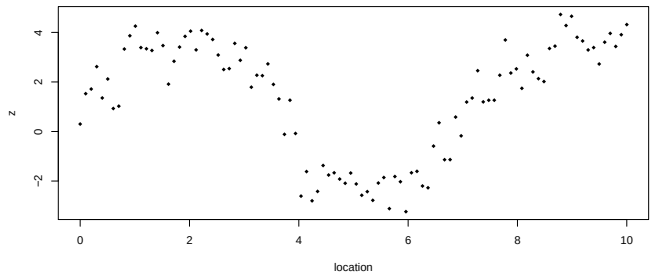
$$\hat{f}(\mathbf{x}) = \Big(\prod_{\ell=1}^{L-1} \prod_{i=1}^{n_\ell} f(y_i^{(\ell)} | N_{\mathbf{y}_i^{(\ell)}})\Big)\Big(\prod_{i=1}^{n} f(z_i | N_{z_i})\Big),$$

where for each $y_i^{(\ell)}$ the full conditioning set $y_1^{(\ell)}, \ldots, y_{i-1}^{(\ell)}$ is replaced by a subset $N_{\mathbf{y}_i^{(\ell)}}$, which denotes the nearest m_ℓ variables in space to variable $y_i^{(\ell)}$ among the previously ordered knot variables $\{y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)}\}$. For each z_i the full conditioning set is replaced by a subset $N_{z_i} = \{N_{z_i}^{(1)}, \ldots, N_{z_i}^{(L-1)}\}$, where $N_{z_i}^{(\ell)} = \{y_i^{(\ell)}\}$ for $i \leq n_\ell$, and $N_{z_i}^{(\ell)}$ consists of the nearest m_ℓ variables in space to variable z_i among the knot variables $\{y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)}\}$ for $i > n_\ell$.

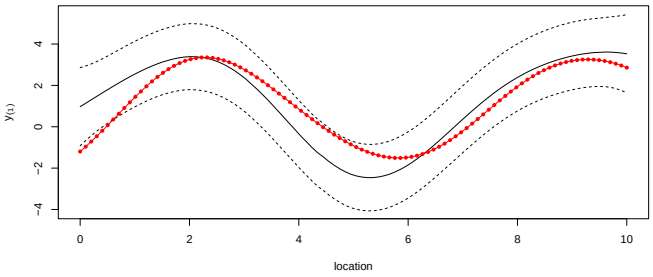
Thus, to specify an MSV approximation, first an ordering of the locations must be chosen, resulting in $\mathbf{s}_1, \ldots, \mathbf{s}_n$. Then, for each level $\ell = 1, \ldots, L-1$, based on having selected n_ℓ and m_ℓ, the Vecchia conditioning set for each variable consists of the nearest m_ℓ previously ordered variables among the n_ℓ knots (i.e., the variables corresponding to the first n_ℓ locations in the ordering). For the level $\mathbf{z}$, the Vecchia conditioning set for each variable consists of the nearest m_ℓ variables among the n_ℓ knots, $\ell = 1, \ldots, L-1$.

2.2.3 Examples of covariance approximations

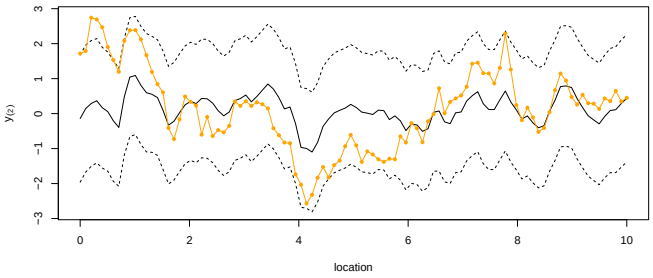
Many types of covariances can be approximated very well using special cases of the Vecchia approximation. We give some examples here, mostly with isotropic covariance functions, which



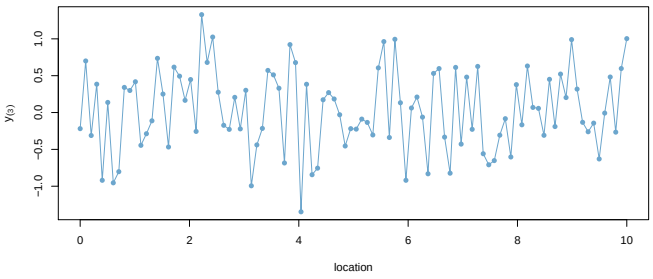
(a) Observations $\mathbf{z}$



(b) $y^{(1)}(\cdot)$; $n_1 = m_1 = 12$



(c) $y^{(2)}(\cdot)$; $m_2 = 1$



(d) $y^{(3)}(\cdot)$

Figure 2.1: A simple toy example of (a) observations $\mathbf{z}$ of a multi-scale process obtained as the sum of components (colored dots and lines) with (b) squared exponential, (c) exponential, and (d) nugget covariance, respectively, on a one-dimensional domain $\mathcal{D} = [0, 10]$. Posterior means (black solid lines) and 95% intervals (black dashed lines) for levels 1 and 2 were obtained using MSV as discussed in Section 2.2.4.3. Knot sets and conditioning set sizes were computed using Algorithms 1 and 2 and led to a virtually exact approximation, so that the approximate posterior summaries are basically identical to those obtained using the exact GP.

are specified as functions of the distance $r = \|\mathbf{s}_i - \mathbf{s}_j\|$ between two locations. However, our approach does not require isotropy. We discuss the polynomial covariance function, squared exponential covariance function, exponential covariance function, Matérn covariance function and nugget examples:

- **Polynomial:** Consider a polynomial $y^{(\ell)}(\mathbf{s}) = \mathbf{p}(\mathbf{s})^\top \boldsymbol{\beta}$ as a function of location $\mathbf{s}$ with p coefficients $\boldsymbol{\beta} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, which is often used to capture a large-scale trend term in spatial applications. Such a polynomial can be approximated exactly using a Vecchia approximation with $n_\ell = m_\ell = p$ (see Proposition 2 in Section 2.3.3). For example, in two-dimensional space, we might set $\mathbf{p}(\mathbf{s}) = (1, s_1, s_2, s_1^2, s_2^2, s_1 s_2)^\top$ for $\mathbf{s} = (s_1, s_2)^\top$ and thus $p = 6$.

- **Squared exponential:** The squared exponential covariance function, $C^{(\ell)}(r) \propto \exp(-r^2/\lambda^2)$, leads to covariance matrices with exponentially decaying spectrum. Thus, the resulting covariance matrices are approximately low-rank. A process of rank n_ℓ can be approximated using Vecchia with a coarse grid of n_ℓ knots over $\mathcal{D}$, and by conditioning on all previous variables in the knot set (i.e., $m_\ell = n_\ell$).

- **Exponential:** Covariance matrices based on the exponential covariance, $C^{(\ell)}(r) \propto \exp(-r/\lambda)$, have a slowly decaying spectrum, and so a good approximation requires the knot set to be essentially equal to the set of observed locations (i.e., $n_\ell \approx n$). However, the inverse of the covariance matrix (i.e., the precision matrix) is typically approximately sparse, meaning that a strong screening effect holds. In the context of a Vecchia approximation, this allows us to choose the conditioning set to consist of only a small number m_ℓ of nearby locations. For example, in one dimension one can achieve an exact approximation by ordering locations from left to right and only conditioning on the $m_\ell = 1$ previous variable.

- **Matérn:** The Matérn class of covariance functions has a smoothness parameter ν, with realizations being k times differentiable if $\nu > k$. It includes the exponential ($\nu = 0.5$) and squared exponential ($\nu = \infty$) covariance as special cases on (almost) opposite ends of the smoothness spectrum. For covariance functions in between these extreme cases, we generally need

fewer and fewer knots but (relatively) larger and larger conditioning sets (i.e., smaller n_ℓ but larger $\frac{m_\ell}{n_\ell}$) as ν increases.

- **Nugget:** A spatially independent noise term, also called a nugget, is the ultimate fine-scale process with covariance function $C^{(\ell)}(r) \propto \mathbb{1}_{[r=0]}$. Due to the independence, the knot set has to be equal to the observed locations (i.e., $n_\ell = n$), but Vecchia is exact even if $m_\ell = 0$.

2.2.4 Inference

2.2.4.1 Matrices needed for inference

Similarly to Proposition 1 in Katzfuss and Guinness (2017), we can write the MSV as

$$\hat{f}(\mathbf{x}) = \prod_{\ell=1}^{L-1} \Big(\prod_{i=1}^{n_\ell} \mathcal{N}\left(y_i^{(\ell)} | B_i^{(\ell)} N_{\mathbf{y}_i^{(\ell)}}, D_i^{(\ell)}\right) \Big) \Big(\prod_{i=1}^{n} \mathcal{N}(z_i | B_i^{(L)} N_{z_i}, D_i^{(L)}) \Big) = \mathcal{N}_{n+\sum_{\ell=1}^{L-1} n_\ell}(\mathbf{x} | \mathbf{0}, \hat{\mathbf{C}}),$$

where $\hat{\mathbf{C}}^{-1} = \mathbf{U}\mathbf{U}^\top$, $Cov(y_i^{(\ell)}, y_j^{(\ell)}) = C^{(\ell)}(\mathbf{s}_i, \mathbf{s}_j)$, and for $\ell = 1, 2, \ldots, L-1$,

$$\begin{aligned} B_i^{(\ell)} &= Cov(y_i^{(\ell)}, N_{\mathbf{y}_i^{(\ell)}}) \, Cov(N_{\mathbf{y}_i^{(\ell)}}, N_{\mathbf{y}_i^{(\ell)}})^{-1}, \\ D_i^{(\ell)} &= Cov(y_i^{(\ell)}, y_i^{(\ell)}) - B_i^{(\ell)} Cov(N_{\mathbf{y}_i^{(\ell)}}, y_i^{(\ell)}), \end{aligned} \tag{2.2}$$

and for $\ell = L$,

$$\begin{aligned} B_i^{(L)} &= Cov\left(z_i, N_{z_i}\right) Cov\left(N_{z_i}, N_{z_i}\right)^{-1}, \\ D_i^{(L)} &= Cov\left(z_i, z_i\right) - B_i^{(L)} Cov\left(N_{z_i}, z_i\right). \end{aligned} \tag{2.3}$$

The upper-triangular sparse matrix $\mathbf{U}$ is specified by the $B_i^{(\ell)}$ and $D_i^{(\ell)}$ as detailed in Section 2.3.1.

2.2.4.2 *Likelihood*

Similarly to Katzfuss and Guinness (2017), the likelihood $\hat{f}(\mathbf{z}) = \int \hat{f}(\mathbf{x}) d\mathbf{y}_{1:L-1}$ can be computed based on $\mathbf{U}$:

$$-2\log \hat{f}(\mathbf{z}) = \sum_{\ell=1}^{L}\sum_{i=1}^{n_\ell} \log D_i^{(\ell)} + 2\sum \log \mathbf{V}_{ii} + \tilde{\mathbf{z}}^\top\tilde{\mathbf{z}} - (\mathbf{V}^{-1}\mathbf{U}_y\tilde{\mathbf{z}})^\top(\mathbf{V}^{-1}\mathbf{U}_y\tilde{\mathbf{z}}) + n\log(2\pi), \tag{2.4}$$

where $\tilde{\mathbf{z}} = \mathbf{U}_z^\top\mathbf{z}, \mathbf{W} := \mathbf{U}_y\mathbf{U}_y^\top, \mathbf{V} = chol(\mathbf{W})$, and $\mathbf{U}_z$ and $\mathbf{U}_y$ are the matrices consisting only of the rows of $\mathbf{U}$ corresponding to $\mathbf{z}$ and $(\mathbf{y}_1, \ldots, \mathbf{y}_{L-1})$, respectively.

This expression of the MSV likelihood can be evaluated cheaply. Thus, while we generally assume model parameters (e.g., in the covariance functions $C^{(\ell)}$) to be fixed here, the MSV likelihood in (2.4) allows us to carry out frequentist and Bayesian inference on unknown model parameters. Note that there might be identifiability issues when the number of levels L is large. To avoid this, the parameter spaces are often restricted, sometimes through prior distributions, to ensure identifiability, with lower levels accounting for longer-range dependence (e.g., Ba and Joseph, 2012).

2.2.4.3 *Prediction*

For prediction at observed and unobserved locations, we first consider the posterior distribution of $\mathbf{y}_{1:L-1} = (\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_{L-1}^\top)^\top$ given $\mathbf{z}$. In an adaptation of the results in Katzfuss et al. (2020a), we have

$$\mathbf{y}_{1:L-1}|\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}^{-1}),$$

where $\boldsymbol{\mu} = -(\mathbf{V}^\top)^{-1}\mathbf{V}^{-1}\mathbf{U}_y\tilde{\mathbf{z}}$ and $\mathbf{W} = \mathbf{U}_y\mathbf{U}_y^\top$ can be computed cheaply based on $\mathbf{U}$ and $\mathbf{V}$.

Now consider linear combinations of the form $\mathbf{H}\mathbf{y}_{1:L-1}$. For example, we might be interested in inference on each scale $\mathbf{y}_\ell = \mathbf{H}_\ell\mathbf{y}_{1:L-1}$, where $\mathbf{H}_\ell$ is a submatrix of the identity, for $\ell = 1, \ldots, L-1$. Similar to Katzfuss et al. (2020a, Sect. 3.3), we have

$$\mathbf{H}\mathbf{y}_{1:L-1}|\mathbf{z} \sim \mathcal{N}(\mathbf{H}\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathbf{H}}),$$

where the covariance matrix can be computed as $\mathbf{\Sigma_H} = (\mathbf{V}^{-1}\mathbf{H}^\top)^\top(\mathbf{V}^{-1}\mathbf{H}^\top)$, and its diagonal elements are $diag(var(\mathbf{H}\mathbf{y}_{1:L-1}|\mathbf{z})) = ((\mathbf{V}^{-1}\mathbf{H}^\top) \circ (\mathbf{V}^{-1}\mathbf{H}^\top))^\top \mathbf{1}$, where $\mathbf{1}$ is a vector of ones and $\circ$ denotes element-wise multiplication.

For prediction of $y^{(\ell)}(\cdot)$ at any unobserved location $\mathbf{s}_0$ based on observed location set $\mathcal{S}_\ell$, we show in Section 2.3.2 that

$$E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) = C^{(\ell)}(\mathbf{s}_0, \mathcal{S}_\ell)\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\mathbf{H}_\ell\boldsymbol{\mu}$$

and

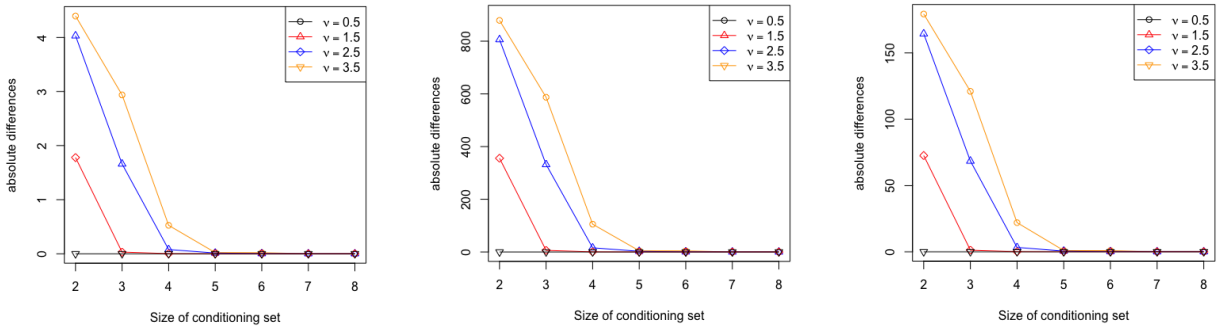
$$var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) = C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - \mathbf{c}_\ell(\mathbf{s}_0)^\top \mathbf{c}_\ell(\mathbf{s}_0) + \tilde{\mathbf{c}}_\ell(\mathbf{s}_0)^\top \tilde{\mathbf{c}}_\ell(\mathbf{s}_0),$$

where $\mathbf{c}_\ell(\mathbf{s}_0) = \mathbf{U}^{(\ell)\top} C^{(\ell)}(\mathcal{S}_\ell, \mathbf{s}_0)$, $\tilde{\mathbf{c}}_\ell(\mathbf{s}_0) = \mathbf{V}^{-1}\mathbf{H}_\ell^\top \mathbf{U}^{(\ell)}\mathbf{c}_\ell(\mathbf{s}_0)$, and $\mathbf{U}^{(\ell)}$ is the block of $\mathbf{U}$ corresponding to knot variables at level ℓ. These posterior distributions are illustrated in Figure 2.1.

