2020

# SEMIVARIOGRAM METHODS FOR MODELING WHITTLE-MATERN PRIORS IN BAYESIAN INVERSE PROBLEMS

Richard David Brown

# SEMIVARIOGRAM METHODS FOR MODELING WHITTLE-MATÉRN PRIORS IN BAYESIAN INVERSE PROBLEMS

By

Richard David Brown

Bachelor of Science, California State University, Chico, Chico, CA, 2013
Master of Science, Colorado State University, Fort Collins, CO, 2015

Dissertation

presented in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in Mathematics

The University of Montana
Missoula, MT

May 2020

Approved by:

Ashby Kinch, Associate Dean of Graduate School
Graduate School

Dr. Johnathan Bardsley, Chair
Mathematical Sciences

Dr. Jonathan Graham
Mathematical Sciences

Dr. Javier Pérez Álvaro
Mathematical Sciences

Dr. Emily Stone
Mathematical Sciences

Dr. Jesse Johnson
Computer Science

Brown, Richard, Ph.D., Spring 2020 Mathematical Sciences

Semivariogram Methods for Modeling Whittle-Matérn Priors in Bayesian Inverse Problems

Chairperson: Dr. Johnathan Bardsley

## Abstract

We present a new technique, based on semivariogram methodology, for obtaining point estimates for use in prior modeling for solving Bayesian inverse problems. This method requires a connection between Gaussian processes with covariance operators defined by the Matérn covariance function and Gaussian processes with precision (inverse-covariance) operators defined by the Green's functions of a class of elliptic stochastic partial differential equations (SPDEs). We present a detailed mathematical description of this connection. We will show that there is an equivalence between these two Gaussian processes when the domain is infinite – for us, $\mathbb{R}^2$ – which breaks down when the domain is finite due to the effect of boundary conditions on Green's functions of PDEs. We show how this connection can be re-established using extended domains. We then introduce the semivariogram method for estimating the Matérn covariance parameters, which specify the Gaussian prior needed for stabilizing the inverse problem. Results are extended from the isotropic case to the anisotropic case where the correlation length in one direction is larger than another. The situation where the correlation length is spatially dependent rather than constant will also be considered. Finally, we compare and contrast the semivariogram method with a fully-Bayesian approach of finding estimates for and quantifying uncertainty in the hyperparameters. We implement each method in two-dimensional image inpainting test cases to show that it works on practical examples. The MATLAB code for all of these methods can be found here: https://github.com/rbrown53/DissertationCodes.

# Acknowledgments

Rather than only acknowledging the people who assisted with the dissertation, I would like this section to recognize (most of) the people who made my entire graduate school experience at the University of Montana such a positive one.

- First and foremost, thank you to Dr. John Bardsley for your continued guidance, support, and availability as my adviser throughout this process. Your ability to keep me on track while still encouraging me to explore different research avenues was so appreciated and was the primary reason I was able to finish on schedule. Thank you so much for choosing me to be your Ph.D. student when I was still unsure of my research plans. This project turned out to be ideal for me in many ways.

- Thank you to all of my defense committee members, Dr. Jon Graham, Dr. Emily Stone, Dr. Javier Pérez Álvaro, and Dr. Jesse Johnson for your willingness to serve on the committee and for the feedback you provided on the dissertation.

- Thank you to Dr. Jon Graham for introducing me to spatial statistics in general and semivariogram methods in particular and for the role you played in writing and assessing my preliminary and comprehensive exams. *Spatial Statistics* was the only statistics course I took at UMT, but it was one of the most important I have ever taken.

- Thank you to Dr. Fred Peck for the role you played in preparing me to be a better instructor and for serving as a teaching mentor for me the last four years. *Teaching College Mathematics* was one of the most influential classes I took in graduate school and I will be a better educator because of you.

- Thank you to Dr. Emily Stone for serving as department chair throughout my tenure at the University of Montana and for continually taking into account my teaching preferences when assigning classes to TAs.

- Thank you to Dr. Kelly McKinnie and Dr. Cory Palmer for serving as the associate chairs of the graduate program during my time at UMT. You were both very helpful in making sure I was on track to graduate.

- Thank you to Lauren Fern, Cindy Leary, Dr. Kelly McKinnie and Dr. Karel Stroethoff for serving as course coordinators for courses I taught during my time at UMT. It was always a pleasure to work, teach and meet regularly with each of you.

- Thank you to Linda Azure, Barbara Ensor, Michelle Selander, Indy Singh, Zsuzsa Weinhandl, and Maria Yost for serving the Mathematical Sciences Department in each of your positions throughout the years. The department could not have functioned without each of you.

- Thank you to my fellow graduate students. You made my time here immensely enjoyable with various activities that included fun conversations, homework sessions, lively debates, video game nights, table tennis matches, shared meals, and Bible studies.

- Thank you to all mathematics and statistics faculty for being so approachable and warm to every graduate student while always being willing to share your time and expertise.

- Thank you to the Bryan, Myers, and Hartse families for donating to allow for financial support for summer research and various other scholarships and awards. Having this funding available makes it possible for graduate students to complete their degrees on time.

- Finally, thank you so very much to my parents, David and Michelle. You have been my biggest supporters throughout my entire life and have valued my education while pushing me to be a better student and person from the beginning. I know I would not be where I am today without you.

# Contents

List of Figures viii

List of Tables xv

1 Introduction 1

1.1 The Matrix **A** and Boundary Conditions . . . . . . . . . . . . . 3

1.2 Bayesian Inverse Problems . . . . . . . . . . . . . . . . . . . . . 8

1.3 The Matérn Class of Covariance Matrices and Whittle-Matérn Priors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2 Theoretical Results 17

2.1 Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . 18

2.1.1 Gaussian Fields . . . . . . . . . . . . . . . . . . . . . . 18

2.1.2 White Noise . . . . . . . . . . . . . . . . . . . . . . . . 19

2.1.3 Green's Functions . . . . . . . . . . . . . . . . . . . . . 20

2.1.4 Fourier Transform . . . . . . . . . . . . . . . . . . . . . 21

2.2 Isotropic Matérn Connection . . . . . . . . . . . . . . . . . . . 24

2.3 Anisotropic Case . . . . . . . . . . . . . . . . . . . . . . . . . 29

2.3.1 Anisotropic SPDE . . . . . . . . . . . . . . . . . . . . . 29

2.3.2 Anisotropic Matérn Connection . . . . . . . . . . . . . . 33

**3 Numerical Methods** **37**

3.1 Preliminaries . . . 37

3.1.1 Kronecker Product . . . 38

3.1.2 vec(·) and diag(·) Functions . . . 38

3.1.3 Discrete Fourier Transform . . . 39

3.1.4 Conjugate Gradient . . . 42

3.2 Discretizing the SPDE . . . 49

3.2.1 Prior Precision Matrices . . . 55

3.3 The Effect of a Finite Domain and Boundary Conditions . . . 56

3.4 Computing MAP Estimators for Whittle-Matérn Priors . . . 60

**4 Semivariogram Methods** **67**

4.1 Isotropic Case . . . 68

4.1.1 Algorithm . . . 71

4.1.2 Numerical Experiments . . . 72

4.1.3 Discussion . . . 76

4.2 Anisotropic Case . . . 76

4.2.1 Algorithm . . . 80

4.2.2 Numerical Experiments . . . 81

4.2.3 Discussion . . . 84

**5 Extensions** **85**

5.1 Regional Anisotropy . . . 85

5.1.1 Preliminary – Cholesky Factorization . . . 85

5.1.2 Regional Precision Matrix . . . 87

5.1.3 Implementation . . . 90

5.1.4 Preconditioner . . . 91

5.1.5 Region Selection . . . . . . . . . . . . . . . . . . . . . . 92

5.1.6 Prior Weights . . . . . . . . . . . . . . . . . . . . . . . . 94

5.1.7 Numerical Experiments . . . . . . . . . . . . . . . . . . . 95

5.1.8 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . 97

5.2 Markov Chain Monte Carlo Methods . . . . . . . . . . . . . . . 98

5.2.1 MCMC Diagnostics . . . . . . . . . . . . . . . . . . . . . 102

5.2.2 MCMC Results . . . . . . . . . . . . . . . . . . . . . . . 106

5.2.3 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . 109

**6 Conclusion** **112**

6.1 MATLAB Code . . . . . . . . . . . . . . . . . . . . . . . . . . 113

**Bibliography** **115**

# List of Figures

1.1 Least squares solution. On the left is a blurry image of Main Hall. On the right is the least-squares solution for the deblurring inverse problem. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2 Boundary artifacts. When boundary conditions are inappropriate, the reconstruction may contain unrealistic artifacts near the borders. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

1.3 Behavior of the Matérn covariance function. The smoothness parameter, $\nu$, primarily affects the covariance at small distances whereas the range parameter, $\ell$, mainly affects the decay rate of the covariance. The horizontal line corresponds to a covariance value of 0.05 and the practical range is the distance at which the covariance intersects this line. . . . . . . . . . . . . . . . . . 12

1.4 Convergence to the Gaussian covariance. On the left, we show that the norm difference between the Matérn and the Gaussian covariance function converges to zero as $\nu$ increases, plotted on the log scale. On the right, we plot the Exponential and Gaussian covariance functions along with the Matérn covariance for various values of $\nu$ with a fixed practical range of 1 to further illustrate that the Matérn covariance is equal to the Exponential covariance when $\nu = 1/2$ and converges to the Gaussian covariance as $\nu \to \infty$. . . . . . . . . . . . . . . . . . . . . . . . . . 14

2.1 Isotropic field. The correlation between any two points depends only on the distance between those points and is independent of direction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2.2 Anisotropic field. The correlation length differs depending on direction. The correlation length is largest in the 45° direction and shortest in the −45° direction. . . . . . . . . . . . . . . . . 30

3.1 Conjugate gradient methods. The image on the left tracks the convergence of the CG algorithm and the image on the right shows PCG. Both methods converged in two iterations. . . . . 49

3.2 Isotropic correlation maps. Plots of the Matérn correlation map (left), the empirical correlation map with $n = 50$ computed on the domain $\Omega = [0,1] \times [0,1]$ (middle), and the empirical correlation map computed on the domain $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$ (right), computed from random draws from the prior (3.17) in 2D with $\nu = 1$ and $\ell = 1/4$. . . . . . . . . . . . . . . . 57

3.3 Anisotropic correlation maps. Plots of the anisotropic Matérn correlation map (left), the empirical correlation map with $n = 50$ computed on the domain $\Omega = [0,1] \times [0,1]$ (middle), and the empirical correlation map computed on the domain $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$ (right), computed from random draws from the prior (3.15) in 2D with $\nu = 1$, $\ell_1 = 1/4$, $\ell_2 = 1/8$, and $\theta = 45°$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

3.4 Effect of the masking matrix. On the top left is the array $\mathbf{A}\boldsymbol{x}$ that is of size $n \times n$, on the top right is $\mathbf{M}\mathbf{A}\boldsymbol{x}$ in array form that is $m \times m$, and on the bottom is a representation of the matrix $\mathbf{M}$ when $n = 16$ and $m = 8$. . . . . . . . . . . . . . . . . . . . . . 61

3.5 Extended domain images. On the left is an array $\mathbf{B}$ that is of size $m \times m$ and on the right is the zero-padded $\tilde{\mathbf{B}}$ that is of size $n \times n$ on the extended domain. . . . . . . . . . . . . . . . . . . . 63

3.6 $\mathbf{M}_a$ array. A standard example when using DFTs and extending the computational domain that shows the central $m \times m$ elements of $\mathbf{M}_a$ being equal to one and all the rest being equal to zero. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64

3.7 Example reconstruction. $\mathbf{X}_\alpha$ is the $n \times n$ reconstruction on the extended domain and the central $m \times m$ elements of $\mathbf{X}_\alpha$ give the reconstruction on the original domain. . . . . . . . . . . . 66

4.1 Semivariogram. A randomly generated spatial field is shown on the left and the empirical semivariogram, along with the Matérn model fit, is given on the right. The fitted hyperparameters are $\nu = 2$ and $\ell = 0.019$, which corresponds to a practical range of 0.102. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

4.2 Full $256 \times 256$ image of Main Hall at the University of Montana with $128 \times 128$ subimage. . . . . . . . . . . . . . . . . . . . . . . 73

4.3 Inpainting setup. On the left is $\mathbf{B}$, the blurred, masked, and noisy data, in array form. On the right is the $\mathbf{M}_a$ array. All array values of $\mathbf{M}_a$ outside of the central $m \times m$ region and those that correspond to the missing elements in $\mathbf{B}$ are zero. All other values are equal to one. . . . . . . . . . . . . . . . . 74

4.4 Two-dimensional image deblurring test case solutions. On the left is a plot of the Tikhonov solution and on the right is a plot of the solution obtained using the Whittle-Matérn prior with $\nu = 1$ and $\ell = 0.0339, 0.0292$ and $0.0530$ for red, green and blue intensities, respectively. . . . . . . . . . . . . . . . . . . . . . . . . 76

4.5 Transforming field from anisotropic to isotropic. The top-left image shows the original field that is anisotropic, the bottom-left image shows the rotated field so the direction of maximum anisotropy is aligned with the $x$-axis, and on the right is the rotated and scaled field that is isotropic. . . . . . . . . . . . . 78

4.6 Directional semivariograms for fields in Figure 4.5. The 12 directional semivariograms for the original, anisotropic field are shown on the left. For each of the 12 plots, semivariance value is plotted against lag distance. The direction of maximum correlation is determined to be 45° with a ratio of 3 since the distance required to pass $\gamma_{\text{crit}} = 0.9$ was largest in that direction and that distance is 3 times greater than the distance needed in the $-45°$ direction. The 12 directional semivariograms for the rotated and scaled field are shown on the right with a ratio of 1. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80

4.7 Inpainting example. The original image showing the rock layers of the Wave in northern Arizona is given on the left. The true image used in the inpainting example is given in the middle. The masked image is given on the right. . . . . . . . . . . . . 82

4.8 Inpainting solutions. The true image (left) is given along with the the isotropic solution (middle) and the anisotropic solution (right). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83

5.1 Regional field. On the left side of the image, the field is isotropic and on the right side it is anisotropic. . . . . . . . . . . . . . . . 86

5.2 CG Iteration History. The CG iteration history is shown for no preconditioner, the level 2 block circulant preconditioner, and the average precision preconditioner. The average precision preconditioner has the fastest convergence rate. . . . . . . . . 92