2.2.5 Automatic choice of knots and conditioning sets

To specify the MSV for a given dataset, for each level we need to determine the knot and conditioning sets, based on a location ordering. To simplify this problem, assume that the locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ are ordered using a maximum-minimum distance ordering (Guinness, 2016; Schäfer et al., 2017), and that the variables in each $\mathbf{y}^{(\ell)} = (y^{(\ell)}(\mathbf{s}_1), \ldots, y^{(\ell)}(\mathbf{s}_n))^\top = (y_1^{(\ell)}, \ldots, y_n^{(\ell)})^\top$ are ordered accordingly. As described in Section 2.2.2, we specify the knot variables $\mathbf{y}_\ell = (y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)})^\top$ as the first n_ℓ variables in this ordering. The conditioning vector $N_{\mathbf{y}_i^{(\ell)}}$ consists of the nearest m_ℓ variables in space to variable $y_i^{(\ell)}$ among previously ordered variables in $\mathbf{y}_\ell$.

Given these constraints, we only need to choose n_ℓ and m_ℓ for each level $\ell = 1, \ldots, L-1$. If we simply set $n_\ell = m_\ell = n$, the approximation will be exact, but this choice leads to computational infeasibility when n is large. Hence, we propose to pick the smallest n_ℓ and m_ℓ at each level such that the improvement in accuracy by increasing n_ℓ and m_ℓ further is negligible. We consider the Kullback-Leibler (KL) divergence between the exact distribution and approximated distribution as a measure of accuracy. Explicit computation of the KL divergence requires $\mathcal{O}(n^3)$ time and is hence impractical for large n, but it turns out that we do not have to calculate it explicitly to



(a) $KL(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)}))$ (b) $\sum_{i=1}^{n} \log(D_i^{(\ell)})$ (c) $\sum_{i \in T} \log(D_i^{(\ell)})$

Figure 2.2: For the KL divergence and two computationally cheaper alternative quantities, differences for subsequent values of the size m_ℓ of the conditioning sets, for a Matérn covariance function with range 1 and different smoothness values ν. Numerically, we show that convergence of KL divergence as a function of m_ℓ is equivalent to convergence of the sum of all log conditional variances, which in turn is closely approximated by the sum of log conditional variances for the last $t = 20$ locations in maxmin ordering.

implement our desired algorithm.

Theorem 1. *For each level $\ell = 1, \ldots, L-1$, the KL divergence between the true distribution $f(\mathbf{y}^{(\ell)})$ and a Vecchia approximation $\hat{f}(\mathbf{y}^{(\ell)})$ is*

$$KL\left(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)})\right) = \frac{1}{2}\sum_{i=1}^{n} \log(D_i^{(\ell)}) - c(f(\mathbf{y}^{(\ell)}))$$

where $D_i^{(\ell)} = Var(y_i^{(\ell)}|N_{\mathbf{y}_i^{(\ell)}})$ is the conditional variance given in (2.2)*, and $c(f(\mathbf{y}^{(\ell)}))$ depends on the exact distribution $f(\mathbf{y}^{(\ell)})$ but is constant with respect to m_ℓ, n_ℓ.*

The proof can be found in Section 2.3.4. Thus, minimizing (as a function of n_ℓ and m_ℓ) this KL divergence at each level $l = 1, \ldots, L-1$ is equivalent to minimizing the sum of (or each of) the $\log(D_i^{(\ell)})$ over all variables or locations. In practice, to achieve further speed-ups for large datasets, we minimize the conditional variances for a systematically chosen subset of locations, specifically the last t locations in the maxmin ordering, with indices $T = \{n - t + 1, \ldots, n\}$. Figure 2.2 illustrates this can be a valid approach.

The resulting proposed procedure for automatically choosing the tuning parameters n_ℓ and m_ℓ is described in Algorithm 2, which relies on Algorithm 1 for choosing m_ℓ for a fixed n_ℓ. Note that

Algorithm 2 can be run in parallel for each $\ell = 1, 2, \ldots, L-1$.

Algorithm 1 `chooseM`: Automatic choice of m_ℓ for given n_ℓ

Input: Covariance $C^{(\ell)}$, tolerance $\varepsilon > 0$, maximum conditioning set size $m_{\max}$, knot set size n_ℓ
for $m_\ell = 1, 2, \ldots, \min(m_{\max}, n_\ell)$ **do**
    Compute $D_j^{(\ell)}(m_\ell) = var(y_j^{(\ell)} | \mathbf{y}_{N_j^{(\ell)}})$ using n_ℓ, m_ℓ for all $j \in T$
    if $\forall j \in T, \left| \frac{\log D_j^{(\ell)}(m_\ell+1) - \log D_j^{(\ell)}(m_\ell)}{\log D_j^{(\ell)}(m_\ell)} \right| < \varepsilon$ or $D_j^{(\ell)}(m_\ell + 1) = \texttt{NA}$ **then**
        Break
    end if
end for
return m_ℓ and corresponding $D_{j \in T}^{(\ell)}$

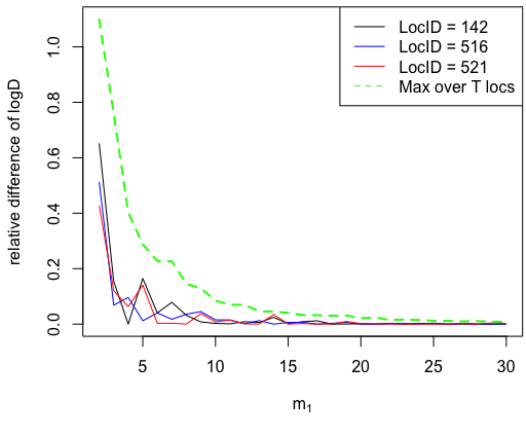
In Algorithm 1, for given n_ℓ, we choose m_ℓ based on $\left| \frac{\log D_j^{(\ell)}(m_\ell+1) - \log D_j^{(\ell)}(m_\ell)}{\log D_j^{(\ell)}(m_\ell)} \right|$, which is the relative difference of the logarithm of conditional variance $D_j^{(\ell)}$ for $j \in T$. In practice, we choose the size of T as $\min\{1000, n\}$. Note that, especially for large m_ℓ, $Cov(\mathbf{y}_{N_j^{(\ell)}}, \mathbf{y}_{N_j^{(\ell)}})$ can become numerically singular, in which case we have $D_j^{(\ell)} = \texttt{NA}$. But this would also imply that enlarging the conditioning set does not result in any improvement in the conditional variance, and so the algorithm will stop. Similarly, we also terminate the algorithm if $D_j^{(\ell)}$ is extremely small (smaller than some specified threshold ε).

In Algorithm 2, we start with $n_\ell = 1$, and compute $D_{j \in T}^{(\ell)}$ using Algorithm 1. To speed up the algorithm, we double the step size of n_ℓ and compute the corresponding $D_{j \in T}^{(\ell)}$ until n_ℓ reaches the data size n or the relative difference of logarithm of conditional variance for each location in T converges.

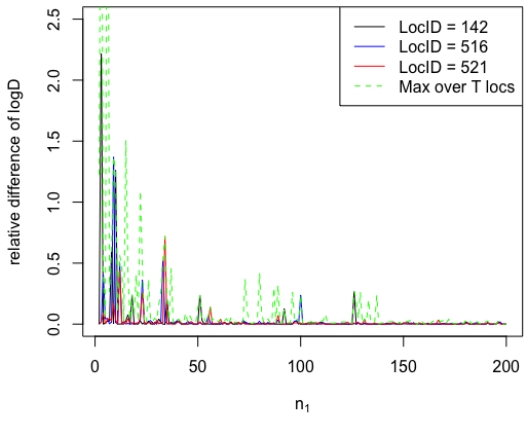
We illustrate Algorithms 1 and 2 for a Matérn covariance in Figure 2.3. We also applied Algorithm 2 to the toy example of $n = 100$ simulated observations in Figure 2.1, for which a virtually exact approximation was obtained for $n_1 = m_1 = 12$, $n_2 = n = 100$, and $m_2 = 1$.

Algorithm 2 Automatic choice of n_ℓ and m_ℓ

Input: $C^{(\ell)}$, n, ε, $m_{\max}$, T. Default: $m_{\max} = 30$
$n_\ell = 0, \texttt{stepsize} = 1, \texttt{MinSum} = \infty$
while $n_\ell \leq n$ **do**
  $n_\ell = n_\ell + \texttt{stepsize}$
  $[m_\ell, D^{(\ell)}_{j \in T}] = \texttt{chooseM}\left(C^{(\ell)}, \varepsilon, m_{\max}, n_\ell\right)$ (Algorithm 1)
  if $\texttt{last}D^{(\ell)}_j$ exists and $\forall j \in T$, $\left|\frac{\log D^{(\ell)}_j - \log \texttt{last}D^{(\ell)}_j}{\log \texttt{last}D^{(\ell)}_j}\right| < \varepsilon$ or $D^{(\ell)}_j = \texttt{NA}$ **then**
    Break
  end if
  if $\sum_{j \in T} D^{(\ell)}_j < \texttt{MinSum}$ **then**
    $\texttt{MinSum} = \sum_{j \in T} D^{(\ell)}_j$
    $\texttt{best\_m}_\ell = m_\ell$
    $\texttt{best\_n}_\ell = n_\ell$
  end if
  $\texttt{last}D^{(\ell)}_{j \in T} = D^{(\ell)}_{j \in T}$
  $\texttt{stepsize} = 2 * \texttt{stepsize}$
end while
return $\texttt{best\_n}_\ell$, and $\texttt{best\_m}_\ell$



(a) Algorithms 1 (given $n_1 = 63$)



(b) Algorithm 2

Figure 2.3: Illustration of Algorithms 1 and 2 for a Matérn covariance with effective range 1, variance 1 and smoothness 3.5 on a 2D domain $\mathcal{D} = [0, 1]^2$ with sample size 900. For illustration purposes, we show the relative difference of logD at three locations, and include the maxima over all locations in the last t locations in the maxmin ordering. (a) shows that given $n_1 = 63$, the relative difference of log conditional variances converges at $m_1 = 20$. (b) shows that the relative difference of log conditional variances converge at $n_1 = 148$, which in turn results in a corresponding $m_1 = 21$.

2.2.6 Sparsity and computational complexity

The matrix $\mathbf{U}$ is sparse and upper triangular. The columns of $\mathbf{U}$ corresponding to $\mathbf{y}_\ell = (y_1^{(\ell)}, \ldots, y_{n_\ell}^{(\ell)})^\top$ have at most m_ℓ nonzero non-diagonal entries per column, so the computational complexity is $\mathcal{O}(n_\ell m_\ell^3)$. Each z_i may condition on m_ℓ variables at each level $\ell = 1, \ldots, L-1$, but the levels are independent, and so the matrix $Cov(N_{z_i}, N_{z_i})$ in $B_i^{(L)}$ in (2.3) is block-diagonal. Hence, computing the columns of $\mathbf{U}$ corresponding to $\mathbf{z}$ takes at most $\mathcal{O}(n \sum_{\ell=1}^{L-1} m_\ell^3)$ time; however, the actual computing time can be much lower, because for any $i \leq n_\ell$, we can simply use $N_{z_i}^{(\ell)} = \{y_i^{(\ell)}\}$.

Thus, $\mathbf{U}$ is highly sparse and can be calculated quickly. This often also results in a sparse Cholesky factor $\mathbf{V}$ of $\mathbf{U}\mathbf{U}^\top$, based on the use of ordering algorithms such as approximate minimum degree. In addition, in-fill can be avoided completely through the use of an incomplete Cholesky algorithm without introducing significant additional error (Schäfer et al., 2020).

The complexity of each iteration in Algorithm 1 is $\mathcal{O}(tm_\ell^3)$, so overall Algorithm 1 has complexity $\mathcal{O}(tm_\ell^4)$, which is Line 5 in Algorithm 2. Because the step size is doubled at each iteration (Line 15), the overall computational complexity of Algorithm 2 is $\mathcal{O}(tm_\ell^4 \log n)$.

2.3 Derivations and proofs

2.3.1 Computing $\mathbf{U}$

The sparse upper triangular matrix $\mathbf{U}$ can be specified by the following rules:
(1) For each $\ell = 1, 2, ..., L-1$, denote $\mathbf{U}^{(\ell)}$ as the block of $\mathbf{U}$ corresponding to level ℓ with size $n_\ell \times n_\ell$. For each $i = 1, 2, ..., n_\ell$,

$$\mathbf{U}_{ii}^{(\ell)} = (D_i^{(\ell)})^{-1/2}.$$

For the conditioning set of $y_i^{(\ell)}$, suppose the s-th element in its conditioning set is $y_{i'}^{(\ell)}$, then

$$\mathbf{U}_{i'i}^{(\ell)} = -\{B_i^{(\ell)}\}^s (D_i^{(\ell)})^{-1/2}.$$

(2) For the data level L, first denote an $n \times n$ diagonal matrix by

$$\mathbf{U}^{(L)(L)} = diag\left((D_1^{(L)})^{-1/2}, (D_2^{(L)})^{-1/2}, ..., (D_n^{(L)})^{-1/2}\right).$$

Next, for $\ell = 1, 2, ..., L-1$, denote an $n_\ell \times n$ matrix $\mathbf{U}^{(L)(\ell)}$ as the block of $\mathbf{U}$ corresponding to $\{N_{z_i}^{(\ell)}, i = 1, 2, ..., n\}$. Then for each i, suppose the s-th element in $N_{z_i}^{(\ell)}$ is $y_{i'}^{(\ell)}$, then

$$\mathbf{U}_{i'i}^{(L)(\ell)} = -\{B_i^{(L)}\}^s (D_i^{(L)})^{-1/2}.$$

(3) Finally, the matrix $\mathbf{U}$ is

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}^{(1)} & & & & \mathbf{U}^{(L)(1)} \\ & \mathbf{U}^{(2)} & & & \mathbf{U}^{(L)(2)} \\ & & \ddots & & \vdots \\ & & & \mathbf{U}^{(L-1)} & \mathbf{U}^{(L)(L-1)} \\ & & & & \mathbf{U}^{(L)(L)} \end{pmatrix}$$

All unmentioned entries in $\mathbf{U}$ are 0.

2.3.2 Prediction at unobserved locations

For simplicity in the proof, denote $\mathcal{S}$ as the locations corresponding to the knot set at level ℓ. First, we show $\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} = \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}$. When $\ell = 1$, denote $\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{pmatrix}$, where $\mathbf{U}_1 = \mathbf{U}^{(1)}$ is the block of $\mathbf{U}$ corresponding to knot variables at level 1. Then

$$\hat{\mathbf{C}}^{-1} = \mathbf{U}\mathbf{U}^\top = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{0} & \mathbf{U}_3 \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^\top & \mathbf{0} \\ \mathbf{U}_2^\top & \mathbf{U}_3^\top \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1\mathbf{U}_1^\top + \mathbf{U}_2\mathbf{U}_2^\top & \mathbf{U}_2\mathbf{U}_3^\top \\ \mathbf{U}_3\mathbf{U}_2^\top & \mathbf{U}_3\mathbf{U}_3^\top \end{pmatrix}.$$

Since $\hat{\mathbf{C}}^{-1}$ can also be written as $\hat{\mathbf{C}}^{-1} = \begin{pmatrix} C^{(1)}(\mathcal{S},\mathcal{S}) & A \\ A^\top & B \end{pmatrix}^{-1} = \begin{pmatrix} E & F \\ F^\top & G \end{pmatrix}$, by the property of matrix inverse in block form, $C^{(1)}(\mathcal{S},\mathcal{S})^{-1} = E - FG^{-1}F^\top$. Thus we have

$$C^{(1)}(\mathcal{S},\mathcal{S})^{-1} = \mathbf{U}_1\mathbf{U}_1^\top + \mathbf{U}_2\mathbf{U}_2^\top - \mathbf{U}_2\mathbf{U}_3^\top(\mathbf{U}_3\mathbf{U}_3^\top)^{-1}\mathbf{U}_3\mathbf{U}_2^\top = \mathbf{U}_1\mathbf{U}_1^\top = \mathbf{U}^{(1)}\mathbf{U}^{(1)\top}.$$

For any $\ell > 1$, similar results hold: $C^{(\ell)}(\mathcal{S},\mathcal{S})^{-1} = \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}$.

Using Algorithm 2 and achieving a KL divergence of (almost) zero, the MSV approximation based on the knot set $\mathbf{y}_\ell$ at level ℓ is (almost) exact, and so we assume that all information about the process at level ℓ is captured by knot set $\mathbf{y}_\ell$. Then, the posterior mean at an unobserved location $\mathbf{s}_0$ can be computed as

$$\begin{aligned} E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) =& E\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell,\mathbf{z})|\mathbf{z}\right) = E\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell)|\mathbf{z}\right) \\ =& C^{(\ell)}(\mathbf{s}_0,\mathcal{S})\left(C^{(\ell)}(\mathcal{S},\mathcal{S})\right)^{-1}E(\mathbf{y}_\ell|\mathbf{z}) \\ =& C^{(\ell)}(\mathbf{s}_0,\mathcal{S})\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\mathbf{H}_\ell\boldsymbol{\mu}. \end{aligned}$$

The posterior variance can be computed as

$$
\begin{aligned}
& var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{z}) \\
=& E\left(var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell, \mathbf{z})|\mathbf{z}\right) + var\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell, \mathbf{z})|\mathbf{z}\right) \\
=& E\left(var(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell)|\mathbf{z}\right) + var\left(E(y^{(\ell)}(\mathbf{s}_0)|\mathbf{y}_\ell)|\mathbf{z}\right) \\
=& E\left(C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} C^{(\ell)}(\mathcal{S}, \mathbf{s}_0)|\mathbf{z}\right) \\
& + var\left(C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} \mathbf{y}_\ell|\mathbf{z}\right) \\
=& C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} C^{(\ell)}(\mathcal{S}, \mathbf{s}_0) \\
& + C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} var(\mathbf{y}_\ell|\mathbf{z})\left(C^{(\ell)}(\mathcal{S}, \mathcal{S})\right)^{-1} C^{(\ell)}(\mathbf{s}_0, \mathcal{S})^\top \\
=& C^{(\ell)}(\mathbf{s}_0, \mathbf{s}_0) - C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top} C^{(\ell)}(\mathcal{S}, \mathbf{s}_0) \\
& + C^{(\ell)}(\mathbf{s}_0, \mathcal{S})\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\mathbf{\Sigma}_{\mathbf{H}_\ell}\mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top} C^{(\ell)}(\mathbf{s}_0, \mathcal{S})^\top.
\end{aligned}
$$

2.3.3 Proof of proposition

Proposition 2. *For a polynomial $y^{(\ell)}(\mathbf{s}) = \mathbf{p}(\mathbf{s})^\top\boldsymbol{\beta}$ as a function of spatial location $\mathbf{s}$ with p coefficients $\boldsymbol{\beta} \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_\beta)$, the corresponding covariance function $C^{(\ell)}(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{p}(\mathbf{s}_i)^\top\mathbf{\Sigma}_\beta\mathbf{p}(\mathbf{s}_j)$ can be captured exactly by setting the knot and conditioning set to be any distinct p locations.*

Proof of Proposition 2. Denote any p distinct locations as $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_p\}$. For polynomial $y(\mathbf{s}) = \mathbf{p}(\mathbf{s})^\top\boldsymbol{\beta}$ with $\boldsymbol{\beta} \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_\beta)$, the system of equations $\{\mathbf{p}(\mathbf{s}_1)^\top\boldsymbol{\beta} = y(\mathbf{s}_1), \mathbf{p}(\mathbf{s}_2)^\top\boldsymbol{\beta} = y(\mathbf{s}_2), ..., \mathbf{p}(\mathbf{s}_p)^\top\boldsymbol{\beta} = y(\mathbf{s}_p), \mathbf{p}(\mathbf{s})^\top\boldsymbol{\beta} = y(\mathbf{s})\}$ is equivalent to the system of equations $\{\mathbf{p}(\mathbf{s}_1)^\top\boldsymbol{\beta} = y(\mathbf{s}_1), \mathbf{p}(\mathbf{s}_2)^\top\boldsymbol{\beta} = y(\mathbf{s}_2), ..., \mathbf{p}(\mathbf{s}_p)^\top\boldsymbol{\beta} = y(\mathbf{s}_p)\}$, thus $P\left(y(\mathbf{s})|y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_p)\right) = 1$. Then the exact distribution for $\mathbf{y} = (y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_n))$ is

$$
\begin{aligned}
f(\mathbf{y}) =& \prod_{i=1}^{n} f(y(\mathbf{s}_i)|y(\mathbf{s}_{h_i})) \\
=& f(y(\mathbf{s}_1))f(y(\mathbf{s}_2)|y(\mathbf{s}_1))f(y(\mathbf{s}_3)|y(\mathbf{s}_1), y(\mathbf{s}_2)) \cdots f(y(\mathbf{s}_p)|y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_{p-1}))\cdot \\
& \prod_{i=p+1}^{n} f(y(\mathbf{s}_i)|y(\mathbf{s}_1), y(\mathbf{s}_2), ..., y(\mathbf{s}_p)),
\end{aligned}
$$

which equals $\hat{f}(\mathbf{y})$ in Vecchia by setting the knot and conditioning set to be $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_p\}$. Thus the covariance can be captured exactly. □

2.3.4 Proof of theorem

Proof of Theorem 1. The following proof is related to Guinness (2016, Thm. 1). Suppose the true covariance is Σ_0 and the approximated covariance is $\hat{\Sigma}$. By the property of normal distribution, KL divergence of each level ℓ can be written as $KL\left(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)})\right) = \frac{1}{2}E\left(-(\mathbf{y}^{(\ell)})^\top\Sigma_0^{-1}\mathbf{y}^{(\ell)}\right) + \frac{1}{2}E\left((\mathbf{y}^{(\ell)})^\top\hat{\Sigma}^{-1}\mathbf{y}^{(\ell)}\right) + \frac{1}{2}\log\frac{|\hat{\Sigma}|}{|\Sigma_0|}$. Since Σ_0 is the true covariance, then the first term can be written as $E\left(-(\mathbf{y}^{(\ell)})^\top\Sigma_0^{-1}\mathbf{y}^{(\ell)}\right) = -n$. Based on MSV, we have $\log|\hat{\Sigma}| = \sum_{i=1}^n \log D_i^{(\ell)}$. Suppose L_0 is the Cholesky factor of Σ_0, then $E\left((\mathbf{y}^{(\ell)})^\top\hat{\Sigma}^{-1}\mathbf{y}^{(\ell)}\right) = tr(UU^T\Sigma_0) = \sum_{i,j}(L_0^TU)_{ij}^2 = n$. Thus, the KL divergence is $KL\left(f(\mathbf{y}^{(\ell)})||\hat{f}(\mathbf{y}^{(\ell)})\right) = \frac{1}{2}\left(-n + n + \sum_{i=1}^n \log D_i^{(\ell)} - \log|\Sigma_0|\right) = \frac{1}{2}\sum_{i=1}^n \log D_i^{(\ell)} - constant.$ □

2.4 Numerical comparison

We considered simulated data from Gaussian processes with $L = 3$ levels with a Matérn, exponential, and nugget covariance, respectively. We compared the following approaches:

- **Standard:** The original Vecchia approximation (Vecchia, 1988), which from our perspective is a 1-level Vecchia applied directly to the covariance function of the data, obtained by collapsing all levels into one as in (2.1).

- **Latent:** The latent Vecchia approach (e.g., Datta et al., 2016; Katzfuss and Guinness, 2017) is viewed as a 2-level Vecchia approximation, for which the second level must be Gaussian white noise, $\tilde{C}^{(2)} = C^{(L)}$, and so all other levels in our model are collapsed into one, $\tilde{C}^{(1)} = \sum_{\ell=1}^{L-1} C^{(\ell)}$.

- **MSV:** The multi-scale Vecchia approximation proposed in previous sections, here with $L = 3$ levels.

As all approaches can be highly accurate but also slow for large conditioning-set sizes, we compared the KL divergence to the true distribution as a function of computational complexity, which

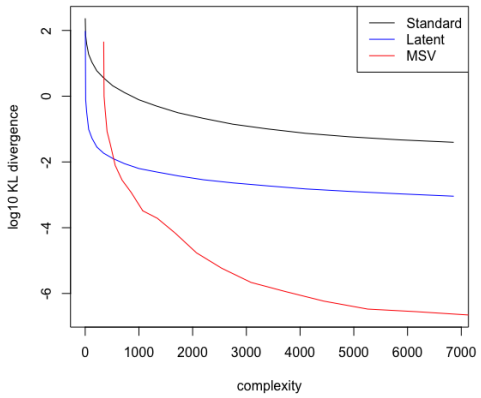


Figure 2.4: Comparison of KL divergence (on a log scale) against computational complexity for simulated data on a one-dimensional domain $\mathcal{D} = [0, 10]$ with $n = 900$ from a 3-level GP with Matérn (smoothness 2.5, variance 1 and effective range 5), exponential (variance 0.3^2 and effective range 2.996), and nugget (0.1^2) covariance.

was taken to be nm^3, $n(m^3 + 1)$, and $n\sum_{\ell=1}^{L-1} m_\ell^3$ for Standard, Latent, and MSV, respectively. (Standard and Latent only use a single conditioning-set size m.)

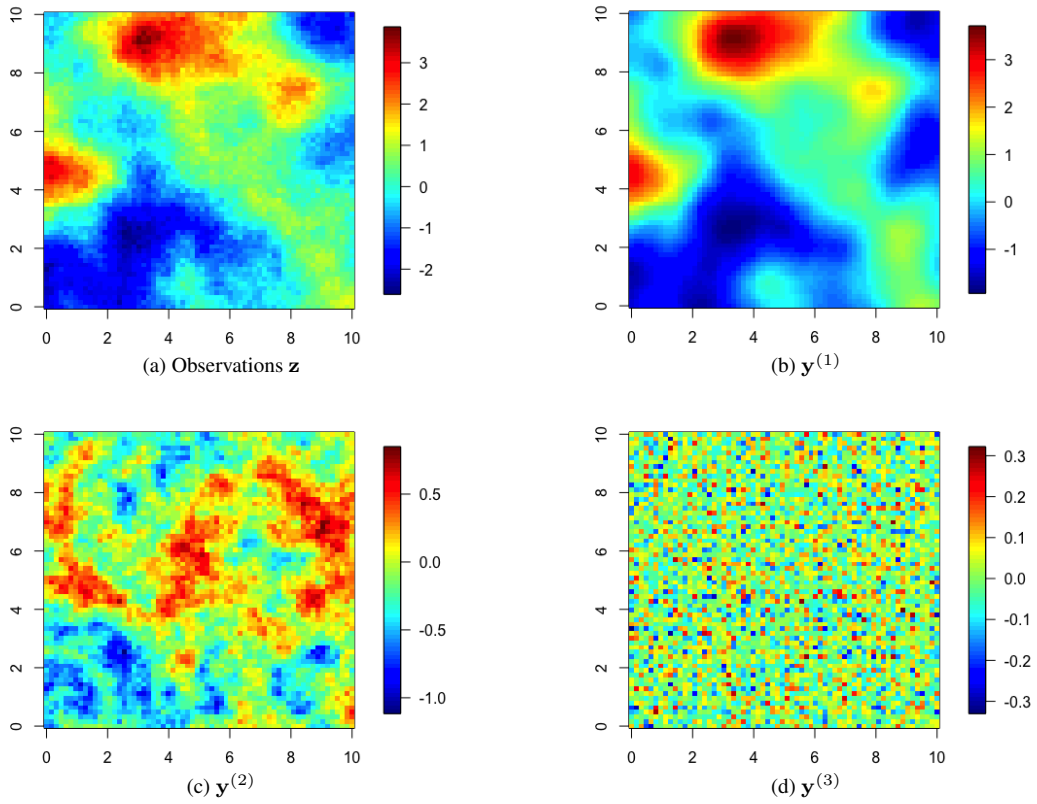
Figure 2.4 shows a comparison on a one-dimensional domain with a relatively small data size of $n = 900$, which allowed us to compute the exact KL divergence. MSV clearly outperformed the other approaches, except for the very-low-complexity setting.

Then, we considered larger datasets of size $n = 6{,}400$ on a two-dimensional domain. One simulated dataset is illustrated in Figure 2.5. Figure 2.6 shows comparisons in terms of KL divergence; to avoid the high computational cost of repeatedly calculating the exact KL divergence, the KL was approximated by subtracting each method's loglikelihood from the loglikelihood for MSV with the largest possible conditioning sets.

MSV was again more accurate than Latent for a given computational complexity, and both methods strongly outperformed Standard Vecchia.

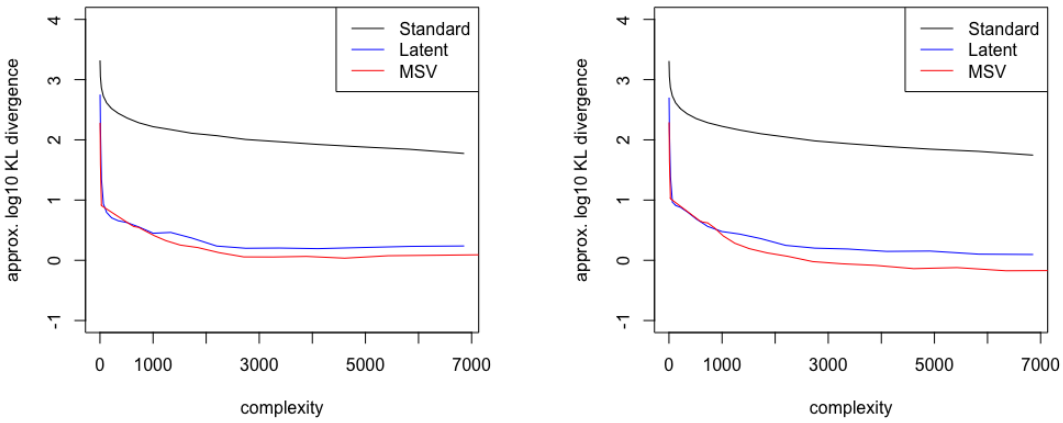
2.5 Application

We applied the MSV method to 148,309 satellite measurements of daytime land-surface temperatures from Heaton et al. (2019). The observations are Level-3 data obtained on August 4, 2016, by the Terra instrument onboard the MODIS satellite, over a latitudinal range of 34.29519



(a) Observations $\mathbf{z}$ (b) $\mathbf{y}^{(1)}$ (c) $\mathbf{y}^{(2)}$ (d) $\mathbf{y}^{(3)}$

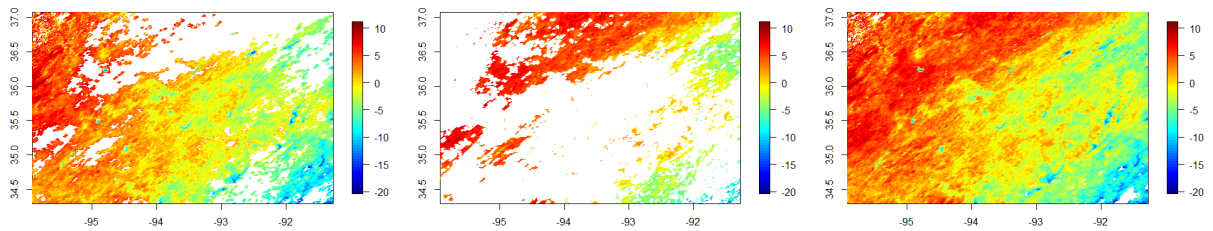
Figure 2.5: 2D example of (a) observations $\mathbf{z}$ of a three-scale process based on components with (b) Matérn (smoothness 2.5, variance 1 and effective range 5), (c) exponential (variance 0.3^2 and effective range 3), and (d) nugget(0.1^2) covariance, respectively, on a two-dimensional domain $\mathcal{D} = [0, 10]^2$.



(a) Matérn effective range $\lambda_1 = 5$ (b) Matérn effective range $\lambda_1 = 8$

Figure 2.6: Comparison of KL divergence (on a log scale) against computational complexity for simulated data on a two-dimensional domain $\mathcal{D} = [0, 10]^2$ with $n = 6{,}400$ from a 3-level GP with Matérn (smoothness 2.5, variance 1, and effective range 5 in (a) and 8 in (b)), exponential (variance 0.3^2 and effective range 2.996), and nugget (0.1^2) covariance

to 37.06811, and a longitude range from -95.91153 to -91.28381. According to the split in Heaton et al. (2019), the training dataset has 105,569 observations, and the testing dataset has 42,740 observations. We considered the centered data obtained by subtracting an overall (constant) mean, which are shown in Figure 2.7. For these centered data, we assumed a 3-level Gaussian process model with mean zero and with Matérn, exponential, and nugget covariance, with six unknown parameters. As in Heaton et al. (2019), the parameters were estimated based on a subsample of size 2,500, resulting in the following parameter estimates: for the Matérn level, variance 19.8656, range 0.3573, smoothness 4.9894; for the exponential level, variance 2.6772, range 0.0665; and nugget variance 0.6917. The resulting covariance function is illustrated in Figure 2.8.