5.3 Making arrays. The image on the left shows a regional field with regions selected by the user. The first region has the red overlay and the second region is the remainder of the image. The image in the middle is the masking array for region one and shown on the right is the masking array for region two. . . 93

5.4 Prior weights. The image on the left shows a sample from the prior with unequal variances in the two regions. The image on the right shows a sample from the prior when weighting the regional precision matrices such that the variances in the two regions are approximately equal. . . . . . . . . . . . . . . . . . . . 94

5.5 Inpainting solutions. The true image (top-left) is given along with the masked image (top-right), the isotropic solution (center-left) the anisotropic solution (center-right), and the regional solution (bottom). . . . . . . . . . . . . . . . . . . . . . . . . . . 96

5.6 MCMC chain examples. Chain 1 on top has a low acceptance rate. Chain 2 in the middle is dependent on many previous iterations. Chain 3 on the bottom exhibits good behavior. . . 103

5.7 MCMC ACF examples. The ACFs for the top, middle, and bottom chains in Figure 5.6 are given on the left, middle, and right, respectively. . . . . . . . . . . . . . . . . . . . . . . . . . 104

5.8 2-D MCMC results. The upper-left shows the $\lambda$, $\delta$, $\alpha = \delta/\lambda$, and $\ell$ sampling chains for the red intensity. The upper-right shows the $\lambda$, $\delta$, $\alpha = \delta/\lambda$, and $\ell$ histograms for the red intensity. The middle-left shows the autocorrelation functions for $\lambda$, $\delta$, $\alpha = \delta/\lambda$, and $\ell$. The middle-right shows the difference between the 97.5th and 2.5th percentiles for each pixel in the image. The bottom-left image shows pair-wise scatter plots. The MCMC solution taken as the mean of the sampling chain of $\boldsymbol{x}$ is on the bottom right. . . . . . . . . . . . . . . . . . . . . . . . . . . . 108

5.9 2-D deblurring images. The upper-left plot shows the true Image; the upper-right shows the masked image; the lower-left image shows the MCMC sample mean reconstruction fixing $\nu$; and the lower-right image shows the MAP reconstruction using variograms. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110

# List of Tables

4.1 Statistics for inpainting MAP estimates. . . . . . . . . . . . . . 84

5.1 Statistics for regional inpainting MAP estimates. . . . . . . . . . 97

5.2 MCMC results – Red color band. . . . . . . . . . . . . . . . . . . 107

5.3 MCMC 95% credible intervals. . . . . . . . . . . . . . . . . . . . . 107

5.4 Statistics for MCMC and MAP estimates. . . . . . . . . . . . . . 109

# Chapter 1

# Introduction

Inverse problems are ubiquitous in science and engineering. They are characterized by the estimation of parameters in a mathematical model from measurements and by a high-dimensional parameter space that typically results from discretizing a function defined on a computational domain. Inverse problems are so named as a contrast to what is known as the *forward problem*. For a forward map $\mathbf{A}$ and model parameters $\boldsymbol{x}$, both given, a forward problem would seek to find $\boldsymbol{b}$ in the setting

$$\boldsymbol{b} = \mathbf{A}\boldsymbol{x} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon}$ is the measurement or model error. The forward problem is well-posed, which means it has a stable solution with respect to perturbation, and typically has a unique solution. For an inverse problem, the goal would be to infer the model parameters $\boldsymbol{x}$ when given the forward map $\mathbf{A}$ and the output $\boldsymbol{b}$.

For typical inverse problems, the process of estimating model parameters

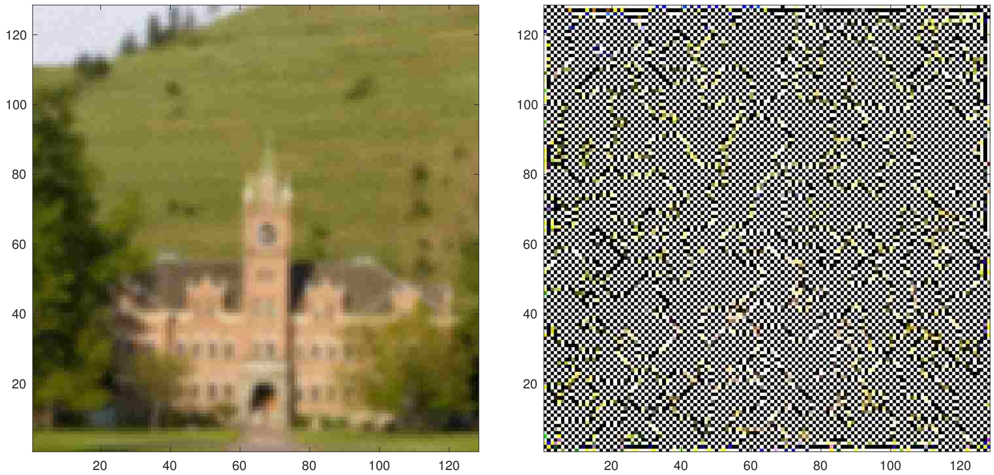

Figure 1.1: Least squares solution. On the left is a blurry image of Main Hall. On the right is the least-squares solution for the deblurring inverse problem.

from measurements is ill-posed, which means small changes in $\boldsymbol{b}$ result in large relative changes in the parameter estimates. In general, inverse problems are ill-posed because the solution does not depend continuously on the observations as a result of the form of the forward map [Bardsley, 2018]. Due to ill-posedness, the least-squares and maximum likelihood solutions to an inverse problem generally give a very poor result. This is illustrated in Figure 1.1. On the left we have a blurry image of Main Hall on the University of Montana campus. For a deblurring problem, the forward map $\mathbf{A}$ is a discretized convolution, which will be described in detail below, and the least-squares solution is given by

$$\boldsymbol{x}_{\mathrm{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\boldsymbol{b}.$$

The least-squares solution is shown on the right of Figure 1.1. It is clear that this reconstruction, which is supposed to deblur the picture, is no good.

The illustration of ill-posedness above motivates the use of regularization in the deterministic setting and the choice of a prior probability density in the Bayesian setting. In this thesis, we consider linear models of the form

$$\boldsymbol{b} = \mathbf{A}\boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_M), \tag{1.1}$$

where $\boldsymbol{b} \in \mathbb{R}^M$ is the vector of measurements, $\mathbf{A} \in \mathbb{R}^{M\times N}$ is the forward model matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the vector of unknown parameters, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_M)$ is the observation noise that follows a zero-mean Gaussian distribution with covariance matrix $\lambda^{-1}\mathbf{I}_M$, with $\mathbf{I}_M$ denoting the $M \times M$ identity. In typical inverse problems, $\mathbf{A}\boldsymbol{x}$ is the discretization of a continuous forward model $Ax$, where $A$ is a linear operator and $x$ is a function. The components of the vector $\boldsymbol{x}$ satisfy $x_i = x(\boldsymbol{u}_i)$, where $\boldsymbol{u}_i \in \mathbb{R}^d$ is the location of the $i$th element of the numerical grid.

## 1.1 The Matrix A and Boundary Conditions

Modeling $A$ correctly is crucial to obtaining a good solution to an inverse problem. The form of $A$ depends on what type of inverse problem is being solved. In many applications, including the deblurring example we used above, $A$ takes the form of a convolution and so, in one dimension, the physical model is

$$b(u) = \int_{-\infty}^{\infty} A(u-u')x(u')du'.$$

The domain of $b$, $A$, and $x$ are usually determined by the measurement device or, for an image, by the field of view. Here we assume $A$ is measured on the interval $[-1, 1]$ and, for simplicity, assume $b$ and $x$ are both measured on the interval $\Omega = [0, 1]$. The interval $\Omega$ is known as the computational domain.

This means the model can be re-expressed as

$$b(u) = \int_{-1}^{2} A(u - u')x(u')du', \quad 0 \leq u \leq 1.$$

To obtain a useable matrix to represent $A$, we need to discretize the integral. Midpoint quadrature can be used with step size $h = 1/n$ to perform the discretization in $A$ and $x$. In general, we allow for a step size of $1/m$ for discretizing $b$, but to simplify notation, we let $m = n$ here so $b$ will also have a discretization step size of $h = 1/n$. Letting $u_i = (i - 1/2)h$, $u'_j = (j - 1/2)h$ and defining $a_{i-j} := A(u_i - u'_j) = A((i - j)h)$, $b_i := b(u_i)$ and $x_j := x(u'_j)$ for $i = 1, \ldots, n$ and $j = -n + 1, \ldots, 2n$, we have

$$b_i = h \sum_{j=i-n+1}^{i+n-1} a_{i-j} x_j, \quad i = 1, \ldots, n.$$

In matrix-vector notation, this has the form

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = h \begin{bmatrix} a_{n-1} & \cdots & a_0 & \cdots & a_{-n+1} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & a_{n-1} & \cdots & a_0 & \cdots & a_{-n+1} \end{bmatrix} \begin{bmatrix} x_{-n+2} \\ \vdots \\ x_1 \\ \vdots \\ x_n \\ \vdots \\ x_{2n-1} \end{bmatrix}. \tag{1.2}$$

Notice that the matrix $\mathbf{A}$ is toeplitz, which means it has constant diagonals, and that $a_{i-j} = 0$ for $|i - j| \geq n$. The values of the matrix $\mathbf{A}$ are chosen depending on the inverse problem at hand. For deblurring problems, the $a_{i-j}$

values are taken from a kernel, one of which is the Gaussian kernel. Using the Gaussian kernel, we have

$$ha_{i-j} = hA((i-j)h) = \frac{h}{\sqrt{2\pi\gamma^2}} \exp\left(-\frac{((i-j)h)^2}{2\gamma^2}\right), \quad 1 \le i, j \le n,$$

where $\gamma^2$ is the variance of the blurring kernel that corresponds to the amount of blurring in the image. The blurrier the image, the larger $\gamma^2$ should be.

Clearly the system (1.2) depends on $(x_{-n+2}, \dots, x_0)$ and $(x_{n+1}, \dots, x_{2n-1})$, all of which are outside the computational domain. For example, $x_0 = x\,(-h/2)$ and $-h/2$ is outside of $\Omega = [0, 1]$. However, the elements of $\boldsymbol{b}$ that are near the boundary of the domain, i.e. close to zero or one, still rely on these elements of $\boldsymbol{x}$. Thus, it is common to impose *boundary conditions* so that (1.2) is a square, $n \times n$ linear system. There are several types of boundary conditions, the most common of which are the Dirichlet (also called zero), periodic, and Neumann boundary conditions. In this thesis, we focus only on the zero and periodic boundary conditions. With zero boundary conditions, we assume $(x_{-n+2}, \dots, x_0) = (x_{n+1}, \dots, x_{2n-1}) = \mathbf{0}$ and with periodic boundary conditions, we assume $(x_{-n+2}, \dots, x_0) = (x_2, \dots, x_n)$ and $(x_{n+1}, \dots, x_{2n-1}) = (x_1, \dots, x_{n-1})$. Hence, for an array $\mathbf{X}$, the one-dimensional representations of the zero and periodic boundary conditions are illustrated, respectively, by the following extensions of an array $\mathbf{X}$ [Hansen et al., 2006, Bardsley and Luttman, 2015]:

$$\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hline \end{array} \quad \text{and} \quad \begin{array}{|c|c|c|}\hline \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \hline \end{array}.$$

When zero boundary conditions are applied in the one-dimensional case, (1.2) reduces to

$$\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = h \begin{bmatrix} a_0 & \cdots & a_{-n+1} \\ \vdots & \ddots & \vdots \\ a_{n-1} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

and when using periodic boundary conditions, (1.2) can be written

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} = h \begin{bmatrix} a_0 & (a_{n-1}+a_{-1}) & \cdots & (a_2+a_{-n+2}) & (a_1+a_{-n+1}) \\ (a_1+a_{-n+1}) & a_0 & (a_{n-1}+a_{-1}) & & (a_2+a_{-n+2}) \\ \vdots & (a_1+a_{-n+1}) & a_0 & \ddots & \vdots \\ (a_{n-2}+a_{-2}) & & \ddots & \ddots & (a_{n-1}+a_{-1}) \\ (a_{n-1}+a_{-1}) & (a_{n-2}+a_{-2}) & \cdots & (a_1+a_{-n+1}) & a_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}.$$

The matrix $\mathbf{A}$ in this case is circulant, an important property that will be defined in Chapter 3 and will play an important role later.

As is shown in [Bardsley, 2018], these results can be generalized to two dimensions, in which case $\Omega = [0,1] \times [0,1]$ and the linear system will take the form

$$b_{ik} = h \sum_{j=i-n+1}^{i+n-1} \sum_{l=k-n+1}^{k+n-1} a_{i-j} a_{l-k} x_{jl}, \quad i,k = 1,\ldots,n.$$

The two-dimensional representations of the zero and periodic boundary conditions are given by

$$\begin{array}{ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \quad \text{and} \quad \begin{array}{ccc} \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \end{array}.$$

We let $m = n$ in this illustration, but in general $\mathbf{A} \in \mathbb{R}^{m \times n}$ in one dimension and $\mathbf{A} \in \mathbb{R}^{m^2 \times n^2}$ in two dimensions. Thus, using the notation from (1.1),

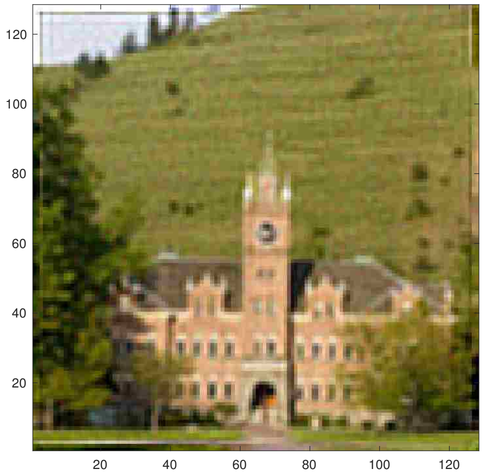

Figure 1.2: Boundary artifacts. When boundary conditions are inappropriate, the reconstruction may contain unrealistic artifacts near the borders.

$N = n$ and $M = m$ when working in one dimension, while $N = n^2$ and $M = m^2$ in a two-dimensional environment.

Care must be taken when selecting boundary conditions because those that do not accurately represent the true unknown outside of the computational domain may lead to a reconstruction of $\boldsymbol{x}$ that contains unrealistic boundary artifacts. An example of this is seen in Figure 1.2. In this example, periodic boundary conditions were used that led to some of the blue sky in the top-left corner of the image being replaced with green grass from the bottom left. One way to combat this to to extend the computational domain. A more thorough discussion on domain extension will be presented in Chapter 3.