(a) training data (b) testing data (c) full data

Figure 2.7: Centered daytime land surface temperature data measured on August 4, 2016 by the Terra instrument onboard the MODIS satellite

For the full training dataset, we used Algorithm 2, with $m_{\max} = 30$, $\varepsilon = 0.001$, and $T = 1000$ (i.e., the last 1000 locations in the maxmin ordering). The algorithm then selected $m_1 = 13, n_1 = 16383$ for level 1 (Matérn), and $m_2 = 23, n_2 = 105569$ for level 2 (exponential covariance). Given these knots and conditioning sets, we computed the posterior predictive distribution using MSV at each level. Note that both training data and testing data were noisy observations, with the true underlying process are unknown. Hence, MSV predictions were obtained at both training locations and testing locations.

The prediction results are shown in Figure 2.9. We can see that the first level (Figure 2.9(a)) captures the large-scale spatial dependence, and the second level (Figure 2.9(b)) captures smaller-

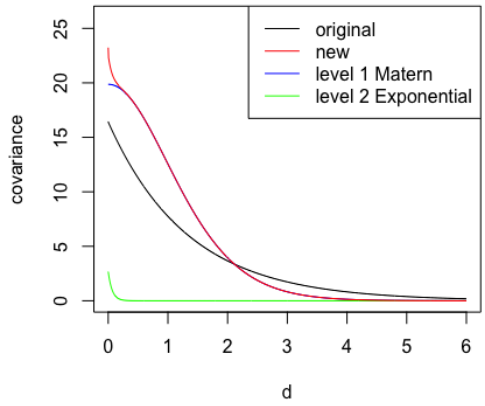
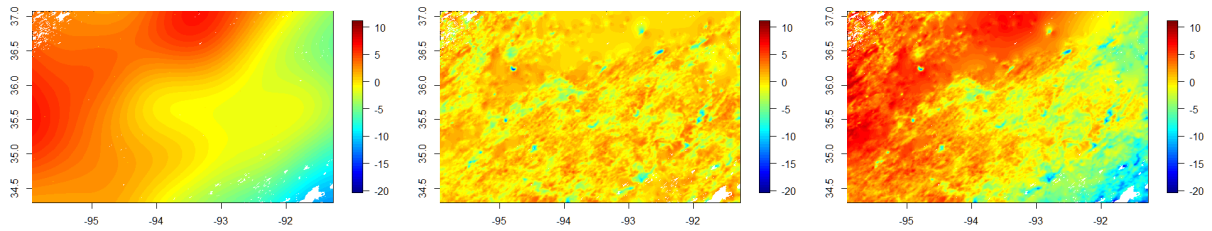


Figure 2.8: Illustration of the first two levels of the new estimated 3-level covariance function as a function of distance. The original exponential covariance (black curve) was estimated in Heaton et al. (2019), with an estimated variance of 16.40771 and a range of 4/3.



(a) prediction at level 1 (b) prediction at level 2 (c) prediction level 1 + level 2

Figure 2.9: MSV predictions for MODIS temperature data

scale spatial dependence. The overall predicted values (i.e., level 1 plus level 2) given in Figure 2.9(c) is very close to the full original observations in Figure 2.7(c).

As the dataset considered here is the same one used for the comparison study of many recent approaches for analyzing large spatial dataset in Heaton et al. (2019), we conducted a comparison between our MSV prediction accuracy on the test data and the top six methods' accuracy studied in Heaton et al. (2019). We considered mean absolute error (MAE) and root mean squared error (RMSE). The results, given in Table 2.1, indicate that the multi-level approach (MSV) was highly competitive with the leading existing methods in Heaton et al. (2019).

Method	MAE	RMSE
MSV	1.11	1.42
LatticeKrig	1.22	1.68
MRA	1.33	1.85
NNGP	1.21	1.64
Partition	1.41	1.80
SPDE	1.10	1.53
Periodic Embedding	1.29	1.79

Table 2.1: Comparison of prediction accuracy scores for MSV and top methods from Heaton et al. (2019) on the MODIS temperature data

2.6 Discussion

We proposed a multi-scale Vecchia (MSV) approximation of Gaussian processes for analysis of the multi-scale phenomena. Our MSV method can tailor a suitable Vecchia approximation to the process acting at each underlying scale. The increasingly small scales of spatial variation can be captured by increasingly large sets of variables, and then an accurate approximation of the spatial dependence is obtained from very large to very fine scales. We conducted inference using the MSV method, explored approximation properties, and provided an algorithm for automatic choice of the number of knot variables and the conditioning set size at each level. We compared our method to existing variants of the Vecchia approximation using simulated data. In an application to MODIS daytime land-surface temperature data, our multi-scale method exhibited highly competitive performance relative to a large set of existing approaches for analyzing large spatial datasets in Heaton et al. (2019). Our method also leads to nice visualizations of the different scales, which can be highly useful in many scientific contexts.

Our algorithm for determining tuning parameters for the Vecchia approximations at different levels or scales is also applicable and useful for single-level (Vecchia, 1988) or two-level (Katzfuss and Guinness, 2017) Vecchia approximations. While we have assumed here for simplicity that the data are obtained as an unweighted sum of the latent processes at the different scales, extending our methodology to observations that are modeled as (different) linear combinations of the individual

scales (including some with zero weight) is straightforward. Our method could also be combined with compositional kernel search (Duvenaud et al., 2013), which expresses the covariance function or kernel of a GP as a sum of kernels, which are obtained using a greedy search over sums and products of a number of base kernels.

3. LOCALLY ANISOTROPIC COVARIANCE FUNCTIONS ON THE SPHERE

3.1 Introduction

Traditionally, geostatistical analysis relied on approximating small or regional spatial domains as flat, as subsets of $\mathbb{R}^2$. However, as satellites have enabled the collection of global data, there is an increased need for covariance functions that are valid on spheres. Thus, our focus here is on spatial data observed on the unit 2-sphere, $\mathbb{S} = \{\tilde{\mathbf{s}} \in \mathbb{R}^3 : \|\tilde{\mathbf{s}}\| = 1\}$.

For processes on a sphere $\mathbb{S}$, two different measures of distance are commonly used. The most natural distance is great-arc (or great-circle) distance, which measures the distance "going along the surface of the sphere." In contrast, chordal distance or Euclidean distance "pierces through the sphere." The relationship between chordal distance and great-arc distance on $\mathbb{S}$ is given by,

$$ch = 2\sin(ga/2), \tag{3.1}$$

where ga is the great-arc distance between two locations on $\mathbb{S}$, and ch is the corresponding chordal (or Euclidean) distance. Thus, any function of chordal distance is also a function of great-circle distance, and vice versa.

It is not trivial to build great-arc-distance-based covariance functions on the sphere because of the curvature of $\mathbb{S}$ (e.g., Jones, 1963). Most well-known classes of covariance functions are valid (i.e., positive definite) only when used with Euclidean (or Mahalanobis) distance on $\mathbb{R}^d$, for some (or all) $d \geq 1$. They are not guaranteed to be valid when used with great-arc distance (e.g. Gneiting, 2013). For example, the Matérn correlation is only valid with great-arc distance for smoothness up to 0.5 (i.e., for $\nu \leq 0.5$). Some valid covariance functions based on great-circle distance were proposed by Das (2000) in closed form, but it is difficult to use these covariance functions in most applications due to unrealistic smoothness assumptions (Stein, 1999).

Much of the work on covariance functions with great-arc distance has focused on isotropic covariance functions. Huang et al. (2011) summarize the validity of commonly used covariance

and variogram functions on the sphere. Gneiting (2013) further developed characterizations and constructions of isotropic positive-definite functions on the sphere, and proved that under a natural support condition, many valid isotropic functions in $\mathbb{R}^3$ allow for the replacement of the chordal distance by the great-arc distance on the sphere. Guinness and Fuentes (2016) study properties of isotropic covariance functions on the sphere. Another class of literature is devoted to deriving new covariance functions on sphere, including stationary and isotropic covariance matrix structures of Gaussian vector random fields on the sphere by Ma (2012, 2015), and parametric isotropic variogram matrix models on the sphere by Du et al. (2013).

Current approaches to modeling nonstationary, anisotropic processes on a sphere are reviewed in Jeong et al. (2017), including the differential operator approach (Jun and Stein, 2007, 2008; Jun, 2014), spherical harmonic representation (Stein et al., 2007), stochastic partial differential equations (Lindgren et al., 2011), kernel convolution (Heaton et al., 2014), and multi-step spectrum (Castruccio et al., 2013; Castruccio and Genton, 2014, 2016). Hitczenko and Stein (2012) investigate some properties of a class of anisotropic processes that are invariant to shifts in longitude on spheres.

We consider here using Euclidean distance and restrict a valid covariance function in $\mathbb{R}^3$ to $\mathbb{S}$ according to Yaglom (1987). This is guaranteed to be valid, in that we know the covariance function is valid for points in $\mathbb{R}^3$, it just so "happens" that all locations now fall onto the surface of our sphere within this space. In addition, results based on chordal and great-arc distance are often indistinguishable (Guinness and Fuentes, 2016). This makes intuitive sense because $\sin(x) \approx x$ for small x, and so for two points on the sphere the covariance is either (close to) zero (if the points are far apart) or the two points have approximately equal chordal distance and great-arc distance as in (3.1) (if they are close together). Matérn functions based on chordal distance also have the expected differentiability properties (Guinness and Fuentes, 2016).

We develop nonstationary, locally anisotropic covariance functions on the sphere based on the locally anisotropic covariance functions for Euclidean space proposed in Paciorek and Schervish (2006). Similar ideas were previously considered in Katzfuss (2011) and Knapp (2012). We de-

scribe how to construct these anisotropic covariance functions in general, and explore properties of the parameterization. The approach is very general, and we provide theorems and conditions such that the resulting correlation function is isotropic or axially symmetric, for sensible parameterizations in specific applications. For modern large datasets on the sphere, the Vecchia approximation (Vecchia, 1988) is applied to achieve computational feasibility and conduct Bayesian parameter inference.

The remainder of this chapter is organized as follows. Section 3.2 reviews a nonstationary correlation function on $\mathbb{R}^d$. In Section 3.3, we construct nonstationary covariance functions on the sphere, and provide theorems and conditions for important special cases such as isotropy and axial symmetry. Section 3.4 reviews the Vecchia approximation for large spatial datasets. Section 3.5 contains proofs of theorems. In Section 3.6, we illustrate Gaussian process realizations using various covariance functions constructed by our method, and provide comparisons in numerical studies. We conclude in Section 3.7.

3.2 Review: A nonstationary correlation function on $\mathbb{R}^d$

Paciorek and Schervish (2006) showed that for any isotropic correlation function ρ that is positive definite on $\mathbb{R}^d$ for all $d \in \mathbb{N}$, a valid nonstationary correlation function with spatially varying anisotropy can be obtained as

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

where

$$q(\mathbf{s}_i, \mathbf{s}_j) = \{2(\mathbf{s}_i - \mathbf{s}_j)'(\boldsymbol{\Sigma}(\mathbf{s}_i) + \boldsymbol{\Sigma}(\mathbf{s}_j))^{-1}(\mathbf{s}_i - \mathbf{s}_j)\}^{1/2} \tag{3.2}$$

is the Mahalanobis distance with respect to the averaged matrix $(\boldsymbol{\Sigma}(\mathbf{s}_i) + \boldsymbol{\Sigma}(\mathbf{s}_j))/2$,

$$c(\mathbf{s}_i, \mathbf{s}_j) := |\boldsymbol{\Sigma}(\mathbf{s}_i)|^{1/4}|\boldsymbol{\Sigma}(\mathbf{s}_j)|^{1/4}|(\boldsymbol{\Sigma}(\mathbf{s}_i) + \boldsymbol{\Sigma}(\mathbf{s}_j))/2|^{-1/2}, \tag{3.3}$$

is a normalization term, and $\boldsymbol{\Sigma}(\mathbf{s})$ for each $\mathbf{s}$ is a positive definite $d \times d$ matrix that determines the

local rotation and scaling of the space at location $\mathbf{s}$.

3.3 Classes of nonstationary covariance functions on the sphere

3.3.1 Construction of the covariance functions

We now develop nonstationary covariance functions on the sphere obtained by simply using a valid nonstationary covariance function (in $\mathbb{R}^3$) based on Paciorek and Schervish (2006). Since we choose the chordal distance as a measurement of distance, the first step is to convert longitude-latitude coordinates to (x, y, z)-coordinates of a 3-dimensional Cartesian coordinate system. For a given unit sphere $\mathbb{S}$, the origin $(0, 0, 0)$ of a Cartesian coordinate system is determined by the centre of $\mathbb{S}$. The direction of the x-axis is defined by a vector, which is from the sphere's center to the (0° longitude, 0° latitude) point. The direction of the y-axis is defined by a vector, which is from the sphere's center to the (90° longitude, 0° latitude) point. Then the z-axis is determined accordingly by x-axis and y-axis. For any point $\mathbf{s} = (l, L)$ with longitude l and latitude L on $\mathbb{S}$, the corresponding coordinates $\tilde{\mathbf{s}} = (x, y, z)$ in the Cartesian coordinate system are

$$
\begin{aligned}
x &= \cos(L)\cos(l), \\
y &= \cos(L)\sin(l), \\
z &= \sin(L).
\end{aligned}
$$

For example, (0° longitude, 0° latitude) on the sphere is converted to $(1, 0, 0)$ in the Cartesian coordinate system. We illustrate the coordinate conversion in Figure 3.1.

To construct a nonstationary correlation function based on Section 3.2, we need to determine the anisotropy matrix $\mathbf{\Sigma}(\cdot)$. It can be parameterized by d scaling parameters and $d - 1$ rotation parameters for d-dimensional Euclidean space (see, e.g., Banerjee et al., 2008). Given the constraint that $\mathbb{S}$ is a 2-dimensional surface embedded in $\mathbb{R}^3$, we prefer to use only two local scaling parameters and one local rotation parameter to parameterize $\mathbf{\Sigma}(\cdot)$.

We first parameterize $\mathbf{\Sigma}(\cdot)$ at the Euclidean coordinate $\tilde{\mathbf{c}} := (1, 0, 0)$, then extend it to $\mathbf{\Sigma}(\cdot)$ at any point. Since the sphere is locally flat around $\tilde{\mathbf{c}} = (1, 0, 0)$ (spherical coordinates $\mathbf{c} = (0, 0)$)

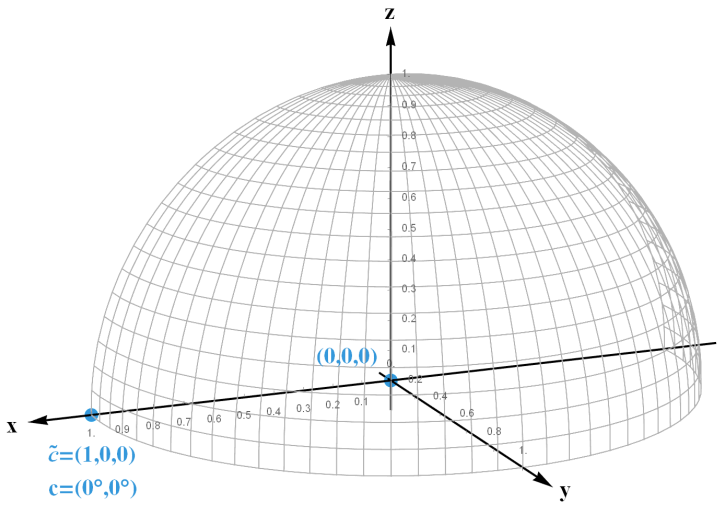


Figure 3.1: A part of a unit sphere displayed in the Cartesian coordinate system. The centre of the sphere is located at the origin $(0, 0, 0)$.

and parallel to the $y - z$ plane, the correlation length in the y-direction and z-direction can be specified by two scaling parameters, $\gamma_1(\mathbf{c}) > 0$ and $\gamma_2(\mathbf{c}) > 0$, respectively (illustrated in Figure 3.2). Denote $\boldsymbol{\gamma}(\mathbf{c}) := (\gamma_1(\mathbf{c}), \gamma_2(\mathbf{c}))'$, then the local scaling matrix at point $\tilde{\mathbf{c}}$ is

$$\mathbf{D}(\boldsymbol{\gamma}(\mathbf{c})) := diag\{1, \gamma_1(\mathbf{c}), \gamma_2(\mathbf{c})\}.$$

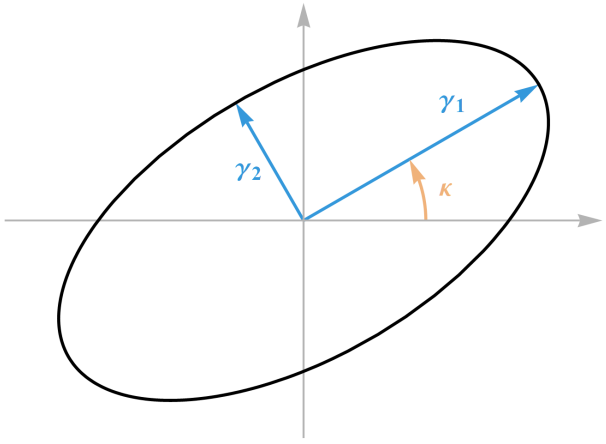


Figure 3.2: Illustration of the scaling and rotation parameters at point $\tilde{\mathbf{c}}$

To describe the rotation at $\tilde{\mathbf{c}}$, we define a rotation parameter $\kappa(\mathbf{c}) \in [0, \pi/2)$ and a rotation matrix $\mathcal{R}_x(\kappa(\mathbf{c}))$ by

$$\mathcal{R}_x(\kappa(\mathbf{c})) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\kappa(\mathbf{c}) & -\sin\kappa(\mathbf{c}) \\ 0 & \sin\kappa(\mathbf{c}) & \cos\kappa(\mathbf{c}) \end{pmatrix},$$

which rotates the (y, z)-coordinate system at $\tilde{\mathbf{c}}$ about the x-axis. With the scaling parameters $\boldsymbol{\gamma}(\mathbf{c})$ and rotation parameter $\kappa(\mathbf{c})$, the local anisotropy matrix at $\tilde{\mathbf{c}}$ (corresponding spherical coordinates $\mathbf{c}$) is then given by,

$$\tilde{\boldsymbol{\Sigma}}(\mathbf{c}) := \mathcal{R}_x(\kappa(\mathbf{c}))\mathbf{D}(\boldsymbol{\gamma}(\mathbf{c}))\mathcal{R}_x(\kappa(\mathbf{c}))'. \tag{3.4}$$

Based on $\tilde{\boldsymbol{\Sigma}}(\mathbf{c})$, we now derive the anisotropy matrix at any point on the sphere. Define a rotation matrix $\mathcal{R}_y(\theta)$, which rotates a vector in $\mathbb{R}^3$ by angle θ about the y-axis, by

$$\mathcal{R}_y(\theta) := \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix},$$

and a rotation matrix $\mathcal{R}_z(\theta)$, which rotates a vector in $\mathbb{R}^3$ by angle θ about the z-axis, by

$$\mathcal{R}_z(\theta) := \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then for any spherical location $\mathbf{s} = (l, L)$ (with corresponding Euclidean coordinates $\tilde{\mathbf{s}}$), we rotate $\tilde{\mathbf{s}}$ about the y-axis and z-axis to the point $\tilde{\mathbf{c}}$ by writing

$$\tilde{\mathbf{c}} = \mathcal{R}_y(-L)\mathcal{R}_z(-l)\tilde{\mathbf{s}}.$$

.

Let $\gamma_1(\mathbf{s})$ and $\gamma_2(\mathbf{s})$ be the scaling parameters at $\mathbf{s}$, and $\kappa(\mathbf{s})$ be the rotation parameter. To derive a quadratic form as in (3.2), we combine these rotation matrices with the anisotropy matrix for $\mathbf{c}$ in (3.4), and obtain

$$\tilde{\boldsymbol{\Sigma}}(\mathbf{s}) := \mathcal{R}_x(\kappa(\mathbf{s}))\mathbf{D}(\boldsymbol{\gamma}(\mathbf{s}))\mathcal{R}_x(\kappa(\mathbf{s}))',$$

$$\tilde{\mathbf{c}}'\tilde{\boldsymbol{\Sigma}}(\mathbf{s})^{-1}\tilde{\mathbf{c}} = \Big(\mathcal{R}_y(-L)\mathcal{R}_z(-l)\,\tilde{\mathbf{s}}\Big)'\tilde{\boldsymbol{\Sigma}}(\mathbf{s})^{-1}\Big(\mathcal{R}_y(-L)\mathcal{R}_z(-l)\,\tilde{\mathbf{s}}\Big) =: \tilde{\mathbf{s}}'\boldsymbol{\Sigma}(\mathbf{s})^{-1}\tilde{\mathbf{s}},$$

where

$$\boldsymbol{\Sigma}(\mathbf{s}) := \mathcal{R}_z(l)\mathcal{R}_y(L)\tilde{\boldsymbol{\Sigma}}(\mathbf{s})\mathcal{R}_y(L)'\mathcal{R}_z(l)'$$

is an anisotropy matrix for spherical domains (i.e., when $\mathcal{D} \subseteq \mathbb{S}$) and can be used in (3.2) and (3.3) to construct a nonstationary correlation function of the form

$$\rho_{NS}(\mathbf{s}_i, \mathbf{s}_j) = c(\mathbf{s}_i, \mathbf{s}_j)\rho\big(q(\mathbf{s}_i, \mathbf{s}_j)\big), \tag{3.5}$$

where ρ is an isotropic correlation function that is positive definite in $\mathbb{R}^d$ for all $d = 1, 2, \ldots$.

Then a valid covariance function is obtained by setting

$$C(\mathbf{s}_i, \mathbf{s}_j) = \sigma(\mathbf{s}_i)\sigma(\mathbf{s}_j)\rho_{NS}(\mathbf{s}_i, \mathbf{s}_j),$$

where the standard deviation $\sigma(\cdot) > 0$ can also vary over space.

3.3.2 Properties

While the approach above is very general, we will typically need a sensible parameterization (i.e., special case) of this approach for specific applications. It is relatively straightforward to obtain such special cases to achieve desired effects. We illustrate this here by providing conditions such that the resulting correlation function is isotropic or axially symmetric.

An isotropic covariance function is a function of only distance between two locations. Due to the one-to-one relationship between great-arc and chordal distance in (3.1), isotropic covariance functions on the sphere are isotropic with respect to chordal and great-arc distance. By adding

constraints to the scaling parameters $\gamma_1(\mathbf{s})$ and $\gamma_2(\mathbf{s})$, the correlation function ρ_{NS} in (3.5) can be forced to be isotropic on the sphere:

Theorem 3. *The correlation function ρ_{NS} in* (3.5) *is isotropic (i.e., depends only on distance) if $\gamma_1(\mathbf{s}) = \gamma_2(\mathbf{s}) \equiv \gamma$ is constant.*

All proofs can be found in Section 3.5.

A subclass of covariance functions that is specifically useful for spherical domains is that of axially symmetric covariance functions (e.g., Stein et al., 2007), which for a pair of locations on the sphere, depend on both latitudes explicitly, but only depend on the longitudes through the difference between them:

Definition 4. *A covariance function $C : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$ is called axially symmetric if there exists a function C_A such that*

$$C(\mathbf{s}_i, \mathbf{s}_j) = C_A(l_i - l_j, L_i, L_j),$$

where $\mathbf{s}_i = (l_i, L_i)$ and $\mathbf{s}_j = (l_j, L_j)$ with longitudes l_i and l_j and latitudes L_i and L_j on $\mathbb{S}$.

Under certain constraints, our general nonstationary correlation function can be forced to be axially symmetric:

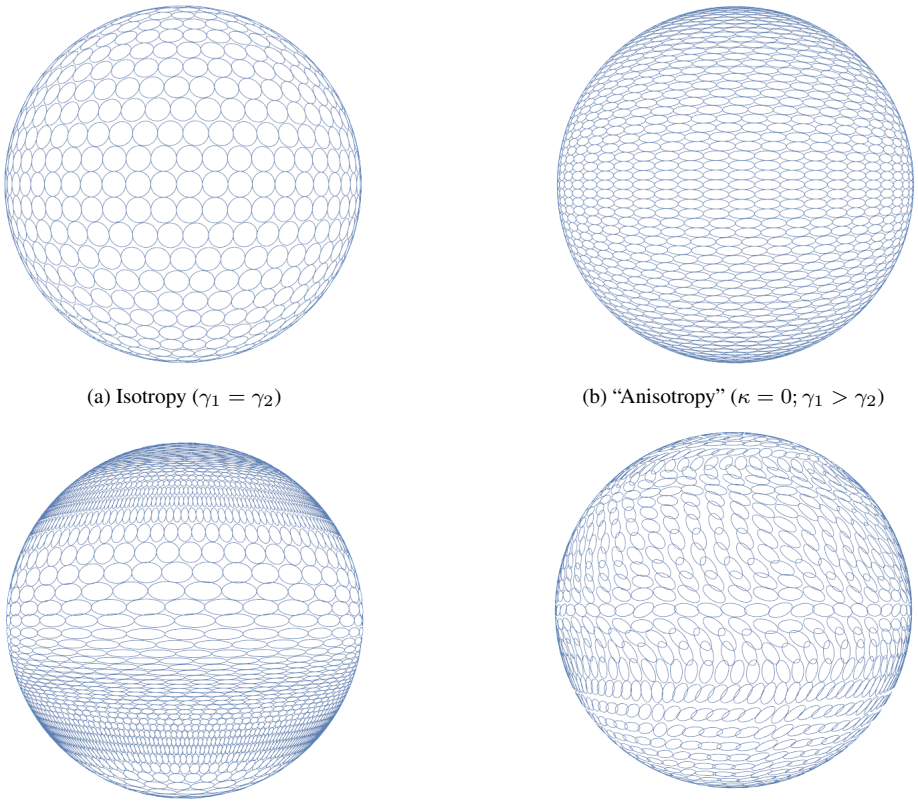
Theorem 5. *The correlation function ρ_{NS} in* (3.5) *is axially symmetric if $\kappa(\mathbf{s}) \equiv 0$ and $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ are functions of latitude only (i.e., they do not depend on longitude).*

Different versions and special cases of our general nonstationary covariance function are illustrated in Figure 3.3, with different settings for the scaling parameters $\gamma_1(\mathbf{s})$ and $\gamma_2(\mathbf{s})$ and the rotation parameter $\kappa(\mathbf{s})$.

3.3.3 Example: A nonstationary Matérn covariance on the sphere

The Matérn correlation function is highly popular in geospatial analysis. It is valid in $\mathbb{R}^d$ for any $d \in \mathbb{N}$ and given by

$$\mathcal{M}_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} r^\nu \mathcal{K}_\nu(r), \qquad r \geq 0,$$



(a) Isotropy ($\gamma_1 = \gamma_2$)

(b) "Anisotropy" ($\kappa = 0; \gamma_1 > \gamma_2$)

(c) Axial symmetry

(d) General nonstationarity with spatially varying rotation $\kappa(\cdot)$

Figure 3.3: Illustration of special cases of the nonstationary class of correlation functions in (3.5) via correlation contours for a set of locations on the sphere.

where $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function of order $\nu > 0$.

Following the construction approach shown in Section 3.3.1, a nonstationary locally anisotropic Matérn covariance function is

$$\mathcal{M}_{NS}(\mathbf{s}_i, \mathbf{s}_j) = \sigma(\mathbf{s}_i)\sigma(\mathbf{s}_j)c(\mathbf{s}_i, \mathbf{s}_j)\mathcal{M}_{(\nu(\mathbf{s}_i)+\nu(\mathbf{s}_j))/2}(q(\mathbf{s}_i, \mathbf{s}_j)), \tag{3.6}$$

where, in addition to the standard deviation $\sigma(\cdot) > 0$, the local scaling and rotation through a $d \times d$ positive definite anisotropy matrix $\mathbf{\Sigma}(\cdot)$, the smoothness parameter $\nu(\cdot)$ can also vary over space (here, over the sphere) as shown in Stein (2005).

Guinness and Fuentes (2016) showed that the local smoothness properties of the Matérn covariance are preserved when restricting a process on Euclidean space to the sphere; for example, a

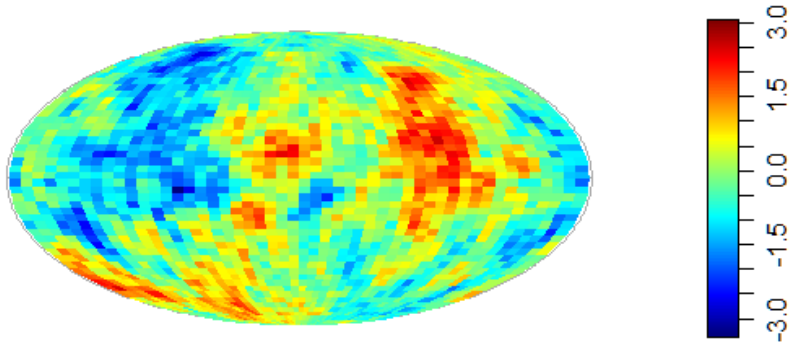


Figure 3.4: A realization from Gaussian process with mean zero and the nonstationary Matérn covariance on sphere, on a regular grid (longitude $\times$ latitude) of size $50 \times 50 = 2{,}500$. We used smoothness $\nu(\mathbf{s}) \equiv 0.5$, Matérn range = 0.668, rotation parameter $\kappa(\mathbf{s}) \equiv 0$, and scaling parameters $\gamma_1(\mathbf{s}) = exp(-0.7 + 0.35\sin(s_1) + 0.44s_2)$, $\gamma_2(\mathbf{s}) = exp(-1.2 + 0.25\sin(s_1) + 0.44s_2)$, where s_1 and s_2 denote longitude and latitude in radians, respectively.

Gaussian process (GP) with covariance function $\mathcal{M}_{NS}$ has m mean square derivatives at $\mathbf{s}$ if and only if $\nu(\mathbf{s}) > m$.

A realization of a GP with the resulting nonstationary Matérn covariance is illustrated in in Figure 3.4.

3.4 Vecchia approximation

For many modern large datasets, including on the sphere, direct application of Gaussian processes is too computationally expensive, as the cost scales cubically in the number of observed data. The approximation proposed by Vecchia (1988) has become highly popular in recent years, which has low computational complexity and reduces the memory burden, but still presents high accuracy according to the KL divergence from the true process (e.g., Guinness, 2018; Katzfuss and Guinness, 2017). In Vecchia approximation, the high-dimensional joint distribution of the entire data vector is substituted by a product of univariate conditional distributions. The conditioning set for each univariate conditional distribution consists of a small subset of previously ordered variables.

Consider a Gaussian process $y(\cdot) \sim GP(0, C)$ on a spatial region $\mathcal{D}$ with covariance function

C, define the history set of i as $h(i) = \{1, 2, ..., i-1\}$ and $h(1) = \emptyset$, then $\mathbf{y}_{h(i)} = (y_1, ..., y_{i-1})'$. The exact distribution of the observed vector $\mathbf{y} = (y_1, y_2, ..., y_n)$ is

$$f(\mathbf{y}) = \prod_{i=1}^{n} f(y_i|\mathbf{y}_{h(i)}).$$

The history set $h(i)$ is substituted by a subset $g(i) \subset h(i)$ in Vecchia approximation, where $g(i)$ consists of the indices of the nearest m data points to the ith data. Then the approximated joint density is obtained by:

$$\hat{f}(\mathbf{y}) = \prod_{i=1}^{n} f(y_i|\mathbf{y}_{g(i)}). \tag{3.7}$$

There are various choices in ordering locations and the conditioning sets (Katzfuss and Guinness, 2017). Considering the approximation accuracy, we will use nearest-neighbor conditioning and maximum-minimum-distance ordering (Guinness, 2018) in the numerical studies, both based on the Euclidean locations. Correlation-based ordering and conditioning that takes into account the potential nonstationary structure (e.g., Katzfuss et al., 2020b) will be considered as future work. Aside from likelihood-based parameter inference using the Vecchia likelihood in (3.7), the Vecchia approximation can also be applied to the joint distribution of GP at prediction locations and observed locations, in order to obtain highly accurate approximations of the (joint) posterior predictive distributions (e.g., Katzfuss et al., 2020a). When data is noisy, the Vecchia approximation can be applied to latent (noise-free) GP as before, and then combined with an incomplete-Cholesky decomposition of the posterior precision matrix to preserve a low computational complexity (Schäfer et al., 2020). We implemented Vecchia inference based on our new covariance function by extending the `R` package `GPvecchia` (Katzfuss et al., 2020c).