Once the form of $A$ has been determined, boundary conditions have been applied, and we have an expression for the matrix $\mathbf{A}$, the conditional probability density function of the random vector $\boldsymbol{b}$ in (1.1) is

$$p(\boldsymbol{b}|\boldsymbol{x}, \lambda) \propto \exp\left(-\frac{\lambda}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2\right), \tag{1.3}$$

where $\propto$ denotes proportionality and $\|\cdot\|$ denotes the $\ell^2$-norm. The maximizer of $p(\boldsymbol{b}|\boldsymbol{x},\lambda)$ with respect to $\boldsymbol{x}$ is known as the maximum likelihood estimator. The maximum likelihood estimator in this context is equivalent to the least-squares solution, which means it is also unstable with respect to errors in $\boldsymbol{b}$ due to ill-posedness. Therefore, we need a way to stabilize the inverse problem reconstruction.

## 1.2 Bayesian Inverse Problems

There are various methods that can be used to stabilize the solutions of inverse problems, but they all involve some form of regularization. In this thesis, we take the Bayesian approach [Kaipio and Somersalo, 2005], which requires the definition of a prior probability density function on $\boldsymbol{x}$.

Bayesian methods, at their core, are derived from conditional probability. It is well known that the conditional probability of an event $A$ given an event $B$, denoted $P(A|B)$, can be written as

$$P(A|B) = \frac{P(A\cap B)}{P(B)}, \quad P(B) \neq 0.$$

Then by that same property, $P(A\cap B) = P(B|A)P(A)$. Additionally, when the sample space, $S$, can be partitioned such that $S = \sum_{j=1}^{k} A_j$, with each $A_i$ mutually exclusive, i.e., $A_i \cap A_j = \emptyset$ (the empty set), and $P(A_j) \neq 0$ for $j = 1,\dots,k$, then

$$P(B) = \sum_{j=1}^{k} P(A_j \cap B) = \sum_{j=1}^{k} P(B|A_j)P(A_j).$$

Therefore,

$$\begin{aligned} P(A_i|B) &= \frac{P(A_i \cap B)}{P(B)} \\ &= \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{k} P(B|A_j)P(A_j)}, \end{aligned}$$

again assuming that $P(B) \neq 0$. This is an expression of Bayes' Theorem in the discrete case. We can use an analogous expression in the multi-dimensional, continuous random variable case to obtain an expression for $p(\boldsymbol{x}|\boldsymbol{b})$:

$$p(\boldsymbol{x}|\boldsymbol{b}) = \frac{p(\boldsymbol{b}|\boldsymbol{x})p(\boldsymbol{x})}{\int p(\boldsymbol{b}|\boldsymbol{t})p(\boldsymbol{t})d\boldsymbol{t}} \propto p(\boldsymbol{b}|\boldsymbol{x})p(\boldsymbol{x}).$$

In this expression, $p(\boldsymbol{x}|\boldsymbol{b})$ is known as the posterior distribution, $p(\boldsymbol{b}|\boldsymbol{x})$ is the likelihood, and $p(\boldsymbol{x})$ is the prior. The prior will often be conditioned on other unknowns called hyperparameters, e.g. $p(\boldsymbol{x}|\delta)$, where $\delta$ is a hyperparameter.

In this thesis, we make the assumption that the prior is Gaussian of the form $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, (\delta\mathbf{P})^{-1})$, which has probability density function

$$p(\boldsymbol{x}|\delta) \propto \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}\boldsymbol{x}\right), \tag{1.4}$$

where $\mathbf{P}$ is the precision (inverse-covariance) matrix. Now that we have defined the prior (1.4) and the likelihood (1.3), using Bayes' law, we multiply them together to obtain the posterior density function

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{b}, \lambda, \delta) &\propto p(\boldsymbol{b}|\boldsymbol{x}, \lambda)p(\boldsymbol{x}|\delta) \\ &\propto \exp\left(-\frac{\lambda}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 - \frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}\boldsymbol{x}\right), \end{aligned} \tag{1.5}$$

whose maximizer is known as the maximum a posteriori (MAP) estimator. The MAP estimator can be equivalently expressed as

$$\begin{aligned}\boldsymbol{x}_\alpha &= \arg\min_{\boldsymbol{x}} \left\{ \frac{\lambda}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}\boldsymbol{x} \right\} \\ &= (\lambda\mathbf{A}^T\mathbf{A} + \delta\mathbf{P})^{-1}\lambda\mathbf{A}^T\boldsymbol{b} \\ &= (\mathbf{A}^T\mathbf{A} + \alpha\mathbf{P})^{-1}\mathbf{A}^T\boldsymbol{b},\end{aligned}$$

for $\alpha = \delta/\lambda$. Thus, the regularization term is $\alpha\mathbf{P}$ and $\alpha$ is referred to as the regularization parameter. Our primary focus in this thesis is to provide formulations and parameter selection techniques for prior precision matrices that have an intuitive interpretation and can be used to regularize, and therefore solve, a wide variety of problems.

## 1.3 The Matérn Class of Covariance Matrices and Whittle-Matérn Priors

It remains to define the prior covariance matrix $\mathbf{C} = \mathbf{P}^{-1}$. The Matérn class of covariance matrices has garnered much praise [Stein, 2012] for its flexibility in capturing many covariance structures and its allowance of direct control over the degree of correlation in the vector $\boldsymbol{x}$ [Guttorp and Gneiting, 2006]. The Matérn covariance matrix is defined by the Matérn covariance function, which was first formulated by Matérn in 1947 [Matérn, 2013],

$$C(r) = \sigma^2 \frac{(r/\ell)^\nu K_\nu(r/\ell)}{2^{\nu-1}\Gamma(\nu)}, \tag{1.6}$$

where $r$ is the separation distance; $K_\nu(\cdot)$ is the modified Bessel function of the second kind of order $\nu$ [Andrews, 1992]; $\Gamma(\cdot)$ is the gamma function; $\ell > 0$ is the range parameter; $\nu > 0$ is the smoothness parameter; and $\sigma^2$ is the marginal variance. Omitting $\sigma^2$ gives the Matérn correlation function. In the isotropic case, when the covariance depends only on the distance between elements, given the covariance parameters $\sigma^2, \nu$, and $\ell$, one can obtain the covariance matrix $\mathbf{C}$ of a vector $\boldsymbol{x} = (x_1, \ldots, x_N)^T$ with spatial positions $\{\boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_N^T\} \subset \mathbb{R}^d$ by letting

$$[\mathbf{C}]_{ij} = \mathrm{Cov}(x_i, x_j) = C(\|\boldsymbol{u}_i - \boldsymbol{u}_j\|),$$

where $C$ is defined by (1.6).

The parameters of the Matérn covariance function are not as straightforward to interpret as the parameters of some other covariance functions. When $\nu$ is small ($\nu \to 0^+$), the spatial process is said to be rough, and when it is large ($\nu \to \infty$), the process is smooth [Guttorp and Gneiting, 2006, Minasny and McBratney, 2005]. Figure 1.3 shows how the covariance function behaves with different values of $\ell$ and $\nu$: on the left, $\ell = \sigma^2 = 1$ and $\nu$ varies, while on the right $\nu = \sigma^2 = 1$ and $\ell$ varies. Note that as $\nu$ increases, the behavior at small lags changes, leading to more correlation at smaller distances and a larger *practical range*, which is defined to be the distance at which the correlation is equal to 0.05; in Figure 1.3, this is the distance at which the covariance function intersects the horizontal line. Meanwhile, as $\ell$ decreases, the decay rate of the covariance increases considerably, which decreases the practical range. Although $\ell$ is known as the range parameter, the parameter $\nu$ also affects the practical range. In [Lindgren et al., 2011], a range approximation $\rho = \ell\sqrt{8\nu}$ is

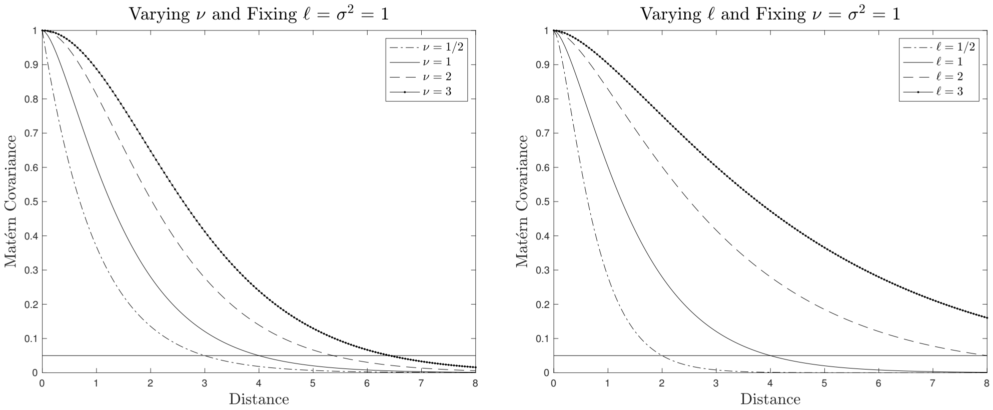

Figure 1.3: Behavior of the Matérn covariance function. The smoothness parameter, $\nu$, primarily affects the covariance at small distances whereas the range parameter, $\ell$, mainly affects the decay rate of the covariance. The horizontal line corresponds to a covariance value of 0.05 and the practical range is the distance at which the covariance intersects this line.

used where $C(\rho) \approx 0.10$.

When $\nu = p + 1/2$ for $p \in \mathbb{N}$, the Matérn covariance function simplifies into some very familiar forms. We will illustrate this simplification here for $\nu = 1/2$. First, there are several different representations of $K_\nu$ one can use, including ones involving integrals and infinite series. One such representation is the following integral form:

$$K_\nu(r/\ell) = \frac{\Gamma(\nu + 1/2)(2r/\ell)^\nu}{\sqrt{\pi}} \int_0^\infty \frac{\cos(t)dt}{(t^2 + (r/\ell)^2)^{\nu+1/2}}.$$

Thus, the Matérn covariance when $\nu = 1/2$ is

$$\begin{aligned} C_{1/2}(r) &= \sigma^2 \frac{1}{2^{1/2-1}\Gamma(1/2)} \left(r/\ell\right)^{1/2} K_{1/2}\left(r/\ell\right) \\ &= \sigma^2 \frac{1}{2^{-1/2}\Gamma(1/2)} \left(r/\ell\right)^{1/2} \frac{\Gamma(1)(2r/\ell)^{1/2}}{\sqrt{\pi}} \int_0^\infty \frac{\cos(t)dt}{t^2 + (r/\ell)^2} \end{aligned}$$

Using contour integration [Bak et al., 2010], one can show $\int_0^\infty \frac{\cos(t)dt}{t^2+(r/\ell)^2} = \frac{\pi e^{-r/\ell}}{2r/\ell}$ and so,

$$\begin{aligned}
C_{1/2}(r) &= \sigma^2 \frac{1}{2^{-1/2}\Gamma(1/2)} (r/\ell)^{1/2} \frac{\Gamma(1)(2r/\ell)^{1/2}}{\sqrt{\pi}} \frac{\pi e^{-r/\ell}}{2r/\ell} \\
&= \sigma^2 \frac{\sqrt{2}}{\sqrt{\pi}} (r/\ell)^{1/2} \frac{\Gamma(1)(2r/\ell)^{1/2}}{\sqrt{\pi}} \frac{\pi e^{-r/\ell}}{2r/\ell} \\
&= \sigma^2 \frac{2r/\ell}{\pi} \frac{\pi e^{-r/\ell}}{2r/\ell} \\
&= \sigma^2 e^{-r/\ell} \\
&= \sigma^2 e^{-3r/\rho},
\end{aligned}$$

which is the Exponential covariance function with range parameter $\rho = 3\ell$.

In general, for $\nu = p + 1/2$ [Gneiting et al., 2010],

$$C_{p+1/2}(r) = \sigma^2 e^{-r/\ell} \sum_{j=0}^{p} \frac{(p+j)!}{(2p)!} \binom{p}{j} (2r/\ell)^{p-j}.$$

So when $\nu = 3/2$, $C_{3/2}(r) = \sigma^2(1 + r/\ell)e^{-r/\ell}$, which is the second-order autoregressive function, and when $\nu = 5/2$, $C_{5/2}(r) = \sigma^2(1 + r/\ell + \frac{1}{3}r^2/\ell^2)e^{-r/\ell}$, which is the third-order autoregressive function.

The Matérn covariance function is also related to the Gaussian covariance function, which is sometimes referred to as the squared exponential distribution. It can be shown [Schabenberger and Gotway, 2017] that as $\nu \to \infty$,

$$C_\nu(r) \to \sigma^2 \exp\left\{-\frac{r^2}{4\nu\ell^2}\right\} = \sigma^2 \exp\left\{-3\left(\frac{r}{\rho}\right)^2\right\}$$

which is the Gaussian model with range parameter $\rho = 2\ell\sqrt{3\nu}$. We verify this convergence numerically on the left in Figure 1.4 using $\ell = \sigma^2 = 1$. On the vertical axis, we have the norm of the difference between the two covariance function for 100 equally-spaced distances ($r$ values) between 0 and 6, plotted on the log scale. That is,

$$\left\| \frac{1}{2^{\nu-1}\Gamma(\nu)} (\boldsymbol{r})^{\nu} K_{\nu}(\boldsymbol{r}) - \exp\left\{ -\frac{\boldsymbol{r}^2}{4\nu} \right\} \right\|_2, \quad \boldsymbol{r} = (0, 0.06, 0.12, \ldots, 5.94, 6).$$

On the horizontal axis we have different values of the smoothness parameter $\nu$ between 1 and 100. 100 is the largest considered $\nu$ value because many computers cannot evaluate the Matérn covariance for $\nu$ much larger than 100. We can see on the left side of Figure 1.4 that the norm difference is converging to

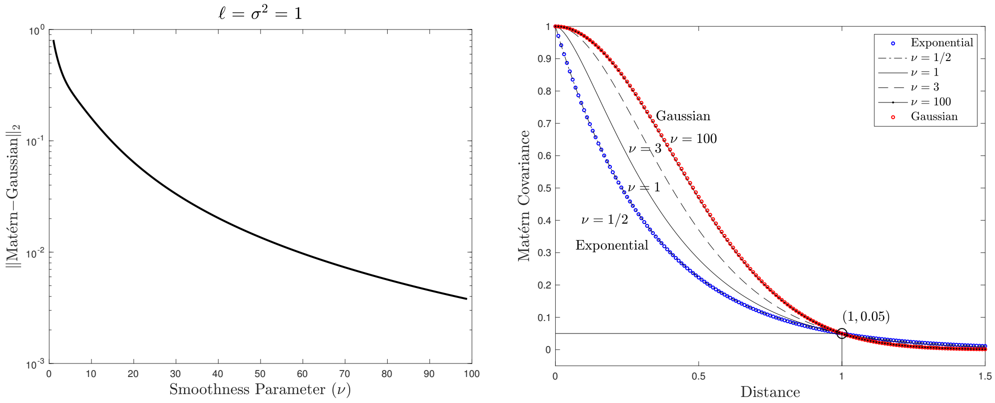

Figure 1.4: Convergence to the Gaussian covariance. On the left, we show that the norm difference between the Matérn and the Gaussian covariance function converges to zero as $\nu$ increases, plotted on the log scale. On the right, we plot the Exponential and Gaussian covariance functions along with the Matérn covariance for various values of $\nu$ with a fixed practical range of 1 to further illustrate that the Matérn covariance is equal to the Exponential covariance when $\nu = 1/2$ and converges to the Gaussian covariance as $\nu \to \infty$.

zero, which numerically verifies the fact that the Gaussian covariance function is the limiting covariance model as $\nu \to \infty$.

The right side of Figure 1.4 now adds the Gaussian and Exponential covariance functions to a plot of Matérn covariance functions with different $\nu$ values. All $\ell$ values are adjusted so the practical range of each curve is equal to one. When $\nu = 100$, the Matérn covariance is virtually indistinguishable from the Gaussian covariance and when $\nu = 1/2$, the Matérn and Exponential covariance functions are identical.

Despite the benefits of using the Matérn class of covariance matrices, its use can be problematic for inverse problems because computing the precision matrix $\mathbf{P}$, which is what appears in the posterior (1.5), requires inverting a dense $N \times N$ matrix. Using the fast Fourier transform (FFT) [Wood and Chan, 1994, Dietrich and Newsam, 1997, Bardsley, 2018] to operate with $\mathbf{P}$ and $\mathbf{C}$ more efficiently is recommended if $\boldsymbol{x}$ is defined on a regular grid and periodic boundary conditions are assumed. In other cases, it is useful that the Matérn covariance function has a direct connection to a class of elliptic SPDEs [Lindgren et al., 2011] whose numerical discretization yields sparse precision matrices, $\mathbf{P}$, that are computationally feasible to work with even when $N$ is large. Connections of this type were first shown to exist by Whittle in [Whittle, 1954], where he showed the connection held for a special case of the Matérn covariance class. Hence, priors that depend on this connection are often referred to as Whittle-Matérn priors. The connection between the general Matérn covariance function and SPDEs has been used in a wide range of applications for defining computationally feasible priors for high-dimensional problems [Roininen et al., 2016, Roininen et al., 2014, Monterrubio-Gómez et al., 2018]. Moreover, work has been done in establishing convergence theorems for, and lattice approximations of, these Whittle-Matérn priors [Roininen et al., 2013].

The remainder of the dissertation is organized as follows. In Chapter 2, we describe in detail the connection between zero-mean Gaussian processes with the isotropic Matérn covariance operator and those that arise as solutions of a class of elliptic SPDEs. This is then extended to the anisotropic case. In Chapter 3, we give an in-depth look at discretizing the SPDE for use as a sparse precision matrix as well as the effect that discretizing and imposing boundary conditions has on the covariance operator. In Chapter 4, we show how to estimate the parameters in the Whittle-Matérn prior using the semivariogram method, and then we show how to use this approach to define the prior when solving a Bayesian inverse problem. Numerical examples are used to show this method works well. Chapter 5 contains two extensions. First, we extend these ideas to consider images with regions that require different covariance structures. We then compare and contrast using the fully-Bayesian method to achieve point estimates and uncertainty quantification in our estimates. We end with conclusions in Chapter 6.

# Chapter 2

# Theoretical Results

This chapter aims to provide detailed descriptions and proofs of the theory necessary for the semivariogram method to be valid. We begin with stochastic partial differential equations. A stochastic partial differential equation (SPDE) is a PDE that includes a random force term or coefficient. In this chapter, we will show that the isotropic Whittle-Matérn class of priors can be specified as the solution of the SPDE

$$(1 - \ell^2 \Delta)^{\beta/2} x(\boldsymbol{u}) = \mathcal{W}(\boldsymbol{u}), \quad \boldsymbol{u} \in \mathbb{R}^d, \quad \beta = \nu + d/2, \quad \ell, \nu > 0, \tag{2.1}$$

where $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{du_i^2}$ is the Laplacian operator in $d$ dimensions and $\mathcal{W}$ is spatial Gaussian white noise with unit variance, which we define below. Although this connection has been shown to exist [Whittle, 1954, Lindgren et al., 2011, Roininen et al., 2014], here we provide a detailed derivation and original proof of this result. This derivation is based on the Green's function of the differential operator. For other linear operators with sufficient smoothness, e.g., the one in the Stokes equations and the one in the standard heat equation, the corresponding SPDEs can be used to define different Gaussian processes [Lord et al., 2014]. The method we employ here provides a potential way to derive the covariance functions of the Gaussian processes induced by other linear SPDEs as well.

## 2.1 Preliminaries

Before deriving the solution of (2.1), we need some preliminary definitions.

### 2.1.1 Gaussian Fields

A stochastic process $\{x(\boldsymbol{u}), \boldsymbol{u} \in \Omega\}$, with $\Omega \subset \mathbb{R}^d$, is a *Gaussian field* [Rue and Held, 2005] if for any $k \geq 1$ and any locations $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k \in \Omega$, the vector $[x(\boldsymbol{u}_1), \dots, x(\boldsymbol{u}_k)]^T$ is a normally distributed random vector with mean

$$\boldsymbol{\mu} = \Big[E[x(\boldsymbol{u}_1)], \dots, E[x(\boldsymbol{u}_k)]\Big]^T,$$

where $E[\,\cdot\,]$ denotes expected value, and covariance matrix

$$\begin{aligned} [\mathbf{C}]_{ij} &= \mathrm{Cov}(x(\boldsymbol{u}_i), x(\boldsymbol{u}_j)) \\ &= E[(x(\boldsymbol{u}_i) - E[x(\boldsymbol{u}_i)])(x(\boldsymbol{u}_j) - E[x(\boldsymbol{u}_j)])], \quad \text{for } 1 \leq i, j \leq k. \end{aligned}$$

The covariance function is defined $C(\boldsymbol{u}_i, \boldsymbol{u}_j) := \mathrm{Cov}(x(\boldsymbol{u}_i), x(\boldsymbol{u}_j))$. It is necessary that the covariance function is positive definite, i.e., for any $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_k\}$, with $k \geq 1$, the covariance matrix $\mathbf{C}$ defined above is positive definite. The Gaussian field is called *stationary* if the mean is constant and the covariance function satisfies $C(\boldsymbol{u}, \boldsymbol{v}) = C(\boldsymbol{u} - \boldsymbol{v})$ and *isotropic* if $C(\boldsymbol{u}, \boldsymbol{v}) = C(\|\boldsymbol{u} - \boldsymbol{v}\|)$.

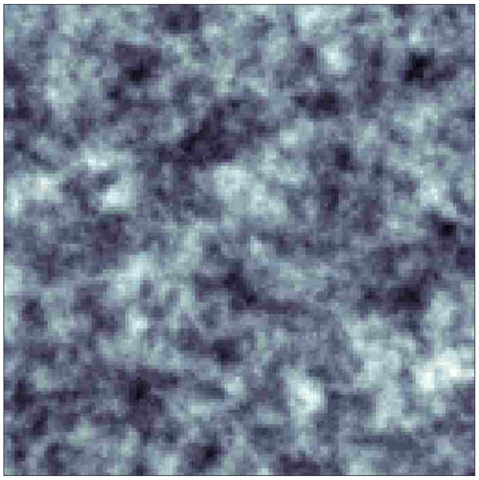

Figure 2.1: Isotropic field. The correlation between any two points depends only on the distance between those points and is independent of direction.

An example of an isotropic field is given in Figure 2.1.

### 2.1.2 White Noise

The term *white noise* [Lord et al., 2014, Walsh, 1986] comes from light. White light is a homogeneous mix of wavelengths, as opposed to colored light, which is a heterogeneous mix of wavelengths. In a similar way, white noise contains a homogeneous mix of all the different basis functions. The mixing of these basis functions is determined by a random process. When this random process is Gaussian, we have *Gaussian white noise*. Basis functions of a function space have the property that any continuous function in that function space can be represented as a linear combination of the basis functions. Consider a domain $\Omega$ and let $\{\phi_j : j = 1, 2, \dots\}$ be an orthonormal basis of $L^2(\Omega)$ where $L^2(\Omega) = \{f : \Omega \to \mathbb{R} \mid \int_\Omega |f(x)|^2 dx < \infty\}$. Then Gaussian white noise is defined by

$$\mathcal{W}(\boldsymbol{u}) = \sum_{j=1}^{\infty} \xi_j \phi_j(\boldsymbol{u}), \quad \xi_j \overset{iid}{\sim} \mathcal{N}(0, \eta^2), \tag{2.2}$$

where $\boldsymbol{u}$ refers to spatial location. With this definition, it is clear that Gaussian white noise has mean zero: $E[\mathcal{W}(\boldsymbol{u})] = \sum_{j=1}^{\infty} E[\xi_j]\, \phi_j(\boldsymbol{u}) = 0$. Moreover, one can show that $\text{Cov}\left(\mathcal{W}(\boldsymbol{u}), \mathcal{W}(\boldsymbol{v})\right) = \eta^2 \delta_f(\boldsymbol{u}-\boldsymbol{v})$, where $\delta_f(\cdot)$ is the *Dirac delta function* [Hassani, 2000], also known as the delta distribution. We include the subscript $f$ to differentiate the delta function from the $\delta$ hyperparameter used elsewhere in this thesis. A well-known and very important property of the Dirac delta function is that it satisfies the *sifting property*:

$$f(\boldsymbol{u}) = \int_{\mathbb{R}^d} \delta_f(\boldsymbol{u} - \boldsymbol{v}) f(\boldsymbol{v}) d\boldsymbol{v},$$

which is so named due to the fact that applying the delta function to the function $f(\boldsymbol{v})$ in the integral sifts out a single value of $f(\boldsymbol{v})$ [Bracewell, 1986].

### 2.1.3 Green's Functions

We now consider differential equations of the form $Lx(\boldsymbol{u}) = f(\boldsymbol{u})$, $\boldsymbol{u} \in \mathbb{R}^d$, where $L$ is a linear, differential operator. A *Green's function* [Gockenbach, 2005, Stakgold and Holst, 2011], $g$, of $L$ is any solution of $Lg(\boldsymbol{u}, \boldsymbol{v}) = \delta_f(\boldsymbol{u}-\boldsymbol{v})$. The Green's function can be used to solve $Lx(\boldsymbol{u}) = f(\boldsymbol{u})$, as we show here.

Let $L : V \to V$ be a linear mapping. Using the weak form of the PDE, we have $\langle Lx, h \rangle = \langle f, h \rangle\ \forall h \in V$, where $\langle \cdot, \cdot \rangle$ denotes an inner product. Since this holds for all functions $h \in V$, it holds for our specific Green's function, $g$. So, $\langle Lx, g \rangle = \langle f, g \rangle$. This implies $\langle x, L^* g \rangle = \langle f, g \rangle$ where $L^*$ is the adjoint of $L$ [D'Angelo, 2013]. Our specific operator is self-adjoint, which means $L = L^*$, and so $L^*g = Lg = \delta_f$ since $g$ is a Green's function. Thus, we have

$$\begin{aligned}
\langle Lx, g\rangle &= \langle f, g\rangle \\
\implies \langle x, L^*g\rangle &= \langle f, g\rangle \\
\implies \langle x, Lg\rangle &= \langle f, g\rangle \\
\implies \langle x, \delta_f\rangle &= \langle f, g\rangle.
\end{aligned}$$

Therefore, using the sifting property of the Dirac delta function, the solution of the equation $Lx(\boldsymbol{u}) = f(\boldsymbol{u})$ can be written as

$$x(\boldsymbol{u}) = \int_{\mathbb{R}^d} g(\boldsymbol{u}, \boldsymbol{v}) f(\boldsymbol{v}) d\boldsymbol{v} \tag{2.3}$$

in the $L^2$ inner product.

### 2.1.4 Fourier Transform

We define the $d$-dimensional Fourier transform [Sneddon, 1995, Kwaśnicki, 2017] as

$$\widehat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\boldsymbol{u}) e^{-i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{u}$$

with inverse Fourier transform

$$f(\boldsymbol{u}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \widehat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega},$$

where $\boldsymbol{\omega} \in \mathbb{C}^d$ are the coordinates in the Fourier-transformed space; the hat $(\widehat{f})$ notation denotes the Fourier transform of a function $f$; and $\boldsymbol{\omega} \cdot \boldsymbol{u}$ denotes the dot product between $\boldsymbol{\omega}$ and $\boldsymbol{u}$. We also use $\mathcal{F}\{\cdot\}(\boldsymbol{\omega})$ to denote the Fourier transform such that $\widehat{f(\boldsymbol{u})} = \mathcal{F}\{f(\boldsymbol{u})\}(\boldsymbol{\omega}) = \widehat{f}(\boldsymbol{\omega})$.

Now, we consider applying the Laplacian operator in $d$ dimensions: $\Delta = \nabla^2 = \sum_{i=1}^{d} \frac{\partial^2}{\partial u_i^2}$. We will apply $\Delta$ to $e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} = e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)}$:

$$\nabla e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} = \begin{bmatrix} i\omega_1 e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} & \dots & i\omega_d e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} \end{bmatrix}^T,$$

and so

$$\begin{aligned}
\Delta e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} &= \nabla \cdot \nabla e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} \\
&= \nabla \cdot \begin{bmatrix} i\omega_1 e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} & \dots & i\omega_d e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} \end{bmatrix}^T \\
&= i^2 \omega_1^2 e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} + \cdots + i^2 \omega_d^2 e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} \\
&= -\omega_1^2 e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} + \cdots + -\omega_d^2 e^{i(\omega_1 u_1 + \cdots + i^2 \omega_d u_d)} \\
&= -(\omega_1^2 + \cdots + \omega_d^2) e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} \\
&= -\|\boldsymbol{\omega}\|^2 e^{i(\omega_1 u_1 + \cdots + \omega_d u_d)} \\
&= -\|\boldsymbol{\omega}\|^2 e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}}.
\end{aligned}$$

When taking the Fourier transform of the Laplacian applied to a function, we have

$$\Delta f(\boldsymbol{u}) = \Delta \left( \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \widehat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega} \right).$$

Since $\Delta$ is a linear operator and is acting only on $\boldsymbol{u}$, we can move it inside the integral:

$$\Delta f(\boldsymbol{u}) = \Delta \left( \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \widehat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega} \right)$$

$$
\begin{aligned}
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \widehat{f}(\boldsymbol{\omega}) \Delta e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega} \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \widehat{f}(\boldsymbol{\omega}) (-\|\boldsymbol{\omega}\|^2) e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega} \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} (-\|\boldsymbol{\omega}\|^2) \widehat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega}.
\end{aligned}
$$

Thus, $\widehat{\Delta f(\boldsymbol{u})} = (-\|\boldsymbol{\omega}\|^2)\widehat{f}(\boldsymbol{\omega})$ in the sense of distributions.