3.5 Proofs of theorems

Proof of Theorem 3. If $\gamma_1(\mathbf{s}) = \gamma_2(\mathbf{s}) \equiv \gamma$ is constant, then

$$\mathbf{D}(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & \gamma \end{pmatrix},$$

$$\begin{aligned}
\tilde{\mathbf{\Sigma}}(\mathbf{s}) &= \mathcal{R}_x(\kappa(s))\mathbf{D}(\gamma(\mathbf{s}))\mathcal{R}_x(\kappa(s))' \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\kappa(s) & -\sin\kappa(s) \\ 0 & \sin\kappa(s) & \cos\kappa(s) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & \gamma \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\kappa(s) & \sin\kappa(s) \\ 0 & -\sin\kappa(s) & \cos\kappa(s) \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & \gamma \end{pmatrix}.
\end{aligned}$$

To compute $\mathbf{\Sigma}(\mathbf{s}) = \mathcal{R}_z(s_1)\mathcal{R}_y(s_2)\tilde{\mathbf{\Sigma}}(\mathbf{s})\mathcal{R}_y(s_2)'\mathcal{R}_z(s_1)'$, we first compute

$$\begin{aligned}
\mathbf{A} := \mathcal{R}_z(s_1)\mathcal{R}_y(s_2) &= \begin{pmatrix} \cos(s_1) & -\sin(s_1) & 0 \\ \sin(s_1) & \cos(s_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(s_2) & 0 & \sin(s_2) \\ 0 & 1 & 0 \\ -\sin(s_2) & 0 & \cos(s_2) \end{pmatrix} \\
&= \begin{pmatrix} \cos(s_1)\cos(s_2) & -\sin(s_1) & \cos(s_1)\sin(s_2) \\ \sin(s_1)\cos(s_2) & \cos(s_1) & \sin(s_1)\sin(s_2) \\ -\sin(s_2) & 0 & \cos(s_2) \end{pmatrix},
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\Sigma}(\mathbf{s}) = \mathbf{A}\tilde{\boldsymbol{\Sigma}}(\mathbf{s})\mathbf{A}' &= \begin{pmatrix} (1-\gamma)x^2+\gamma & (1-\gamma)xy & (1-\gamma)xz \\ (1-\gamma)xy & (1-\gamma)y^2+\gamma & (1-\gamma)yz \\ (1-\gamma)xz & (1-\gamma)yz & (1-\gamma)z^2+\gamma \end{pmatrix} \\
&= (1-\gamma)\begin{pmatrix} x \\ y \\ z \end{pmatrix}\begin{pmatrix} x & y & z \end{pmatrix} + \gamma\mathbf{I}_3 \\
&= (1-\gamma)\tilde{\mathbf{s}}\tilde{\mathbf{s}}' + \gamma\mathbf{I}_3,
\end{aligned}$$

where $x = \cos(s_2)\cos(s_1)$, $y = \cos(s_2)\sin(s_1)$, $z = \sin(s_2)$ are the coordinates of a Cartesian coordinate system. Then

$$\begin{aligned}
|\boldsymbol{\Sigma}(\mathbf{s})| &= det\{(1-\gamma)\,\tilde{\mathbf{s}}\tilde{\mathbf{s}}' + \gamma\mathbf{I}_3\} \\
&= \gamma^3 \cdot det\left\{\mathbf{I}_3 + \frac{1-\gamma}{\gamma}\tilde{\mathbf{s}}\tilde{\mathbf{s}}'\right\} \\
&= \gamma^3 \cdot det\left\{1 + \frac{1-\gamma}{\gamma}\tilde{\mathbf{s}}'\tilde{\mathbf{s}}\right\} \\
&= \gamma^3 \cdot det\left\{1 + \frac{1-\gamma}{\gamma}\cdot 1\right\} \\
&= \gamma^2
\end{aligned}$$

is a constant with respect to $\mathbf{s}$. And for $i \neq j$,

$$(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1} = \left((1-\gamma)\tilde{\mathbf{s}}_\mathbf{i}\tilde{\mathbf{s}}_\mathbf{i}' + (1-\gamma)\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + 2\gamma\mathbf{I}_3\right)^{-1}.$$

WLOG, we ignore the constant coefficients inside the inverse, and then

$$\begin{aligned}
(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1} &= \left(\tilde{\mathbf{s}}_\mathbf{i}\tilde{\mathbf{s}}_\mathbf{i}' + \tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I}_3\right)^{-1} \\
&= (\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I}_3)^{-1} - (\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I}_3)^{-1}\tilde{\mathbf{s}}_\mathbf{i}[1 + \tilde{\mathbf{s}}_\mathbf{i}'(\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I}_3)^{-1}\tilde{\mathbf{s}}_\mathbf{i}]^{-1}\tilde{\mathbf{s}}_\mathbf{i}'(\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I}_3)^{-1}.
\end{aligned}$$

Let $\mathbf{B} := (\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I_3})^{-1}$, and so

$$(\mathbf{\Sigma}(\mathbf{s_i}) + \mathbf{\Sigma}(\mathbf{s_j}))^{-1} = \mathbf{B} - \mathbf{B}\tilde{\mathbf{s}}_\mathbf{i}(1 + \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i})^{-1}\tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B} = \mathbf{B} - \frac{\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i}\tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}}{1 + \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i}},$$

$$\begin{aligned} q^2(\mathbf{s}_i, s_j) &\propto (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\mathbf{\Sigma}(\mathbf{s_i}) + \mathbf{\Sigma}(\mathbf{s_j}))^{-1}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) \\ &\propto (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'\mathbf{B}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) - \frac{1}{1 + \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i}}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i}\tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}). \end{aligned}$$

So computation of $q(\mathbf{s}_i, s_j)$ only involves terms $\tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i}$, $\tilde{\mathbf{s}}_\mathbf{j}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{j}$ and $\tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{j}$. Because

$$\mathbf{B} = (\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \mathbf{I}_3)^{-1} = \mathbf{I}_3 - \tilde{\mathbf{s}}_\mathbf{j}(1 + \tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j})^{-1}\tilde{\mathbf{s}}_\mathbf{j}' = \mathbf{I}_3 - \frac{1}{2}\tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}',$$

we have

$$\begin{aligned} \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{i} &= \tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{i} - \frac{1}{2}(\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j})^2 = 1 - \frac{1}{2}(\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j})^2 \\ \tilde{\mathbf{s}}_\mathbf{j}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{j} &= \tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j} - \frac{1}{2}(\tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j})^2 = 1 - \frac{1}{2} = \frac{1}{2} \\ \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{B}\tilde{\mathbf{s}}_\mathbf{j} &= \tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j} - \frac{1}{2}(\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j})(\tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j}) = \tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j} - \frac{1}{2}\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j} = \frac{1}{2}\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}. \end{aligned}$$

Further,

$$\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j} = \left[(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) - \tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j}\right]/2 = \left[(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) - 2\right]/2.$$

So $q(\mathbf{s}_i, \mathbf{s}_j)$ just relies on $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$. For the normalization term $c(\mathbf{s}_i, \mathbf{s}_j)$, since we have proved that $|\boldsymbol{\Sigma}(\mathbf{s_i})| = |\boldsymbol{\Sigma}(\mathbf{s_j})| \equiv \gamma^2$,

$$
\begin{aligned}
c(\mathbf{s}_i, \mathbf{s}_j) &= |\boldsymbol{\Sigma}(\mathbf{s_i})|^{1/4} |\boldsymbol{\Sigma}(\mathbf{s_j})|^{1/4} |(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))/2|^{-1/2} \\
&\propto |(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1}|^{1/2} \\
&\propto \left(|(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1}| \cdot |(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})| / [(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})] \right)^{1/2} \\
&\propto \left(|(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1}| \cdot det\{(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'\} / [(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})] \right)^{1/2} \\
&\propto \left(det \left\{ (\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1} (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})' \right\} / [(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})] \right)^{1/2} \\
&\propto \left(det \left\{ (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\boldsymbol{\Sigma}(\mathbf{s_i}) + \boldsymbol{\Sigma}(\mathbf{s_j}))^{-1} (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) \right\} / [(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})] \right)^{1/2} \\
&\propto \left\{ \frac{q^2(\mathbf{s}_i, \mathbf{s}_j)}{(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})} \right\}^{1/2}.
\end{aligned}
\tag{3.8}
$$

We have proved that $q(\mathbf{s}_i, \mathbf{s}_j)$ just depends on $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$, so $c(\mathbf{s}_i, \mathbf{s}_j)$ also relies on the distance $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$ only.

Overall, we can show

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

only depends on the distance $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$, where $\rho(q)$ is a valid isotropic correlation function. So $\rho_{NS}(\mathbf{s}_i, \mathbf{s}_j)$ is isotropic. □

Proof of Theorem 5. If $\kappa(\mathbf{s}) \equiv 0$ and $\gamma_1(\cdot)$, $\gamma_2(\cdot)$ depend on s_2 only, then $\mathcal{R}_x(\kappa(s)) \equiv \mathcal{R}_x(0) = \mathbf{I}_3$.

Then

$$\begin{aligned}\tilde{\boldsymbol{\Sigma}}(\mathbf{s}) =& \mathbf{D}(\gamma(\mathbf{s})) \\ =& \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma_1(s_2) & 0 \\ 0 & 0 & \gamma_2(s_2) \end{pmatrix} \\ =& \begin{pmatrix} 1 & 0 & 0 \\ 0 & \gamma_1(s_2) & 0 \\ 0 & 0 & \gamma_1(s_2) \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \gamma_2(s_2) - \gamma_1(s_2) \end{pmatrix}.\end{aligned}$$

Due to the results in Theorem 3, we have

$$\begin{aligned}\boldsymbol{\Sigma}(\mathbf{s}) =& \mathcal{R}_z(s_1)\mathcal{R}_y(s_2)\tilde{\boldsymbol{\Sigma}}(\mathbf{s})\mathcal{R}_y(s_2)'\mathcal{R}_z(s_1)' \\ =& (1-\gamma_1(s_2))\tilde{\mathbf{s}}\tilde{\mathbf{s}}' + \gamma_1(s_2)\mathbf{I}_3 + (\gamma_2(s_2) - \gamma_1(s_2))\tilde{\mathbf{s}}^*(\tilde{\mathbf{s}}^*)',\end{aligned}$$

where

$$\tilde{\mathbf{s}}^* = \begin{pmatrix} \cos(s_1)\sin(s_2) \\ \sin(s_1)\sin(s_2) \\ \cos(s_2) \end{pmatrix}, \quad (\tilde{\mathbf{s}}^*)'(\tilde{\mathbf{s}}^*) = 1.$$

Thus

$$
\begin{aligned}
|\mathbf{\Sigma}(\mathbf{s})| &= det\{(1-\gamma_1(s_2))\ \tilde{\mathbf{s}}\tilde{\mathbf{s}}' + \gamma_1(s_2)\mathbf{I}_3 + (\gamma_2(s_2)-\gamma_1(s_2))\tilde{\mathbf{s}}^*(\tilde{\mathbf{s}}^*)'\} \\
&= \gamma_1(s_2)^3 \cdot det\left\{ \frac{1-\gamma_1(s_2)}{\gamma_1(s_2)}\tilde{\mathbf{s}}\tilde{\mathbf{s}}' + \frac{\gamma_2(s_2)-\gamma_1(s_2)}{\gamma_1(s_2)}\tilde{\mathbf{s}}^*(\tilde{\mathbf{s}}^*)' + \mathbf{I}_3 \right\} \\
&= \gamma_1(s_2)^3 \cdot det\left\{ \begin{pmatrix} \frac{1-\gamma_1(s_2)}{\gamma_1(s_2)}\tilde{\mathbf{s}} & \frac{\gamma_2(s_2)-\gamma_1(s_2)}{\gamma_1(s_2)}\tilde{\mathbf{s}}^* \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{s}}' \\ (\tilde{\mathbf{s}}^*)' \end{pmatrix} + \mathbf{I}_3 \right\} \\
&= \gamma_1(s_2)^3 \cdot det\left\{ \begin{pmatrix} \tilde{\mathbf{s}}' \\ (\tilde{\mathbf{s}}^*)' \end{pmatrix} \begin{pmatrix} \frac{1-\gamma_1(s_2)}{\gamma_1(s_2)}\tilde{\mathbf{s}} & \frac{\gamma_2(s_2)-\gamma_1(s_2)}{\gamma_1(s_2)}\tilde{\mathbf{s}}^* \end{pmatrix} + \mathbf{I}_2 \right\} \\
&= \gamma_1(s_2)^3 \cdot \begin{vmatrix} \frac{1-\gamma_1(s_2)}{\gamma_1(s_2)} + 1 & 2\sin(s_2)\cos(s_2) \\ 2\sin(s_2)\cos(s_2) & \frac{\gamma_2(s_2)-\gamma_1(s_2)}{\gamma_1(s_2)} + 1 \end{vmatrix} \\
&= \gamma_1(s_2)\gamma_2(s_2) - 4\gamma_1(s_2)^3\sin^2(s_2)\cos^2(s_2)
\end{aligned}
$$

only depend on s_2. WLOG, ignore $\gamma_1(s_2)$, $\gamma_2(s_2)$ again (they only depends on s_2),

$$
\begin{aligned}
&(\mathbf{\Sigma}(\mathbf{s_i}) + \mathbf{\Sigma}(\mathbf{s_j}))^{-1} \\
=& \left[\tilde{\mathbf{s}}_\mathbf{i}\tilde{\mathbf{s}}_\mathbf{i}' + \tilde{\mathbf{s}}_\mathbf{i}^*(\tilde{\mathbf{s}}_\mathbf{i}^*)' + \tilde{\mathbf{s}}_\mathbf{j}\tilde{\mathbf{s}}_\mathbf{j}' + \tilde{\mathbf{s}}_\mathbf{j}^*(\tilde{\mathbf{s}}_\mathbf{j}^*)' + \mathbf{I}_3\right]^{-1} \\
=& \left[\begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}' \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)' \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j} & \tilde{\mathbf{s}}_\mathbf{j}^* \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j}' \\ (\tilde{\mathbf{s}}_\mathbf{j}^*)' \end{pmatrix} + \mathbf{I}_3 \right]^{-1} \\
=& \mathbf{V}^{-1} - \mathbf{V}^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \left[\mathbf{I}_2 + \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}' \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)' \end{pmatrix} \mathbf{V}^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \right]^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}' \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)' \end{pmatrix} \mathbf{V}^{-1} \\
=& \mathbf{V}^{-1} - \mathbf{V}^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \left[\mathbf{I}_2 + \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i}^* \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i} & (\tilde{\mathbf{s}}_\mathbf{i}^*)'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \right]^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}' \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)' \end{pmatrix} \mathbf{V}^{-1},
\end{aligned}
$$

where

$$
\mathbf{V} = \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j} & \tilde{\mathbf{s}}_\mathbf{j}^* \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j}' \\ (\tilde{\mathbf{s}}_\mathbf{j}^*)' \end{pmatrix} + \mathbf{I}_3.
$$

Then

$$
\begin{aligned}
q^2(\mathbf{s}_i, \mathbf{s}_j) \propto & (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\mathbf{\Sigma}(\mathbf{s}_\mathbf{i}) + \mathbf{\Sigma}(\mathbf{s}_\mathbf{j}))^{-1}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) \\
\propto & (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'\mathbf{V}^{-1}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) - (\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'\mathbf{V}^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \cdot \\
& \left[\mathbf{I}_2 + \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i} & \tilde{\mathbf{s}}_\mathbf{i}'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i}^* \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i} & (\tilde{\mathbf{s}}_\mathbf{i}^*)'\mathbf{V}^{-1}\tilde{\mathbf{s}}_\mathbf{i}^* \end{pmatrix} \right]^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{i}' \\ (\tilde{\mathbf{s}}_\mathbf{i}^*)' \end{pmatrix} \mathbf{V}^{-1}(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}).
\end{aligned}
$$

Because

$$
\begin{aligned}
\mathbf{V}^{-1} &= \mathbf{I}_3 - \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j} & \tilde{\mathbf{s}}_\mathbf{j}^* \end{pmatrix} \left[\mathbf{I}_2 + \begin{pmatrix} 1 & \tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j}^* \\ (\tilde{\mathbf{s}}_\mathbf{j})'\tilde{\mathbf{s}}_\mathbf{j} & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j}' \\ (\tilde{\mathbf{s}}_\mathbf{j}^*)' \end{pmatrix} \\
&= \mathbf{I}_3 - \frac{1}{4 - (\tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j}^*)^2} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j} & \tilde{\mathbf{s}}_\mathbf{j}^* \end{pmatrix} \left[\begin{pmatrix} 2 & -\tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{j}^* \\ -(\tilde{\mathbf{s}}_\mathbf{j})'\tilde{\mathbf{s}}_\mathbf{j} & 2 \end{pmatrix} \right]^{-1} \begin{pmatrix} \tilde{\mathbf{s}}_\mathbf{j}' \\ (\tilde{\mathbf{s}}_\mathbf{j}^*)' \end{pmatrix},
\end{aligned}
$$

we can figure out that the computation of $q^2(\mathbf{s}_i, \mathbf{s}_j)$ only involves the following types of terms

$$
\begin{cases}
\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{i} = 1 \\
\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{i}^* = \tilde{\mathbf{s}}_\mathbf{i}' \cdot \left[(\tan(s_{i2}))\tilde{\mathbf{s}}_\mathbf{i} + \left(0, 0, \cos(s_{i2}) - \dfrac{\sin^2(s_{i2})}{\cos(s_{i2})}\right)' \right] \\
\qquad = \tan(s_{i2}) + \sin(s_{i2}) \left[\cos(s_{i2}) - \dfrac{\sin^2(s_{i2})}{\cos(s_{i2})} \right] \\
(\tilde{\mathbf{s}}_\mathbf{i}^*)'\tilde{\mathbf{s}}_\mathbf{i}^* = 1 \\
(\tilde{\mathbf{s}}_\mathbf{i}^*)'\tilde{\mathbf{s}}_\mathbf{j}^* = \left[(\tan(s_{i2}))\tilde{\mathbf{s}}_\mathbf{i} + \left(0, 0, \cos(s_{i2}) - \dfrac{\sin^2(s_{i2})}{\cos(s_{i2})}\right)' \right]' \cdot \\
\qquad \left[(\tan(s_{j2}))\tilde{\mathbf{s}}_\mathbf{j} + \left(0, 0, \cos(s_{j2}) - \dfrac{\sin^2(s_{j2})}{\cos(s_{j2})}\right)' \right] \\
\qquad = \tan(s_{i2})\tan(s_{j2})(\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}) + \tan(s_{i2})\sin(s_{i2}) \left[\cos(s_{j2}) - \dfrac{\sin^2(s_{j2})}{\cos(s_{j2})} \right] \\
\qquad\quad + \tan(s_{j2})\sin(s_{j2}) \left[\cos(s_{i2}) - \dfrac{\sin^2(s_{i2})}{\cos(s_{i2})} \right] \\
\qquad\quad + \left[\cos(s_{i2}) - \dfrac{\sin^2(s_{i2})}{\cos(s_{i2})} \right] \left[\cos(s_{j2}) - \dfrac{\sin^2(s_{j2})}{\cos(s_{j2})} \right] \\
\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}^* = \tan(s_{j2})(\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}) + \sin(s_{i2}) \left[\cos(s_{j2}) - \dfrac{\sin^2(s_{j2})}{\cos(s_{j2})} \right] \\
\tilde{\mathbf{s}}_\mathbf{j}'\tilde{\mathbf{s}}_\mathbf{i}^* = \tan(s_{i2})(\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}) + \sin(s_{j2}) \left[\cos(s_{i2}) - \dfrac{\sin^2(s_{i2})}{\cos(s_{i2})} \right] \\
\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j} = [(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}) - 2]/2.
\end{cases}
$$

We can change the index i to j for the first 3 terms and they are still valid. Thus these values only depend on s_{i2}, s_{j2} and $\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}$, and $\tilde{\mathbf{s}}_\mathbf{i}'\tilde{\mathbf{s}}_\mathbf{j}$ can be written in terms of $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$. The computation of $q^2(\mathbf{s}_i, \mathbf{s}_j)$ only relies on the distance $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$ and the longitudes s_{i2}, s_{j2}. Similar to (3.8) in the proof of Theorem 3, we can also show that $c(\mathbf{s}_i, \mathbf{s}_j)$ is a function of $(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})'(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j})$, s_{i2} and s_{j2}. Then $\rho_{NS}(\mathbf{s}_i, \mathbf{s}_j) = c(\mathbf{s}_i, \mathbf{s}_j)\rho(q(\mathbf{s}_i, \mathbf{s}_j)) := \rho_A(\tilde{\mathbf{s}}_\mathbf{i} - \tilde{\mathbf{s}}_\mathbf{j}, s_{i2}, s_{j2})$, so it is axially symmetric. □

3.6 Numerical study

We performed simulations to demonstrate how Gaussian processes with various covariance functions actually behave on the sphere. We started with the regular Matérn covariance function (smoothness 0.5, range 0.668, and a nugget effect 0.05^2), and then constructed covariance functions based on (3.6) on the sphere. The scaling parameters $\gamma_1(\mathbf{s})$ and $\gamma_2(\mathbf{s})$ were given by

$$\gamma_1(\mathbf{s}) = exp(\beta_{10} + \beta_{11}\sin(s_1) + \beta_{12}s_2), \tag{3.9}$$

$$\gamma_2(\mathbf{s}) = exp(\beta_{20} + \beta_{21}\sin(s_1) + \beta_{22}s_2), \tag{3.10}$$

where $\mathbf{s} = (s_1, s_2)$ with longitude s_1 and latitude s_2.

We considered three different "true" covariance functions:

- **Isotropy:** According to Theorem 3, the correlation function ρ_{NS} in (3.5) is isotropic if $\gamma_1(\mathbf{s}) = \gamma_2(\mathbf{s}) \equiv \gamma$ is constant, and so we set the parameters in (3.9)–(3.10) as

$$\beta_1 = (\beta_{10}, \beta_{11}, \beta_{12}) = (-0.5, 0, 0),$$

$$\beta_2 = (\beta_{20}, \beta_{21}, \beta_{22}) = (-0.5, 0, 0).$$

- **Axial symmetry:** According to Theorem 5, the correlation function ρ_{NS} in (3.5) is axially symmetric if $\kappa(\mathbf{s}) \equiv 0$ and $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ are functions of latitude only, and so we set the parameters as

$$\beta_1 = (\beta_{10}, \beta_{11}, \beta_{12}) = (-0.5, 0, 1.44),$$

$$\beta_2 = (\beta_{20}, \beta_{21}, \beta_{22}) = (-3.2, 0, 1.44).$$

- **General nonstationarity:** A more general nonstationary covariance function can be obtained by setting

$$\kappa = 0.8,$$

$$\beta_1 = (\beta_{10}, \beta_{11}, \beta_{12}) = (-0.5, -1.2, 1.44),$$

$$\beta_2 = (\beta_{20}, \beta_{21}, \beta_{22}) = (-3.2, -0.3, 1.44).$$

We simulated data from a GP with each of the three true covariance functions. Each of the datasets was then analyzed under each of the three assumptions:

• **Isotropy:** unknown β_{10}, β_{20}, with fixed $\beta_{11} = \beta_{12} = \beta_{21} = \beta_{22} = \kappa = 0$.

• **Axial symmetry:** unknown $\beta_{10}, \beta_{12}, \beta_{20}, \beta_{22}$, with fixed $\beta_{11} = \beta_{21} = \kappa = 0$.

• **General nonstationarity:** unknown $\beta_{10}, \beta_{11}, \beta_{12}, \beta_{20}, \beta_{21}, \beta_{22}, \kappa$.

In each setting, we assumed independent normal priors for the unknown parameters with mean 0 and standard deviation 6 (for the rotation parameter κ, we assumed a uniform prior). The variances and smoothness were known. We used an adaptive MCMC algorithm (Vihola, 2012) for Bayesian inference based on the Vecchia approximation in (3.7) of the likelihood, using maximum-minimum-distance ordering, nearest-neighbor conditioning, and conditioning-set size 10. We also used the Vecchia approximation to calculate the posterior predictive distribution at test locations, with the unknown parameters integrated out based on the MCMC samples.

For each setting, we simulated GP realizations at $2{,}500$ gridded locations on the sphere, which were split into a training and a test set, under two different scenarios: (a) simple random sampling of 30% of realizations as test data; (b) designating all realizations within a randomly selected test region as test data. The test region was generated by randomly sampling a location on the sphere as the center of a rectangle, assuming the rectangle has length 2.5 (across longitude in radians) and width 1 (across latitude in radians). The resulting test region contains 340 grid points. To measure the prediction performance, we compared the mean absolute error (MAE), root mean squared error (RMSE), the continuous ranked probability score (CRPS) (e.g., Gneiting and Katzfuss, 2014), and the energy score among the different models. MAE and RMSE consider only point predictions, CRPS evaluates marginal predictive distributions, and the energy score evaluates the joint predictive distribution for the entire test set. Examples of predictions in the case of isotropic, axially

(a) True model - Isotropic

	Random				Region			
	MAE	RMSE	CRPS	Energy	MAE	RMSE	CRPS	Energy
Isotropic	0.2569	0.3303	0.1842	6.4160	0.5727	0.6921	0.4034	8.8546
Axially symmetric	0.2566	0.3302	0.1835	6.4427	0.5816	0.7018	0.4079	9.1464
Nonstationary	0.2568	0.3302	0.1842	6.4231	0.5726	0.6917	0.3987	8.9940

(b) True model - Axially symmetric

	Random				Region			
	MAE	RMSE	CRPS	Energy	MAE	RMSE	CRPS	Energy
Isotropic	0.3407	0.4507	0.2455	8.7584	0.7317	0.8726	0.4952	10.9912
Axially symmetric	0.3390	0.4473	0.2401	8.6879	0.6869	0.8223	0.4863	10.9177
Nonstationary	0.3387	0.4470	0.2415	8.7104	0.6976	0.8319	0.4831	10.7633

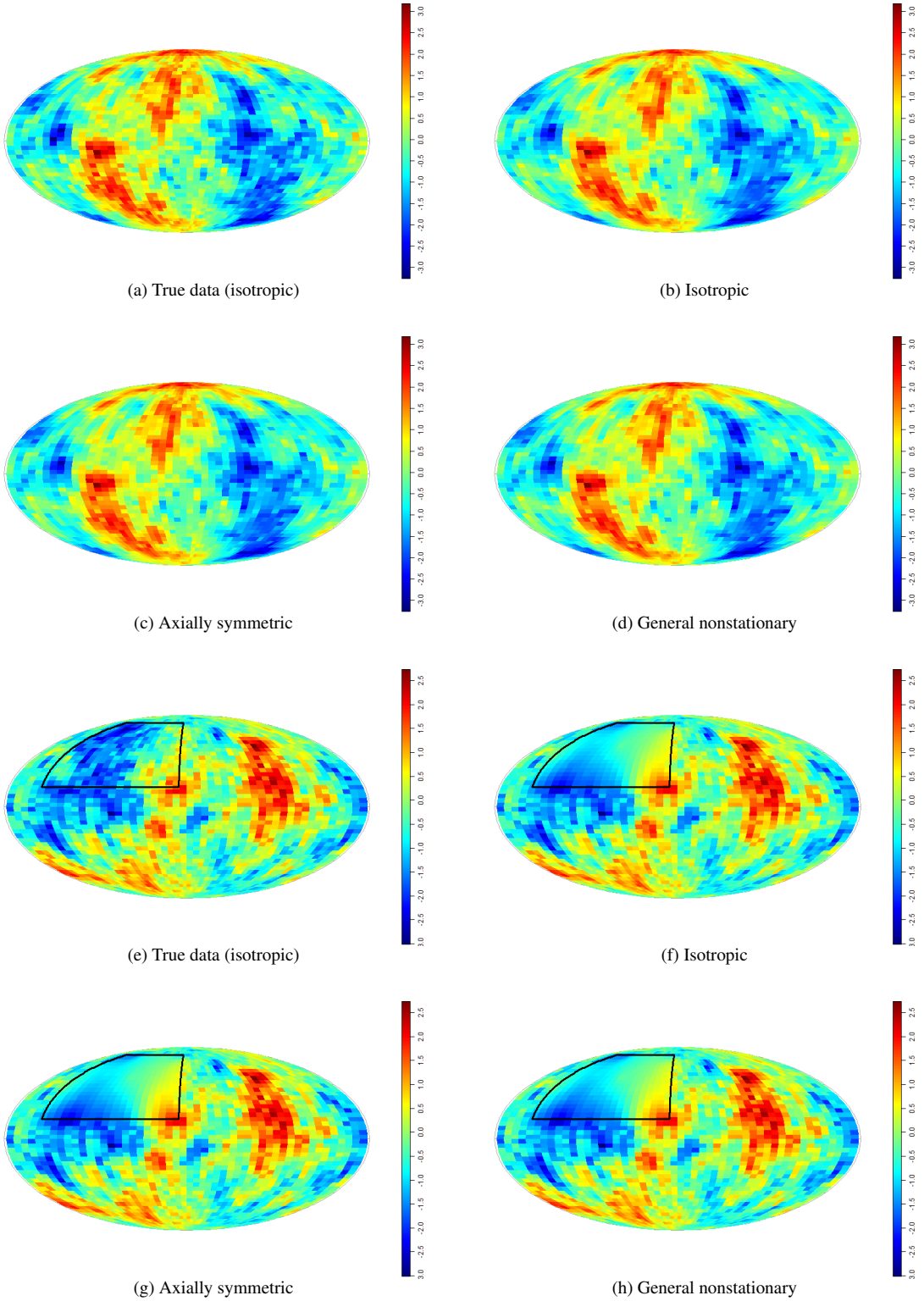
(c) True model - Nonstationary

	Random				Region			
	MAE	RMSE	CRPS	Energy	MAE	RMSE	CRPS	Energy
Isotropic	0.3975	0.5259	0.2863	10.1948	0.8655	1.0174	0.5928	13.1071
Axially symmetric	0.3940	0.5211	0.2817	10.1614	0.7593	0.9018	0.5266	11.6615
Nonstationary	0.3719	0.4890	0.2669	9.4416	0.6581	0.7958	0.4586	10.2956

Table 3.1: Prediction scores, each averaged over 10 simulated datasets, for three different true models and three different assumed models. Test sets were selected via simple random sampling (Random) or based on a randomly selected test region (Region).

symmetric, and general nonstationary data are demonstrated in Figure 3.5, Figure 3.6, and Figure 3.7, respectively.

The prediction scores are summarized in Table 3.1. For isotropic dataset, the scores were fairly similar under the assumptions of isotropic, axially symmetric, and nonstationary structure. The difference became more pronounced when the true data were from the axially symmetric model. Furthermore, in the modeling of more general nonstationary data, using the correct nonstationary structure considerably outperformed the other two covariance structures. Typically, the differences were more pronounced for test regions then for random test sets, as expected.



(a) True data (isotropic)

(b) Isotropic

(c) Axially symmetric

(d) General nonstationary

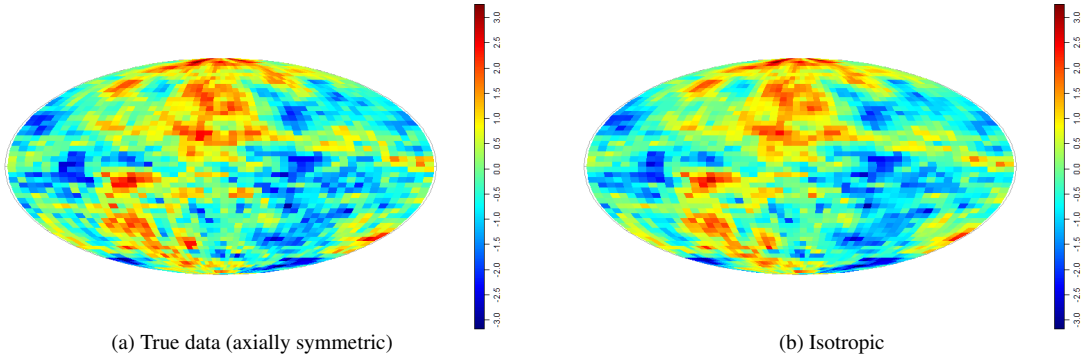
(e) True data (isotropic)

(f) Isotropic

(g) Axially symmetric

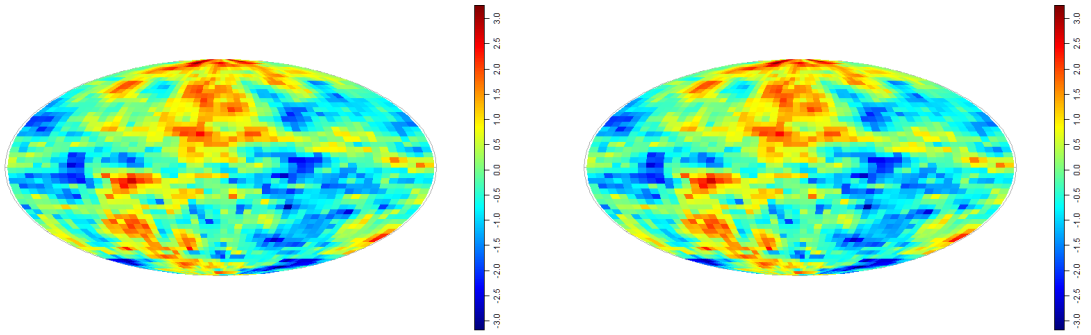
(h) General nonstationary

Figure 3.5: For data simulated using an **isotropic** covariance function, illustration of predictions under three different assumptions (isotropic, axially symmetric, and general nonstationary) based on randomly selected test data (a, b, c, d) and for a test region (black lines in (e, f, g, h))