We can iterate on this to obtain

$$
\Delta^n f(\boldsymbol{u}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} (-\|\boldsymbol{\omega}\|^2)^n \widehat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{\omega}, \quad n = 0, 1, 2\ldots
$$

and so $\widehat{\Delta^n f(\boldsymbol{u})} = (-\|\boldsymbol{\omega}\|^2)^n \widehat{f}(\boldsymbol{\omega})$, which can be generalized as a pseudo-differential operator for non-integer $n$. Now using symbols, we can define an operator, $L$, by the symbol $(1 + \ell^2\|\boldsymbol{\omega}\|^2)^\beta$. This symbol corresponds to $L = (1 - \ell^2\Delta)^\beta$ and so the Fourier transform of $(1 - \ell^2\Delta)^\beta f(\boldsymbol{u})$ is said to be $(1 + \ell^2\|\boldsymbol{\omega}\|^2)^\beta \widehat{f}(\boldsymbol{u})$.

It will also be important to know the Fourier transform of the Dirac delta function, $\delta_f(\cdot)$.

$$
\begin{aligned}
\widehat{\delta_f}(\boldsymbol{\omega}) &= \int_{\mathbb{R}^d} \delta_f(\boldsymbol{u}) e^{-i\boldsymbol{\omega}\cdot\boldsymbol{u}} d\boldsymbol{u} \\
&= e^{-i\boldsymbol{0}\cdot\boldsymbol{u}} \quad \text{(using the sifting property)} \\
&= 1.
\end{aligned}
$$

Thus, the Fourier transform of the Dirac delta function is the constant 1 in the sense of distributions.

## 2.2 Isotropic Matérn Connection

In this section, we will prove the following theorem concerning the solution of the SPDE (2.1).

**Theorem 2.2.1.** *The solution $x(\boldsymbol{u})$ of (2.1) is a Gaussian field with mean zero and Matérn covariance function defined by (1.6).*

*Proof.* To begin, we note that the Green's function for (2.1) is the solution of

$$(1-\ell^2\Delta)^{\beta/2}g(\boldsymbol{u},\boldsymbol{v}) = \delta_f(\boldsymbol{v}-\boldsymbol{u}). \tag{2.4}$$

Using (2.3), the solution to (2.1) is given by

$$x(\boldsymbol{u}) = \int_{\mathbb{R}^d} g(\boldsymbol{u},\boldsymbol{v})\mathcal{W}(\boldsymbol{v})d\boldsymbol{v}, \tag{2.5}$$

making $x(\boldsymbol{u})$ a Gaussian field since it is a linear transformation of Gaussian white noise.

We now compute the mean and covariance of the Gaussian field, $x(\boldsymbol{u})$, defined by (2.5). Since the Green's function is a strictly-positive, symmetric, and rapidly decaying function, we can apply Fubini's theorem [Saks, 1937] to obtain the mean of $x(\boldsymbol{u})$:

$$E[x(\boldsymbol{u})] = E\left[\int_{\mathbb{R}^d} g(\boldsymbol{u},\boldsymbol{v})\mathcal{W}(\boldsymbol{v})d\boldsymbol{v}\right] = \int_{\mathbb{R}^d} g(\boldsymbol{u},\boldsymbol{v})E\left[\mathcal{W}(\boldsymbol{v})\right]d\boldsymbol{v} = 0.$$

Since $x(\boldsymbol{u})$ has mean zero, the covariance is given by

$$\begin{aligned}\mathrm{Cov}(x(\boldsymbol{u}),x(\boldsymbol{u'})) &= E[x(\boldsymbol{u})x(\boldsymbol{u'})]\\ &= E\left[\int_{\mathbb{R}^d} g(\boldsymbol{u},\boldsymbol{v})\mathcal{W}(\boldsymbol{v})d\boldsymbol{v}\int_{\mathbb{R}^d} g(\boldsymbol{u'},\boldsymbol{v'})\mathcal{W}(\boldsymbol{v'})d\boldsymbol{v'}\right]\end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathbb{R}^d} \left( \int_{\mathbb{R}^d} E[\mathcal{W}(\boldsymbol{v})\mathcal{W}(\boldsymbol{v'})] g(\boldsymbol{u}, \boldsymbol{v}) d\boldsymbol{v} \right) g(\boldsymbol{u'}, \boldsymbol{v'}) d\boldsymbol{v'} \\
&= \int_{\mathbb{R}^d} \left( \int_{\mathbb{R}^d} \delta_f(\boldsymbol{v} - \boldsymbol{v'}) g(\boldsymbol{u}, \boldsymbol{v}) d\boldsymbol{v} \right) g(\boldsymbol{u'}, \boldsymbol{v'}) d\boldsymbol{v'} \\
&= \int_{\mathbb{R}^d} g(\boldsymbol{u}, \boldsymbol{v'}) g(\boldsymbol{u'}, \boldsymbol{v'}) d\boldsymbol{v'}.
\end{aligned}
$$

If we define $C(\boldsymbol{u}, \boldsymbol{u'}) := \operatorname{Cov}\Big(x(\boldsymbol{u}), x(\boldsymbol{u'})\Big)$, the previous result implies that if $L = (1 - \ell^2 \Delta)^{\beta/2}$, then for our linear $L$ acting only on $\boldsymbol{u'}$,

$$
\begin{aligned}
LC(\boldsymbol{u}, \boldsymbol{u'}) &= L \int_{\mathbb{R}^d} g(\boldsymbol{u}, \boldsymbol{v'}) g(\boldsymbol{u'}, \boldsymbol{v'}) d\boldsymbol{v'} \\
&= \int_{\mathbb{R}^d} \Big[ Lg(\boldsymbol{u'}, \boldsymbol{v'}) \Big] g(\boldsymbol{u}, \boldsymbol{v'}) d\boldsymbol{v'} \\
&= \int_{\mathbb{R}^d} \delta_f(\boldsymbol{u'} - \boldsymbol{v'}) g(\boldsymbol{u}, \boldsymbol{v'}) d\boldsymbol{v'} \\
&= g(\boldsymbol{u}, \boldsymbol{u'}). \qquad (2.6)
\end{aligned}
$$

To derive the Green's function $g$ in (2.6), we first define $g(\boldsymbol{u}) := g(\boldsymbol{u}, \boldsymbol{0})$. Then (2.4) implies

$$(1 - \ell^2 \Delta)^{\beta/2} g(\boldsymbol{u}) = \delta_f(\boldsymbol{u}). \qquad (2.7)$$

To proceed, we must take the Fourier transform of both sides of (2.7). Using the results obtained in Section 2.1.4, this yields

$$(1 + \ell^2 \|\boldsymbol{\omega}\|^2)^{\beta/2} \widehat{g}(\boldsymbol{\omega}) = 1.$$

Thus, the Fourier transform of the Green's function is

$$\widehat{g}(\boldsymbol{\omega}) = (1 + \ell^2 \|\boldsymbol{\omega}\|^2)^{-\beta/2}. \qquad (2.8)$$

Next, we assume *stationarity* so that the covariance only depends on the relative locations of the points, i.e., $\boldsymbol{r} := \boldsymbol{u} - \boldsymbol{v}$. Then $E[x(\boldsymbol{u})x(\boldsymbol{v})] = E[x(\boldsymbol{r})x(\boldsymbol{0})] = C(\boldsymbol{r}, \boldsymbol{0}) := C(\boldsymbol{r})$ and (2.6) can be expressed $LC(\boldsymbol{r}) = g(\boldsymbol{r})$. If we take the Fourier transform of both sides of this equation, and appeal to (2.8), we obtain

$$\begin{aligned}
&\mathcal{F}\{(1-\ell^2\Delta)^{\beta/2}C(\boldsymbol{r})\}(\boldsymbol{\omega}) = \mathcal{F}\{g(\boldsymbol{r})\}(\boldsymbol{\omega}) \\
\implies\ &(1+\ell^2\|\boldsymbol{\omega}\|^2)^{\beta/2}\hat{C}(\boldsymbol{\omega}) = (1+\ell^2\|\boldsymbol{\omega}\|^2)^{-\beta/2}.
\end{aligned}$$

Thus

$$\hat{C}(\boldsymbol{\omega}) = (1+\ell^2\|\boldsymbol{\omega}\|^2)^{-\beta},$$

and we can write

$$\begin{aligned}
C(\boldsymbol{r}) &= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\hat{C}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}\cdot\boldsymbol{r}}d\boldsymbol{\omega} \\
&= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}(1+\ell^2\|\boldsymbol{\omega}\|^2)^{-\beta}e^{i\boldsymbol{\omega}\cdot\boldsymbol{r}}d\boldsymbol{\omega}.
\end{aligned}$$

For $d = 1$, we have

$$\begin{aligned}
C(r) &= \frac{1}{2\pi}\int_{\mathbb{R}}(1+\ell^2\omega^2)^{-\beta}e^{i\omega r}d\omega \\
&= \frac{1}{2\pi}\int_{\mathbb{R}}(1+\ell^2\omega^2)^{-\beta}\left(\cos(\omega r) + i\sin(\omega r)\right)d\omega \\
&= \frac{1}{2\pi}\int_{\mathbb{R}}(1+\ell^2\omega^2)^{-\beta}\cos(\omega r)d\omega + \frac{i}{2\pi}\int_{\mathbb{R}}(1+\ell^2\omega^2)^{-\beta}\sin(\omega r)d\omega.
\end{aligned}$$

Notice now that $(1+\ell^2\omega^2)^{-\beta}$ is an even function of $\omega$ and $\sin(\omega r)$ is an odd function. Thus, $\int_{\mathbb{R}}(1+\ell^2\omega^2)^{-\beta}\sin(\omega r)d\omega = 0$. Therefore,

$$C(r) = \frac{1}{2\pi}\int_{\mathbb{R}}(1+\ell^2\omega^2)^{-\beta}\cos(\omega r)d\omega$$

$$
\begin{aligned}
&= \frac{2}{2\pi}\int_0^\infty (1+\ell^2\omega^2)^{-\beta}\cos(\omega r)d\omega \quad \text{(since the integrand is even)}\\
&= \frac{1}{\pi}\sqrt{\frac{\pi}{2}}\int_0^\infty (1+\ell^2\omega^2)^{-\beta}\sqrt{\frac{2}{\pi}}\frac{\cos(\omega r)}{\sqrt{\omega r}}\sqrt{\omega r}d\omega\\
&= \frac{1}{\sqrt{2\pi}}\int_0^\infty (1+\ell^2\omega^2)^{-\beta}J_{-1/2}(\omega r)\sqrt{\omega r}d\omega\\
&= \frac{1}{(2\pi)^{\frac{d}{2}}r^{\frac{d-1}{2}}}\int_0^\infty \omega^{\frac{d-1}{2}}(1+\ell^2\omega^2)^{-\beta}J_{\frac{d-2}{2}}(\omega r)(\omega r)^{1/2}d\omega,
\end{aligned}
$$

where $J_\nu(\cdot)$ is the Bessel function of the first kind of order $\nu$ [Andrews, 1992]. It is known that an expression for $J_{-1/2}(t)$ is $\sqrt{\frac{2}{\pi}}\frac{\cos(t)}{\sqrt{t}}$.

For $d = 2$, we have

$$
C(\boldsymbol{r}) = \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2}(1+\ell^2\|\boldsymbol{\omega}\|^2)^{-\beta}e^{i\boldsymbol{\omega}\cdot\boldsymbol{r}}d\boldsymbol{\omega}.
$$

Since the Laplacian, $\Delta$, is invariant under rotations and translations and we are assuming isotropy in the covariance, we have radial symmetry. Thus, we can let $\omega = \|\boldsymbol{\omega}\| = \sqrt{\omega_1^2+\omega_2^2}$ and $r = \|\boldsymbol{r}\| = \sqrt{r_1^2+r_2^2}$. We will also use the fact that $\boldsymbol{\omega}\cdot\boldsymbol{r} = \|\boldsymbol{\omega}\|\|\boldsymbol{r}\|\cos(\theta-\phi) = \omega r\cos(\theta-\phi)$ for $(r_1,r_2) = (r,\phi)$ and $(\omega_1,\omega_2) = (\omega,\theta)$. Then

$$
\begin{aligned}
&\frac{1}{(2\pi)^2}\int_{\mathbb{R}^2}(1+\ell^2\|\boldsymbol{\omega}\|^2)^{-\beta}e^{i\boldsymbol{\omega}\cdot\boldsymbol{r}}d\boldsymbol{\omega}\\
&= \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^\infty (1+\ell^2\omega^2)^{-\beta}e^{i\omega r\cos(\theta-\phi)}\omega d\omega d\theta\\
&= \frac{1}{(2\pi)^2}\int_0^\infty (1+\ell^2\omega^2)^{-\beta}\left(\int_0^{2\pi} e^{i\omega r\cos(\theta-\phi)}d\theta\right)\omega d\omega.
\end{aligned}
$$

Additionally, since $e^{i\omega r\cos(\theta-\phi)}$ has period $2\pi$, $\int_0^{2\pi} e^{i\omega r\cos(\theta-\phi)}d\theta = \int_0^{2\pi} e^{i\omega r\cos(\theta)}d\theta$, which is a representation of $2\pi J_0(\omega r)$, where $J_0(\cdot)$ is the Bessel function of the first kind of order 0. Therefore,

$$\begin{aligned}
C(r) &= \frac{1}{(2\pi)^2}\int_0^\infty (1+\ell^2\omega^2)^{-\beta} 2\pi J_0(\omega r)\omega d\omega \\
&= \frac{1}{2\pi}\int_0^\infty (1+\ell^2\omega^2)^{-\beta} J_0(\omega r)\omega d\omega \\
&= \frac{1}{(2\pi)^{\frac{d}{2}} r^{\frac{d-1}{2}}}\int_0^\infty \omega^{\frac{d-1}{2}} (1+\ell^2\omega^2)^{-\beta} J_{\frac{d-2}{2}}(\omega r)(\omega r)^{1/2} d\omega \\
&= \frac{1}{(2\pi)^{\frac{d}{2}} r^{\frac{d-1}{2}} \ell^{2\beta}}\int_0^\infty \omega^{\frac{d-1}{2}} (\ell^{-2}+\omega^2)^{-\beta} J_{\frac{d-2}{2}}(\omega r)(\omega r)^{1/2} d\omega.
\end{aligned}$$

This is very similar to the Hankel transform [Piessens, 2000]. Therefore, when $d = 1$ or $2$ we have a generalized representation of the covariance. Although not shown here, the Hankel transform can be used to show this representation of the covariance holds for $d > 2$ as well when assuming stationarity and isotropy in the field.

Using the integral identity [Bateman, 1954, Eq. 20, p. 24, vol. II] with $x = \omega, y = r, \alpha = \ell^{-1}, \nu = \frac{d-2}{2}$, and $\mu = \beta - 1$, we obtain

$$C(r) = \frac{\ell^{\beta-\frac{d}{2}} r^{\beta-\frac{d}{2}} K_{\frac{d}{2}-\beta}(r/\ell)}{\ell^{2\beta}(2\pi)^{\frac{d}{2}} 2^{\beta-1}\Gamma(\beta)}$$

where $K_\nu(\cdot)$ is the modified Bessel function of the second kind of order $\nu$ and $\Gamma(\cdot)$ is the gamma function. Now let $\nu = \beta - \frac{d}{2}$. This integral converges when $\ell, d, \nu > 0$. Additionally, since $K_\nu = K_{-\nu}$, we have,

$$\begin{aligned}
C(r) &= \frac{\ell^\nu r^\nu K_{-\nu}(r/\ell)}{\ell^{2\nu+d}\pi^{\frac{d}{2}} 2^{\nu+d-1}\Gamma(\nu+\frac{d}{2})} = \frac{(r/\ell)^\nu K_\nu(r/\ell)}{\ell^d \pi^{\frac{d}{2}} 2^{\nu+d-1}\Gamma(\nu+\frac{d}{2})} \\
&= \frac{\Gamma(\nu)}{\ell^d\Gamma(\nu+\frac{d}{2})}\frac{(r/\ell)^\nu K_\nu(r/\ell)}{\pi^{\frac{d}{2}} 2^{\nu-1} 2^d \Gamma(\nu)} = \frac{\Gamma(\nu)}{\ell^d (4\pi)^{\frac{d}{2}}\Gamma(\nu+\frac{d}{2})}\frac{(r/\ell)^\nu K_\nu(r/\ell)}{2^{\nu-1}\Gamma(\nu)}.
\end{aligned}$$

Letting $\sigma^2 = \frac{\Gamma(\nu)}{\ell^d (4\pi)^{\frac{d}{2}} \Gamma(\nu + \frac{d}{2})}$, $C(r)$ can be equivalently expressed

$$C(r) = \sigma^2 \frac{(r/\ell)^\nu K_\nu(r/\ell)}{2^{\nu-1}\Gamma(\nu)}, \tag{2.9}$$

which is exactly the Matérn covariance function (1.6). □

## 2.3 Anisotropic Case

The solution to (2.1) is an isotropic Gaussian field, which means the correlation length is the same in every direction. This isotropy assumption is often not satisfied and so it will be useful to have an alternate SPDE formulation for the case when correlation lengths differ with direction. Specifically, we will focus on modeling *geometric anisotropy*, which means that the correlation length changes with direction in a smooth fashion between an axis of maximum correlation and an axis of minimum correlation [Cressie, 2015]. There are other types of anisotropy, e.g. zonal anisotropy, but in this dissertation we will only consider geometric anisotropy. Hence, any mention of anisotropy going forward will refer to geometric anisotropy. The groundwork for constructing priors that can model anisotropy has been laid in works such as [Roininen et al., 2014, Lindgren et al., 2011, Hale, 2013]. An example of an anisotropic field is given in Figure 2.2 where it is clear that the correlation persists the longest in the 45° direction.

### 2.3.1 Anisotropic SPDE

We will derive an anisotropic SPDE based on the isotropic SPDE (2.1). We will only consider the two-dimensional case, but results can be extended to

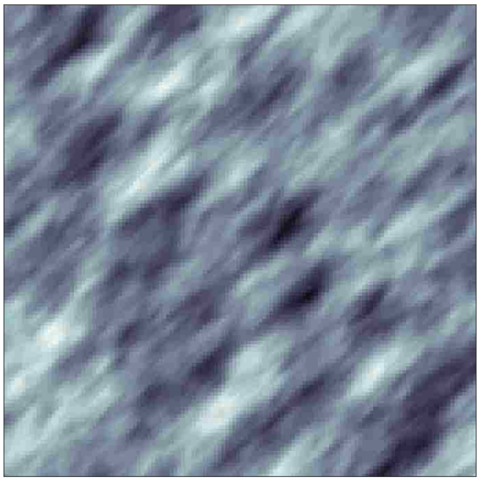

Figure 2.2: Anisotropic field. The correlation length differs depending on direction. The correlation length is largest in the 45° direction and shortest in the $-45°$ direction.

$d > 1$ dimensions. In two dimensions, for a Gaussian field with correlation length $\ell_1$ in the direction of the angle $\theta$, where $-\pi/2 < \theta \leq \pi/2$ is measured counter-clockwise from the $x$-axis, and correlation length $\ell_2$ in the direction perpendicular to $\theta$, we can make the following change of variables from isotropic to anisotropic coordinates:

$$\boldsymbol{w} = \begin{bmatrix} \cos\theta & -\ell_2/\ell_1 \sin\theta \\ \sin\theta & \ell_2/\ell_1 \cos\theta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.10}$$

and thus

$$\begin{aligned} w_1(u_1, u_2) &= \cos\theta u_1 - \ell_2/\ell_1 \sin\theta u_2 \\ w_2(u_1, u_2) &= \sin\theta u_1 + \ell_2/\ell_1 \cos\theta u_2. \end{aligned}$$

We will apply the change of variables (2.10) to both sides of (2.1) to obtain the analogous anisotropic SPDE. The Laplacian on the left-hand side can be altered using the chain rule:

$$\begin{aligned}\frac{\partial}{\partial u_1} &= \frac{\partial}{\partial w_1}\frac{\partial w_1}{\partial u_1} + \frac{\partial}{\partial w_2}\frac{\partial w_2}{\partial u_1} \quad \text{and} \\ \frac{\partial}{\partial u_2} &= \frac{\partial}{\partial w_1}\frac{\partial w_1}{\partial u_2} + \frac{\partial}{\partial w_2}\frac{\partial w_2}{\partial u_2},\end{aligned}$$

which means

$$\begin{aligned}\frac{\partial^2}{\partial u_1^2} &= \left(\frac{\partial^2}{\partial w_1^2}\frac{\partial w_1}{\partial u_1} + \frac{\partial^2}{\partial w_1 \partial w_2}\frac{\partial w_2}{\partial u_1}\right)\frac{\partial w_1}{\partial u_1} + \left(\frac{\partial^2}{\partial w_1 \partial w_2}\frac{\partial w_1}{\partial u_1} + \frac{\partial^2}{\partial w_2^2}\frac{\partial w_2}{\partial u_1}\right)\frac{\partial w_2}{\partial u_1} \\ &= \cos^2\theta \frac{\partial^2}{\partial w_1^2} + 2\sin\theta\cos\theta \frac{\partial^2}{\partial w_1 \partial w_2} + \sin^2\theta \frac{\partial^2}{\partial w_2^2}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial^2}{\partial u_2^2} &= \left(\frac{\partial^2}{\partial w_1^2}\frac{\partial w_1}{\partial u_2} + \frac{\partial^2}{\partial w_2 \partial w_1}\frac{\partial w_2}{\partial u_2}\right)\frac{\partial w_1}{\partial u_2} + \left(\frac{\partial^2}{\partial w_1 \partial w_2}\frac{\partial w_1}{\partial u_2} + \frac{\partial^2}{\partial w_2^2}\frac{\partial w_2}{\partial u_2}\right)\frac{\partial w_2}{\partial u_2} \\ &= (\ell_2/\ell_1)^2 \sin^2\theta \frac{\partial^2}{\partial w_1^2} - 2(\ell_2/\ell_1)^2 \sin\theta\cos\theta \frac{\partial^2}{\partial w_1 \partial w_2} + (\ell_2/\ell_1)^2 \cos^2\theta \frac{\partial^2}{\partial w_2^2}.\end{aligned}$$

Thus,

$$\begin{aligned}\Delta &= \frac{\partial^2}{\partial u_1^2} + \frac{\partial^2}{\partial u_2^2} \\ &= \Big((\ell_2/\ell_1)^2 \sin^2\theta + \cos^2\theta\Big)\frac{\partial^2}{\partial w_1^2} + \Big((\ell_2/\ell_1)^2 \cos^2\theta + \sin^2\theta\Big)\frac{\partial^2}{\partial w_2^2} \\ &\qquad - 2\Big((\ell_2/\ell_1)^2 \sin\theta\cos\theta - \sin\theta\cos\theta\Big)\frac{\partial^2}{\partial w_1 \partial w_2}.\end{aligned}$$

The right hand side of (2.1) is updated by changing the coordinates of the white noise. The inverse transformation of (2.10) is

$$\boldsymbol{u} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\tau\sin\theta & \tau\cos\theta \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} := f(\boldsymbol{w}), \tag{2.11}$$

where $\tau = \ell_1/\ell_2$. Now, we define the transformed white noise basis functions as

$$\tilde{\phi}_j(\boldsymbol{w}) = \psi_j(f(\boldsymbol{w}))|\det(J_f(\boldsymbol{w}))|^{1/2} = \psi_j(f(\boldsymbol{w}))(\ell_1/\ell_2)^{1/2},$$

where $\det(J_f(\boldsymbol{w}))$ denotes the determinant of the Jacobian of the transformation $f(\boldsymbol{w})$, which is $(\ell_1/\ell_2)^{1/2}$ in our case. This will preserve the orthonormal properties of the basis functions. Then, appealing to (2.2),

$$\begin{aligned} \mathcal{W}(\boldsymbol{w}) &= \sum_{j=1}^{\infty} \xi_j \tilde{\phi}_j(\boldsymbol{w}), \quad \xi_j \overset{iid}{\sim} \mathcal{N}(0, \eta^2) \\ &= \sum_{j=1}^{\infty} \xi_j \phi_j(\boldsymbol{u})(\ell_1/\ell_2)^{1/2} = (\ell_1/\ell_2)^{1/2}\mathcal{W}(\boldsymbol{u}), \end{aligned}$$

which means $\mathcal{W}(\boldsymbol{u}) = (\ell_2/\ell_1)^{1/2}\mathcal{W}(\boldsymbol{w})$.

So, taking $\ell = \ell_1$ and making the appropriate substitutions, (2.1) is converted to the anisotropic SPDE:

$$\begin{aligned} \Bigg(1 - \Bigg[(a_\theta^2 + b_\theta^2)\frac{\partial^2}{\partial w_1^2} + (c_\theta^2 + d_\theta^2)\frac{\partial^2}{\partial w_2^2} \\ - 2(a_\theta c_\theta - b_\theta d_\theta)\frac{\partial^2}{\partial w_1 \partial w_2}\Bigg]\Bigg)^{\beta/2} x(\boldsymbol{w}) = (\ell_2/\ell_1)^{1/2}\mathcal{W}(\boldsymbol{w}) \end{aligned}$$

where $a_\theta = \ell_2 \sin\theta$, $b_\theta = \ell_1 \cos\theta$, $c_\theta = \ell_2 \cos\theta$, and $d_\theta = \ell_1 \sin\theta$. For

$$\mathbf{R} = \begin{bmatrix} \ell_1 \cos\theta & \ell_1 \sin\theta \\ -\ell_2 \sin\theta & \ell_2 \cos\theta \end{bmatrix},$$

the above SPDE can be written

$$\left(1 - \nabla \cdot \mathbf{R}^T \mathbf{R} \nabla\right)^{\beta/2} x(\boldsymbol{w}) = (\ell_2/\ell_1)^{1/2} \mathcal{W}(\boldsymbol{w}). \tag{2.12}$$

Notice that if $\ell_1 = \ell_2$, this SPDE is equivalent to (2.1) with $\ell = \ell_1$.

### 2.3.2 Anisotropic Matérn Connection

Like in the isotropic case, we are interested in the properties of the solution of (2.12), especially its covariance function. First, we define the anisotropic Matérn covariance function [Haskard, 2007] as

$$C(r_w) = \sigma^2 \frac{(r_w/\zeta)^\nu K_\nu(r_w/\zeta)}{2^{\nu-1}\Gamma(\nu)}, \tag{2.13}$$

with

$$\zeta = \frac{\ell_1}{\sqrt{\cos^2(\psi - \theta) + (\ell_1/\ell_2)^2 \sin^2(\psi - \theta)}},$$

where $r_w = \|\boldsymbol{w}_i - \boldsymbol{w}_j\|$ is the distance between the anisotropic coordinates, $\zeta$ is the new range parameter in the direction of $\psi$, $\ell_1$ is the correlation length in the direction of $\theta$ and $\ell_2$ is the correlation length in the direction perpendicular to $\theta$. Notice that the smoothness parameter, $\nu$, is unaffected.

The remainder of this subsection contains results used to prove the following theorem, which, to our knowledge, is novel.