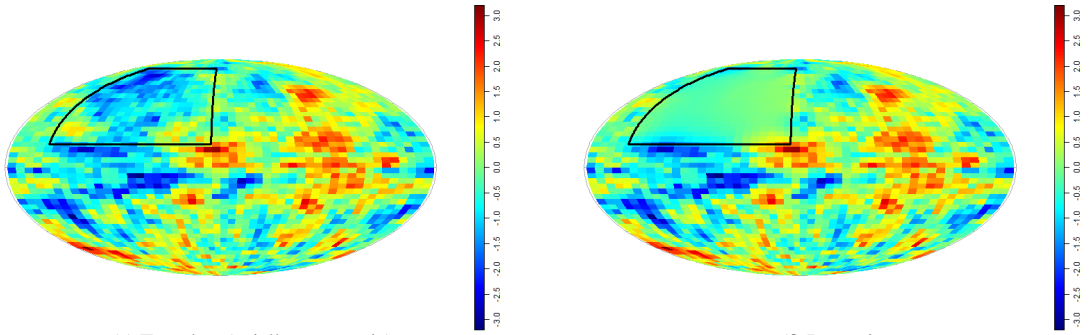
(a) True data (axially symmetric)

(b) Isotropic



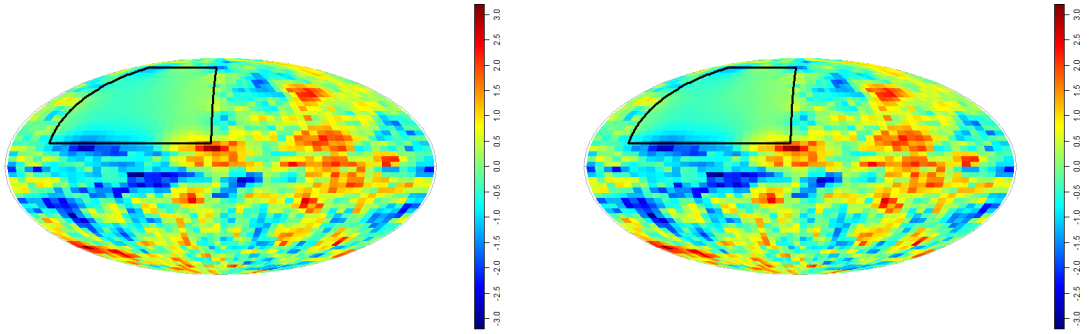
(c) Axially symmetric

(d) General nonstationary



(e) True data (axially symmetric)

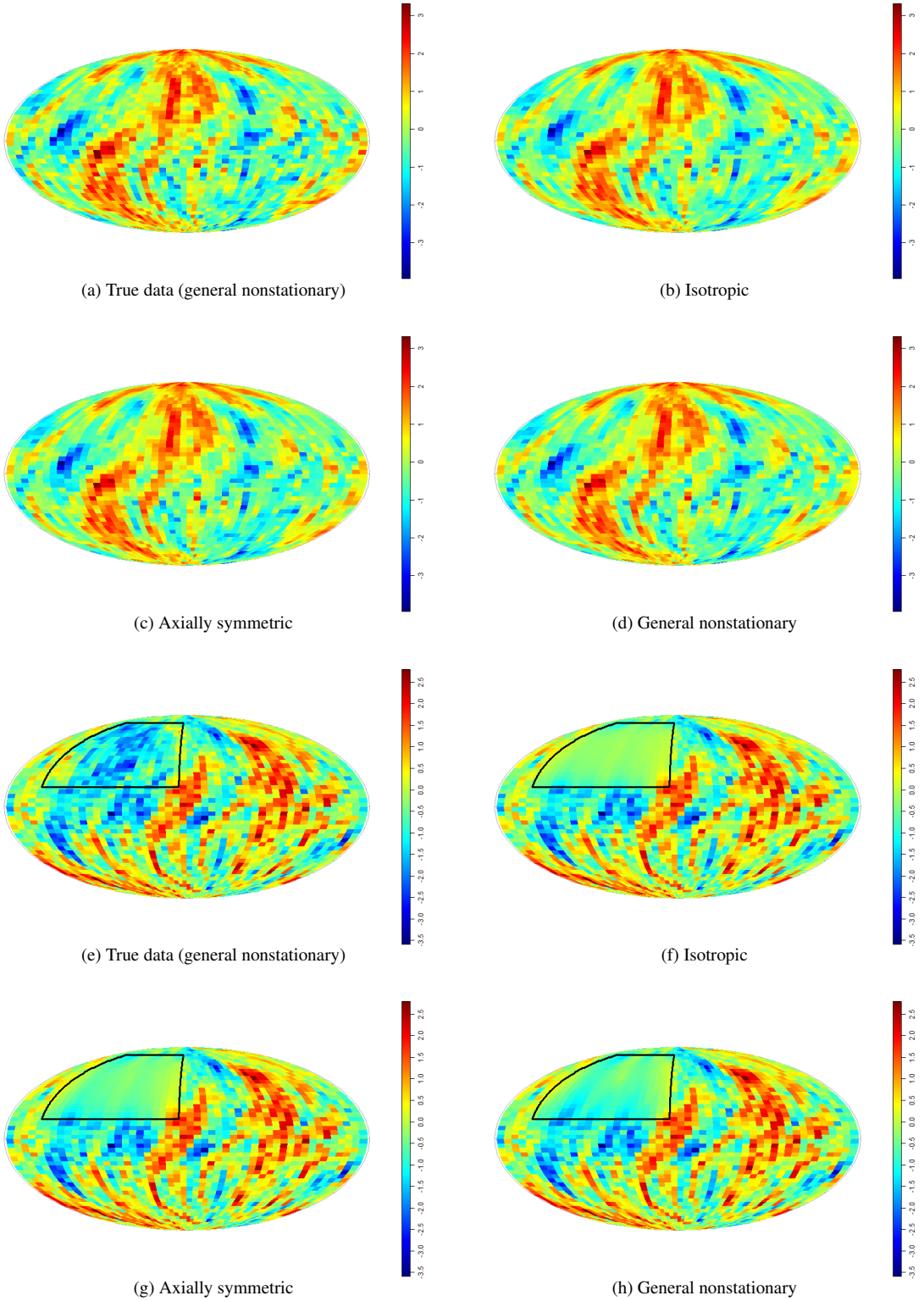
(f) Isotropic



(g) Axially symmetric

(h) General nonstationary

Figure 3.6: For data simulated using an **axially symmetric** covariance function, illustration of predictions under three different assumptions (isotropic, axially symmetric, and general nonstationary) based on randomly selected test data (a, b, c, d) and for a test region (black lines in (e, f, g, h))



(a) True data (general nonstationary)

(b) Isotropic

(c) Axially symmetric

(d) General nonstationary

(e) True data (general nonstationary)

(f) Isotropic

(g) Axially symmetric

(h) General nonstationary

Figure 3.7: For data simulated using a **general nonstationary** covariance function, illustration of predictions under three different assumptions (isotropic, axially symmetric, and general nonstationary) based on randomly selected test data (a, b, c, d) and for a test region (black lines in (e, f, g, h))

3.7 Discussion

We proposed a general approach for constructing nonstationary, locally anisotropic covariance functions on the sphere based on covariance functions for Euclidean space. We described how to adapt a valid nonstationary covariance function in $\mathbb{R}^3$ to the sphere $\mathbb{S}$, and explored properties of the parameterization. By adding certain constraints to the scaling and rotation parameters, the resulting covariance function is isotropic or axially symmetric, which are widely used assumptions in geospatial analysis of global data. For large datasets on the sphere, direct application of Gaussian process is too computationally expensive, and so we applied the Vecchia approximation to achieve fast inference. We illustrated realizations obtained from Gaussian processes with various covariance functions constructed by our method, and provided numerical comparisons.

4. CONCLUSIONS

In this dissertation, we presented fast inference methods for multi-scale and global spatial processes. The proposed multi-scale Vecchia (MSV) approximation of Gaussian processes exhibited highly competitive performance in the temperature-data application relative to a large set of existing approaches for analyzing large spatial data. The visualizations of the different scales from our MSV method can be useful in many scientific contexts. For global spatial analysis, we developed a general approach for constructing nonstationary, locally anisotropic covariance functions on the sphere based on covariance functions for Euclidean space. By adding certain constraints to the scaling and rotation parameters, the resulting covariance functions are isotropic or axially symmetric, which are widely used assumptions in geospatial analysis of global data. Combining this approach with the Vecchia approximation, our method can achieve fast inference for large global spatial data.

REFERENCES

Ba, S. and Joseph, V. R. (2012). Composite Gaussian process models for emulating expensive functions. *Annals of Applied Statistics*, 6(4):1838–1860.

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.

Castruccio, S. and Genton, M. G. (2014). Beyond axial symmetry: An improved class of models for global data. *Stat*, 3(1):48–55.

Castruccio, S. and Genton, M. G. (2016). Compressing an ensemble with statistical models: An algorithm for global 3d spatio-temporal temperature. *Technometrics*, 58(3):319–328.

Castruccio, S., Stein, M. L., et al. (2013). Global space–time models for climate ensembles. *The Annals of Applied Statistics*, 7(3):1593–1611.

Comer, M. L. and Delp, E. J. (1999). Segmentation of textured images using a multiresolution gaussian autoregressive model. *IEEE Transactions on Image Processing*, 8(3):408–420.

Cotton, W. R., Bryan, G., and Van den Heever, S. C. (2010). *Storm and Cloud Dynamics*, volume 99. Academic press.

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.

Das, B. (2000). *Global Covariance Modeling: A Deformation Approach to Anisotropy*. Ph.D. thesis, University of Washington.

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.

Du, J., Ma, C., and Li, Y. (2013). Isotropic variogram matrix functions on spheres. *Mathematical Geosciences*, 45(3):341–357.

Du, J., Zhang, H., and Mandrekar, V. S. (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *The Annals of Statistics*, 37:3330–3361.

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. *Proceedings of the 30th International Conference on Machine Learning*, 28:1166–1174.

Ferreira, M. A. and Lee, H. K. (2007). *Multiscale Modeling: A Bayesian Perspective*. Springer.

Ferreira, M. A., West, M., Lee, H. K., Higdon, D. M., et al. (2006). Multi-scale and hidden resolution time series models. *Bayesian Analysis*, 1(4):947–967.

Furrer, R., Genton, M. G., and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523.

Gneiting, T. (2013). Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4):1327–1349.

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

Gotway, C. A. and Young, L. J. (2002). Combining incompatible spatial data. *Journal of the American Statistical Association*, 97(458):632–648.

Guinness, J. (2016). Permutation methods for sharpening gaussian process approximations. *arXiv:1609.05372*.

Guinness, J. (2018). Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics*, 60(4):415–429.

Guinness, J. and Fuentes, M. (2016). Isotropic covariance functions on spheres: Some properties and modeling considerations. *Journal of Multivariate Analysis*, 143:143–152.

Heaton, M., Katzfuss, M., Berrett, C., and Nychka, D. (2014). Constructing valid spatial processes on the sphere using kernel convolutions. *Environmetrics*, 25(1):2–15.

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2019). A case study competition among methods for analyzing large spatial data.

Journal of Agricultural, Biological, and Environmental Statistics, 24(3):398–425.

Hitczenko, M. and Stein, M. L. (2012). Some theory for anisotropic processes on the sphere. *Statistical Methodology*, 9(1-2):211–227.

Huang, C., Zhang, H., and Robeson, S. M. (2011). On the validity of commonly used covariance and variogram functions on the sphere. *Mathematical Geosciences*, 43(6):721–733.

Huang, H.-C., Cressie, N., and Gabrosek, J. (2002). Fast, resolution-consistent spatial prediction of global processes from satellite data. *Journal of Computational and Graphical Statistics*, 11(1):63–88.

Jeong, J., Jun, M., and Genton, M. G. (2017). Spherical process models for global spatial statistics. *Statistical Science: a review journal of the Institute of Mathematical Statistics*, 32(4):501.

Jones, R. (1963). Stochastic processes on a sphere. *Annals of Mathematical Statistics*, 34(1):213–218.

Jun, M. (2014). Matérn-based nonstationary cross-covariance models for global processes. *Journal of Multivariate Analysis*, 128:134–146.

Jun, M. and Stein, M. L. (2007). An approach to producing space–time covariance functions on spheres. *Technometrics*, 49(4):468–479.

Jun, M. and Stein, M. L. (2008). Nonstationary covariance models for global data. *Annals of Applied Statistics*, 2(4):1271–1289.

Katzfuss, M. (2011). *Hierarchical Spatial and Spatio-Temporal Modeling of Massive Datasets, with Application to Global Mapping of CO2*. Ph.D. thesis, The Ohio State University.

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. (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 Gong, W. (2020). A class of multi-resolution approximations for large spatial datasets. *Statistica Sinica*, 30(4):2203–2226.

Katzfuss, M. and Guinness, J. (2017). A general framework for vecchia approximations of gaussian

processes. *arXiv:1708.06302*.

Katzfuss, M., Guinness, J., Gong, W., and Zilber, D. (2020a). Vecchia approximations of Gaussian-process predictions. *Journal of Agricultural, Biological, and Environmental Statistics*, 25(3):383–414.

Katzfuss, M., Guinness, J., and Lawrence, E. (2020b). Scaled Vecchia approximation for fast computer-model emulation. *arXiv:2005.00386*.

Katzfuss, M., Jurek, M., Zilber, D., Gong, W., Guinness, J., Zhang, J., and Schäfer, F. (2020c). *GPvecchia: Fast Gaussian-process inference using Vecchia approximations*. R package version 0.1.3.

Kim, S.-W., Yoon, S.-C., Kim, J., and Kim, S.-Y. (2007). Seasonal and monthly variations of columnar aerosol optical properties over east asia determined from multi-year modis, lidar, and aeronet sun/sky radiometer measurements. *Atmospheric Environment*, 41(8):1634–1651.

Knapp, A. (2012). *Global Bayesian Nonstationary Spatial Modeling for Very Large Datasets*. Bachelor thesis, University of Heidelberg.

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 (Statistical Methodology)*, 73(4):423–498.

Ma, C. (2012). Stationary and isotropic vector random fields on spheres. *Mathematical Geosciences*, 44(6):765–778.

Ma, C. (2015). Isotropic covariance matrix functions on all spheres. *Mathematical Geosciences*, 47(6):699–717.

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. E. (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., 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.

Saquib, S. S., Bouman, C. A., and Sauer, K. (1996). A non-homogeneous mrf model for multiresolution bayesian estimation. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 2, pages 445–448. IEEE.

Schäfer, F., Katzfuss, M., and Owhadi, H. (2020). Sparse Cholesky factorization by Kullback-Leibler minimization. *arXiv:2004.14455*.

Schäfer, F., Sullivan, T. J., and Owhadi, H. (2017). Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *arXiv:1706.02205*.

Skøien, J. O., Blöschl, G., and Western, A. (2003). Characteristic space scales and timescales in hydrology. *Water Resources Research*, 39(10).

Snelson, E. and Ghahramani, Z. (2007). Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics*, pages 524–531.

Sobolewska, M. A., Siemiginowska, A., Kelly, B. C., and Nalewajko, K. (2014). Stochastic modeling of the Fermi/LAT γ-ray blazar variability. *Astrophysical Journal*, 786(143).

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York, NY.

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

Stein, M. L. et al. (2007). Spatial variation of total column ozone on a global scale. *The Annals of Applied Statistics*, 1(1):191–210.

Tzeng, S., Huang, H.-C., and Cressie, N. (2005). A fast, optimal spatial-prediction method for massive datasets. *Journal of the American Statistical Association*, 100(472):1343–1357.

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

Vihola, M. (2012). Robust adaptive metropolis algorithm with coerced acceptance rate. *Statistics*

and Computing, 22(5):997–1008.

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

Wilson, A. G. and Adams, R. P. (2013). Gaussian process kernels for pattern discovery and extrapolation. *Proceedings of the 30th International Conference on Machine Learning*, 28(3):1067–1075.

Wilson, A. G., Gilboa, E., Nehorai, A., and Cunningham, J. P. (2014). Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, pages 3626–3634.

Yaglom, A. (1987). *Correlation Theory of Stationary and Related Random Functions, Vol. 1*. Springer, New York, NY.

Zhu, J., Morgan, C. L., Norman, J. M., Yue, W., and Lowery, B. (2004). Combined mapping of soil properties using a multi-scale tree-structured spatial model. *Geoderma*, 118(3-4):321–334.