**Theorem 2.3.1.** *The solution $x(\boldsymbol{w})$ of (2.12) is a Gaussian field with mean zero and anisotropic Matérn covariance function defined by (2.13).*

*Proof.* First, we derive the Green's function for (2.12), which is the solution of

$$\left(1 - \nabla \cdot \mathbf{R}^T\mathbf{R}\nabla\right)^{\beta/2} g(\boldsymbol{w}, \boldsymbol{v}) = \delta_f(\boldsymbol{v} - \boldsymbol{w}). \tag{2.14}$$

Using (2.3), the solution to (2.12) is given by

$$x(\boldsymbol{w}) = (\ell_2/\ell_1)^{1/2} \int_{\mathbb{R}^2} g(\boldsymbol{w}, \boldsymbol{v})\mathcal{W}(\boldsymbol{v})d\boldsymbol{v}, \tag{2.15}$$

which makes $x(\boldsymbol{w})$ a Gaussian field since it is a linear transformation of Gaussian white noise. Be aware that we are still assuming stationarity in our field. To derive the Green's function $g$ in (2.15), we first define $g(\boldsymbol{w}) := g(\boldsymbol{w}, \mathbf{0})$. Then (2.14) implies

$$\left(1 - \nabla \cdot \mathbf{R}^T\mathbf{R}\nabla\right)^{\beta/2} g(\boldsymbol{w}) = \delta_f(\boldsymbol{w}). \tag{2.16}$$

We would like to change from the anisotropic coordinates $\boldsymbol{w}$ to anisotropic coordinates $\boldsymbol{u}$ in (2.16) so we can use the results from Section 2.2. We again use (2.11) for the coordinate change and, in a similar fashion as was done earlier, we apply the chain rule to replace $\partial^2/\partial w_1^2$, $\partial^2/\partial w_2^2$, and $\partial^2/(\partial w_1 \partial w_2)$ in $\left(1 - \nabla \cdot \mathbf{R}^T\mathbf{R}\nabla\right)^{\beta/2}$ with partial derivatives in terms of $\boldsymbol{u}$. When making this change, the coefficients of $\partial^2/\partial u_1^2$, $\partial^2/\partial u_2^2$, and $\partial^2/(\partial u_1 \partial u_2)$ are $\ell_1^2$, $\ell_1^2$, and 0, respectively and so we have $(1 - \nabla \cdot \mathbf{R}^T\mathbf{R}\nabla)g(\boldsymbol{w}) = (1 - \ell_1^2\Delta)g(\boldsymbol{u})$. Additionally, we can change variables in the Delta function on the right side of (2.16) by multiplying by the determinant of the Jacobian of (2.11): $\ell_1/\ell_2$.

Thus, the change of variables transforms (2.16) into the equation

$$\left(1-\ell_1^2\Delta\right)^{\beta/2} g(\boldsymbol{u}) = (\ell_1/\ell_2)\delta_f(\boldsymbol{u}), \tag{2.17}$$

which is equivalent to (2.7) up to a constant. Hence, we can apply the results of Section 2.2. Namely, after changing variables, the solution of (2.12) is a Gaussian field with mean zero and the isotropic Matérn covariance function defined by (1.6). Notice that the constant that multiplies the Delta function on the right-hand side of (2.17) and the constant that multiplies the integral in (2.15) will cancel when going through the process of deriving the covariance function since the constant in (2.15) gets squared.

We must now make one final change of variables back to $\boldsymbol{w}$ from $\boldsymbol{u}$ so our covariance function will be in terms of the anisotropic coordinates rather than the isotropic ones. Since the input to the Matérn correlation function must be a distance between isotropic spatial locations, we need to represent an isotropic distance, $r_u$, in terms of the anisotropic coordinates. Consider $\boldsymbol{r} := \boldsymbol{w}_i - \boldsymbol{w}_j$. Then, defining $r_w := \|\boldsymbol{r}\| = \|\boldsymbol{w}_i - \boldsymbol{w}_j\|$,

$$\begin{aligned} r_u := \|\boldsymbol{u}_i - \boldsymbol{u}_j\| &= \left\| \begin{bmatrix} \cos\theta & \sin\theta \\ -\tau\sin\theta & \tau\cos\theta \end{bmatrix} (\boldsymbol{w}_i - \boldsymbol{w}_j) \right\| = \left\| \begin{bmatrix} \cos\theta & \sin\theta \\ -\tau\sin\theta & \tau\cos\theta \end{bmatrix} \boldsymbol{r} \right\| \\ &= \left\| \begin{bmatrix} \cos\theta & \sin\theta \\ -\tau\sin\theta & \tau\cos\theta \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \right\| = \left\| \begin{bmatrix} r_1\cos\theta + r_2\sin\theta \\ -r_1\tau\sin\theta + r_2\tau\cos\theta \end{bmatrix} \right\|. \end{aligned}$$

Now we convert to polar coordinates with $r_1 = r_w\cos\psi$ and $r_2 = r_w\sin\psi$.

Then

$$\begin{aligned} r_u &= \left\| \begin{bmatrix} r_w \cos\psi \cos\theta + r_w \sin\psi \sin\theta \\ -r_w \tau \cos\psi \sin\theta + r_w \tau \sin\psi \cos\theta \end{bmatrix} \right\| = \left\| \begin{bmatrix} r_w \cos(\psi - \theta) \\ r_w \tau \sin(\psi - \theta) \end{bmatrix} \right\| \\ &= r_w \left[ \cos^2(\psi - \theta) + \tau^2 \sin^2(\psi - \theta) \right]^{1/2}. \end{aligned}$$

Therefore, we need to adjust the distance between the vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ by $[\cos^2(\psi - \theta) + \tau^2 \sin^2(\psi - \theta)]^{1/2}$ in order to get the distances to plug into the isotropic Matérn correlation function. Thus, the isotropic Matérn covariance function has been generalized to the anisotropic case using the same change of variables as in (2.10). Adjusting the anisotropic distances is equivalent to defining the anisotropic Matérn covariance function as we have in (2.13). □

# Chapter 3

# Numerical Methods

In this chapter, we will go into detail about how to discretize the SPDEs from Chapter 2 to form prior precision matrices that can be used to solve inverse problems. We also discuss the effect restricting to a finite domain has on the covariance of the solutions to the SPDEs and what must be done in order to combat this potential issue.

## 3.1 Preliminaries

Before proceeding, we need some additional preliminary explanations of some numerical techniques we use in this chapter and beyond.

### 3.1.1 Kronecker Product

The Kronecker product for matrices $\mathbf{C} \in \mathbb{R}^{m\times n}$ and $\mathbf{D} \in \mathbb{R}^{r\times s}$, is defined by

$$\mathbf{C} \otimes \mathbf{D} = \begin{bmatrix} c_{11}\mathbf{D} & c_{12}\mathbf{D} & \dots & c_{1n}\mathbf{D} \\ c_{21}\mathbf{D} & c_{22}\mathbf{D} & \dots & c_{2n}\mathbf{D} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1}\mathbf{D} & c_{m2}\mathbf{D} & \dots & c_{mn}\mathbf{D} \end{bmatrix}_{mr\times ns}$$

with $c_{ij} = [\mathbf{C}]_{ij}$ [Van Loan, 2000]. For example, for the $2 \times 2$ matrix of all ones, $\mathbf{J}_2$,

$$\mathbf{J}_2 \otimes \mathbf{I}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{I}_2 \otimes \mathbf{J}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

### 3.1.2 vec(·) and diag(·) Functions

For an $n \times n$ matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix},$$

the function $\text{vec}(\mathbf{A})$ converts the matrix $\mathbf{A}$ to a vector by stacking its columns as follows: $\text{vec}(\mathbf{A}) = (a_{11}, a_{21}, \dots, a_{n1}, a_{12}, \dots, a_{1n}, \dots, a_{nn})^T$. A useful property related to the $\text{vec}(\cdot)$ function and Kronecker products is [Bardsley, 2018]:

$$\text{vec}(\mathbf{D}\mathbf{X}\mathbf{C}^T) = (\mathbf{C} \otimes \mathbf{D})\text{vec}(\mathbf{X}). \tag{3.1}$$

The diag$(\cdot)$ function takes a vector of length $n$ and creates a diagonal $n \times n$ matrix with the diagonal elements given by the elements of the vector. For example, if we have a vector $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^T$, then

$$\text{diag}(\boldsymbol{x}) = \begin{bmatrix} x_1 & 0 & \dots & 0 \\ 0 & x_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x_n \end{bmatrix}.$$

### 3.1.3 Discrete Fourier Transform

If we are working on a regular grid in one dimension and periodic boundary conditions are assumed, then the forward map $\mathbf{A}$ in (1.1) and the precision matrix $\mathbf{P}$ are both circulant. An $n \times n$ circulant matrix has the form

$$\mathbf{A} = \begin{bmatrix} a_0 & a_{n-1} & \dots & a_2 & a_1 \\ a_1 & a_0 & a_{n-1} & & a_2 \\ \vdots & a_1 & a_0 & \ddots & \vdots \\ a_{n-2} & & \ddots & \ddots & a_{n-1} \\ a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \end{bmatrix},$$

so that the entire matrix is defined by its first column, which we will denote as $\boldsymbol{a}_s$. Notice that circulant matrices are also Toeplitz since their diagonals are constant.

The advantage of working with circulant matrices is in the fact that they can be diagonalized using the discrete Fourier transform (DFT) [Bardsley, 2018]. The diagonalization using the DFT can be accomplished in the following way. First, the DFT matrix has the form

$$[\mathbf{F}]_{ij} = \frac{1}{\sqrt{n}} \omega^{(i-1)(j-1)/n}, \quad 1 \leq i, j \leq n,$$

where $\omega = \exp(2\pi\sqrt{-1}/n)$. Some properties of the DFT matrix $\mathbf{F}$ include the fact that $\mathbf{F}$ is symmetric, i.e. $\mathbf{F}^T = \mathbf{F}$, and the inverse DFT matrix $\mathbf{F}^{-1} = \mathbf{F}^*$, where '$*$' denotes the conjugate transpose. It can be shown that

$$\mathbf{A} = \mathbf{F}^* \mathrm{diag}(\sqrt{n}\mathbf{F}\boldsymbol{a}_s)\mathbf{F}$$

and the eigenvalues of $\mathbf{A}$ are equal to $\widehat{\boldsymbol{a}}_s := \sqrt{n}\mathbf{F}\boldsymbol{a}_s$. Therefore, we can compute matrix inverses and multiplication quickly by taking $\mathbf{A}^{-1} = \mathbf{F}^*\mathrm{diag}(1/\widehat{\boldsymbol{a}}_s)\mathbf{F}$, where the division is component-wise, and an expression like $\left(\mathbf{A}^T\mathbf{A} + \alpha\mathbf{P}\right)^{-1}\mathbf{A}^T\boldsymbol{b}$ can be computed using

$$\left(\mathbf{A}^T\mathbf{A} + \alpha\mathbf{P}\right)^{-1}\mathbf{A}^T\boldsymbol{b} = \mathbf{F}^*\mathrm{diag}\left(\frac{\mathrm{conj}(\widehat{\boldsymbol{a}}_s)}{|\widehat{\boldsymbol{a}}_s|^2 + \alpha\widehat{\boldsymbol{p}}_s}\right)\mathbf{F}\boldsymbol{b},$$

where $\mathrm{conj}(\cdot)$ denotes the component-wise complex conjugate, $|\cdot|^2$ denotes the component-wise squared-modulus, $\widehat{\boldsymbol{p}}_s$ are the eigenvalues of $\mathbf{P}$ (assuming $\mathbf{P}$ is circulant), and division is again component-wise.

### Two-Dimensional DFT

In two dimensions, we lose the overall circulant structure of the matrix $\mathbf{A}$. The matrix instead becomes block circulant with circulant blocks (BCCB), which has the following form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_{n-1} & \dots & \mathbf{A}_2 & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{A}_{n-1} & & \mathbf{A}_2 \\ \vdots & \mathbf{A}_1 & \mathbf{A}_0 & \ddots & \vdots \\ \mathbf{A}_{n-2} & & \ddots & \ddots & \mathbf{A}_{n-1} \\ \mathbf{A}_{n-1} & \mathbf{A}_{n-2} & \dots & \mathbf{A}_1 & \mathbf{A}_0 \end{bmatrix}.$$

Here each $\mathbf{A}_i$ for $0 \leq i \leq n-1$ is circulant and of size $n \times n$, which means $\mathbf{A}$ is $n^2 \times n^2$. We can obtain $\boldsymbol{a}_s$ in this case by assigning the first column of $\mathbf{A}_{j-1}$ as the $j$th column of $\boldsymbol{a}_s$ for $j = 1, 2, \dots, n$. Thus, in the two-dimensional case, $\boldsymbol{a}_s$ is an $n \times n$ array instead of an $n \times 1$ vector. Additionally, the DFT matrix in two dimensions, which we denoted by $\mathbf{F}$ in one dimension, is defined as $\mathbf{F}_{\text{2D}} = \mathbf{F} \otimes \mathbf{F}$ (and so $\mathbf{F}^*_{\text{2D}} = \mathbf{F}^* \otimes \mathbf{F}^*$), where $\otimes$ denotes the Kronecker product.

In two dimensions, the eigenvalues of the BCCB matrix $\mathbf{A}$ are $\widehat{\boldsymbol{a}}_s = n\mathbf{F}\boldsymbol{a}_s\mathbf{F}$ and the diagonalization of $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{F}^*_{\text{2D}} \text{diag}\left(\text{vec}(\widehat{\boldsymbol{a}}_s)\right) \mathbf{F}_{\text{2D}}.$$

This again can lead to quick computations. In practice, these computations are typically done on arrays instead of vectors. For example, if $\boldsymbol{x} = \text{vec}(\mathbf{X})$ and $\boldsymbol{b} = \text{vec}(\mathbf{B})$, then $\mathbf{A}\boldsymbol{x} = \text{vec}\Big(\mathbf{F}^*(\widehat{\boldsymbol{a}}_s \odot [\mathbf{FXF}])\mathbf{F}^*\Big)$, where $\odot$ denotes component-wise multiplication, and

$$\left(\mathbf{A}^T\mathbf{A} + \alpha\mathbf{P}\right)^{-1} \mathbf{A}^T\boldsymbol{b} = \text{vec}\left(\mathbf{F}^* \left(\frac{\text{conj}(\widehat{\boldsymbol{a}}_s)}{|\widehat{\boldsymbol{a}}_s|^2 + \alpha\widehat{\boldsymbol{p}}_s} \odot [\mathbf{FBF}]\right) \mathbf{F}^*\right),$$

where $\widehat{\boldsymbol{p}}_s$ are the eigenvalues of the BCCB matrix $\mathbf{P}$. Notice that we can use (3.1) to write $\mathbf{F}_{2\mathrm{D}}\boldsymbol{x} = (\mathbf{F} \otimes \mathbf{F})\boldsymbol{x} = \mathrm{vec}(\mathbf{FXF})$ and $\mathbf{F}_{2\mathrm{D}}^{*}\boldsymbol{x} = \mathrm{vec}(\mathbf{F}^{*}\mathbf{XF}^{*})$. Therefore, when performing computations on arrays, $\mathbf{FXF}$ and $\mathbf{F}^{*}\mathbf{XF}^{*}$ compute the two-dimensional DFT and inverse DFT, respectively.

To more efficiently perform the multiplication by $\mathbf{F}$ and $\mathbf{F}^{*}$, many modern computer programs, such as MATLAB, use an algorithm known as the fast Fourier transform (FFT), which computes $\mathbf{F}\boldsymbol{x}$ in $\mathcal{O}(n \log n)$ operations. In MATLAB, for an $n \times 1$ vector $\boldsymbol{v}$ and an $n \times n$ matrix $\mathbf{W}$, the following code can be used:

$$\begin{aligned}
\texttt{fft(v)} &= \sqrt{n}\mathbf{F}\boldsymbol{v} \\
\texttt{ifft(v)} &= \frac{1}{\sqrt{n}}\mathbf{F}^{*}\boldsymbol{v} \\
\texttt{fft2(W)} &= n\mathbf{FWF} \\
\texttt{ifft2(W)} &= \frac{1}{n}\mathbf{F}^{*}\mathbf{WF}^{*}.
\end{aligned}$$

This is useful since it eliminates the need to create and store $\mathbf{F}$ and $\mathbf{F}^{*}$.

### 3.1.4 Conjugate Gradient

The conjugate gradient (CG) and preconditioned conjugate gradient (PCG) methods are algorithms for finding numerical solutions for systems of linear equations like those in the form

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}, \tag{3.2}$$

where $\mathbf{A}$ and $\boldsymbol{b}$ are known and we are interested in finding the solution $\boldsymbol{x}$. There are some restrictions on $\mathbf{A}$ in that it must be symmetric (i.e. $\mathbf{A}^T = \mathbf{A}$) and positive definite (i.e. $\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} > 0$ for all nonzero $\boldsymbol{x} \in \mathbb{R}^n$). A matrix with both of these properties is known as symmetric positive definite (SPD). If these are satisfied, CG is one of the most useful numerical techniques for solving large linear systems of equations [Nocedal and Wright, 2006].

### The Method

We will first explain the CG method before discussing the PCG method. The key to CG is the fact that solving the expression (3.2) is equivalent to the following minimization problem:

$$\arg\min_{\boldsymbol{x}} J(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x},$$

since both give the same solution, which we will denote $\boldsymbol{x}_*$. Using vector derivatives, the gradient of $J$ is equal to

$$\nabla J(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} - \boldsymbol{b} := \boldsymbol{r}(\boldsymbol{x}),$$

which is also known as the residual of the linear system. The only use of $\boldsymbol{r}$ in this section will refer to the residual as opposed to a vector of distances, as it is elsewhere in the thesis. Since CG is an iterative method, when $\boldsymbol{x} = \boldsymbol{x}_k$, the $k$th iteration of $\boldsymbol{x}$, the $k$th iteration of the residual is given by

$$\boldsymbol{r}_k = \mathbf{A}\boldsymbol{x}_k - \boldsymbol{b}. \tag{3.3}$$

Therefore, the direction of steepest descent of the function $J(\boldsymbol{x})$ at iteration $k$ is $-\nabla J(\boldsymbol{x}) = -\boldsymbol{r}_k$.

Now, the conjugacy in the name of the CG method comes from the fact that it uses the property that two vectors, $\boldsymbol{u}$ and $\boldsymbol{v}$, are called *conjugate* with respect to a SPD matrix $\mathbf{A}$ if $\boldsymbol{u}^T\mathbf{A}\boldsymbol{v} = 0$ for $\boldsymbol{u} \neq \boldsymbol{v}$. Take a set of nonzero, conjugate vectors $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$. Since they are conjugate, it is simple to show they are also linearly independent, and hence, form a basis in $\mathbb{R}^n$. This means the solution can be written as a linear combination of these vectors,

$$\boldsymbol{x}_* = \boldsymbol{x}_0 + \sum_{i=1}^{n} \alpha_i \boldsymbol{p}_i.$$

Therefore, if we let

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \tag{3.4}$$

where the subscript denotes iteration number. We will obtain the solution $\boldsymbol{x}_*$ in at most $n$ iterations since it is certain that $\boldsymbol{x}_n = \boldsymbol{x}_*$. Ideally, we will choose these $\boldsymbol{p}_k$ vectors in such a way that we can approximate $\boldsymbol{x}_*$ in fewer than $n$ iterations, which is necessary for large $n$.

We need an expression for $\alpha_k$ and $\boldsymbol{p}_k$. To determine the $\alpha_k$ values, we will minimize $J(\boldsymbol{x}_{k+1})$ with respect to $\alpha_k$:

$$\begin{aligned} J(\boldsymbol{x}_{k+1}) = J(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) &= \frac{1}{2}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)^T \mathbf{A} (\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) - \boldsymbol{b}^T(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) \\ &= \frac{1}{2}\boldsymbol{x}_k^T \mathbf{A} \boldsymbol{x}_k + \alpha_k \boldsymbol{x}_k^T \mathbf{A} \boldsymbol{p}_k + \frac{1}{2}\alpha_k^2 \boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k - \boldsymbol{b}^T \boldsymbol{x}_k - \alpha_k \boldsymbol{b}^T \boldsymbol{p}_k, \end{aligned}$$

and so

$$\frac{d}{d\alpha_k} J(\boldsymbol{x}_{k+1}) = 0 \implies \boldsymbol{x}_k^T \mathbf{A} \boldsymbol{p}_k + \alpha_k \boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k - \boldsymbol{b}^T \boldsymbol{p}_k = 0.$$

Solving this for $\alpha_k$ gives

$$\begin{aligned}\alpha_k &= \frac{\boldsymbol{b}^T\boldsymbol{p}_k - \boldsymbol{x}_k^T\mathbf{A}\boldsymbol{p}_k}{\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_k} = \frac{(\boldsymbol{b}^T - \boldsymbol{x}_k^T\mathbf{A})\boldsymbol{p}_k}{\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_k} \\ &= -\frac{\boldsymbol{r}_k^T\boldsymbol{p}_k}{\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_k}. \end{aligned} \tag{3.5}$$

Since $\boldsymbol{p}_k$ can be thought of as the step direction, and we showed $-\boldsymbol{r}_k$ is the steepest descent direction, we will update the $\boldsymbol{p}_k$ by using the residuals and the previous $\boldsymbol{p}$ vector:

$$\boldsymbol{p}_{k+1} = -\boldsymbol{r}_{k+1} + \beta_k\boldsymbol{p}_k. \tag{3.6}$$

The choice of $\beta_k$ will be discussed below and $\boldsymbol{r}_{k+1}$ is updated by using (3.3) and (3.4). Specifically,

$$\begin{aligned}\boldsymbol{r}_{k+1} &= \mathbf{A}\boldsymbol{x}_{k+1} - \boldsymbol{b} \quad \text{by (3.3)} \\ &= \mathbf{A}(\boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k) - \boldsymbol{b} \quad \text{by (3.4)} \\ &= \mathbf{A}\boldsymbol{x}_k - \boldsymbol{b} + \alpha_k\mathbf{A}\boldsymbol{p}_k \\ &= \boldsymbol{r}_k + \alpha_k\mathbf{A}\boldsymbol{p}_k. \end{aligned} \tag{3.7}$$

$\beta_k$ must be chosen so that the conjugacy of the $\boldsymbol{p}_i$ vectors with respect to $\mathbf{A}$ is retained, i.e. $\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_{k+1} = 0$. Left-multiplying (3.6) by $\boldsymbol{p}_k^T\mathbf{A}$ yields

$$\begin{aligned}\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_{k+1} &= \boldsymbol{p}_k^T\mathbf{A}(-\boldsymbol{r}_{k+1} + \beta_k\boldsymbol{p}_k) = 0 \\ \implies & -\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{r}_{k+1} + \beta_k\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_k = 0. \end{aligned}$$

Solving this for $\beta_k$ gives us

$$\beta_k = \frac{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{r}_{k+1}}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k}.$$

To slightly increase the efficiency of this algorithm, we will now make a few adjustments. Firstly, it is possible to show that $\boldsymbol{r}_i^T \boldsymbol{r}_j = 0$ for $i \neq j$ and $\boldsymbol{r}_i^T \boldsymbol{p}_j = 0$ for $i > j$. Using this fact and (3.6), we can change the expression for $\alpha_k$ in (3.5) to be

$$\alpha_k = -\frac{\boldsymbol{r}_k^T \boldsymbol{p}_k}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k} = -\frac{\boldsymbol{r}_k^T(-\boldsymbol{r}_k + \beta_k \boldsymbol{p}_{k-1})}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k} = \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k - \beta_k \boldsymbol{r}_k^T \boldsymbol{p}_{k-1}}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k} = \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k}.$$

Now, using the orthogonality of the $\boldsymbol{r}_i$ vectors and the fact in (3.7) that $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \mathbf{A} \boldsymbol{p}_k$ and thus $\mathbf{A}\boldsymbol{p}_k = (\boldsymbol{r}_k - \boldsymbol{r}_{k+1})/\alpha_k$, $\beta_k$ can be written

$$\begin{aligned}\beta_k &= \frac{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{r}_{k+1}}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k} = \frac{(\boldsymbol{r}_k - \boldsymbol{r}_{k+1})^T \boldsymbol{r}_{k+1}/\alpha_k}{\boldsymbol{p}_k^T(\boldsymbol{r}_k - \boldsymbol{r}_{k+1})/\alpha_k} = \frac{\boldsymbol{r}_k^T \boldsymbol{r}_{k+1} - \boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}{(-\boldsymbol{r}_k + \beta_{k-1}\boldsymbol{p}_{k-1})^T(\boldsymbol{r}_k - \boldsymbol{r}_{k+1})} \\ &= \frac{\boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}{\boldsymbol{r}_k \boldsymbol{r}_k}.\end{aligned}$$

Now that we have more economical representations of $\alpha_k$ and $\beta_k$, we can write the CG algorithm in full as Algorithm 1.

**Algorithm 1** Conjugate Gradient Method

```
0. Set $k = 0$, $\boldsymbol{r}_0 = \mathbf{A}\boldsymbol{x}_0 - \boldsymbol{b}$, and $\boldsymbol{p}_0 = -\boldsymbol{r}$;
1. Set $\alpha_k = \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{p}_k^T \mathbf{A} \boldsymbol{p}_k}$;
2. Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$;
3. Set $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k \mathbf{A} \boldsymbol{p}_k$;
4. Set $\beta_k = \frac{\boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T \boldsymbol{r}_k}$;
5. Set $\boldsymbol{p}_{k+1} = -\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$;
6. Return to step 1 and repeat until $\boldsymbol{r}_{k+1}$ is sufficiently small.
```

We have stated that it will take at most $n$ iterations for CG to reach the solution, $\boldsymbol{x}_*$. While that is true, it often reaches an acceptable solution much more quickly. For instance, if $\mathbf{A}$ has $r$ distinct eigenvectors, the CG algorithm will obtain the solution in at most $r$ iterations. Additionally, there are two error bounds that can be useful in assessing convergence [Nocedal and Wright, 2006]. If we denote the eigenvalues of $\mathbf{A}$ as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, then the following two bounds hold:

$$(1) \quad (\boldsymbol{x}_{k+1} - \boldsymbol{x}_*)^T \mathbf{A} (\boldsymbol{x}_{k+1} - \boldsymbol{x}_*) \leq \left( \frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 (\boldsymbol{x}_0 - \boldsymbol{x}_*)^T \mathbf{A} (\boldsymbol{x}_0 - \boldsymbol{x}_*)$$

$$(2) \quad (\boldsymbol{x}_{k+1} - \boldsymbol{x}_*)^T \mathbf{A} (\boldsymbol{x}_{k+1} - \boldsymbol{x}_*) \leq \left( \frac{\sqrt{\lambda_n / \lambda_1} - 1}{\sqrt{\lambda_n / \lambda_1} + 1} \right)^k (\boldsymbol{x}_0 - \boldsymbol{x}_*)^T \mathbf{A} (\boldsymbol{x}_0 - \boldsymbol{x}_*).$$

Therefore, it is clear that the speed of the CG convergence is dependent on the eigenvalues of $\mathbf{A}$.

### Preconditioning

Since the eigenvalues of $\mathbf{A}$ are important in the convergence speed of the CG algorithm, we can increase the convergence rate by using a preconditioner matrix, $\mathbf{G} = \mathbf{E}\mathbf{E}^T$, so that the CG depends on the eigenvalues of $\mathbf{E}^{-T}\mathbf{A}\mathbf{E}^{-1}$ instead of $\mathbf{A}$. Using a preconditioner will yield the preconditioned conjugate gradient (PCG) method. The best preconditioner is one in which $\mathbf{E}^{-T}\mathbf{A}\mathbf{E}^{-1} \approx \mathbf{I}$, the identity matrix, and $\mathbf{G}$ is efficient to invert. Using a preconditioner is equivalent to making the change of variables $\boldsymbol{y} = \mathbf{E}\boldsymbol{x}$ and then solving the linear system

$$(\mathbf{E}^{-T}\mathbf{A}\mathbf{E}^{-1})\boldsymbol{y} = \mathbf{E}^{-T}\boldsymbol{b},$$

**Algorithm 2** Preconditioned Conjugate Gradient Method

```
0. Set $k = 0$, $\boldsymbol{r}_0 = \mathbf{A}\boldsymbol{x}_0 - \boldsymbol{b}$, $\boldsymbol{z}_0 = \mathbf{G}^{-1}\boldsymbol{r}_0$, and $\boldsymbol{p}_0 = -\boldsymbol{z}_0$;
1. Set $\alpha_k = \dfrac{\boldsymbol{r}_k^T\boldsymbol{r}_k}{\boldsymbol{p}_k^T\mathbf{A}\boldsymbol{p}_k}$;
2. Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k$;
3. Set $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k\mathbf{A}\boldsymbol{p}_k$;
4. Set $\boldsymbol{z}_{k+1} = \mathbf{G}^{-1}\boldsymbol{r}_{k+1}$;
5. Set $\beta_k = \dfrac{\boldsymbol{z}_{k+1}^T\boldsymbol{r}_{k+1}}{\boldsymbol{z}_k^T\boldsymbol{r}_k}$;
6. Set $\boldsymbol{p}_{k+1} = -\boldsymbol{z}_{k+1} + \beta_k\boldsymbol{p}_k$;
7. Return to step 1 and repeat until $\boldsymbol{r}_{k+1}$ is sufficiently small.
```

which is equivalent to (3.2). The resulting PCG algorithm is given as Algorithm 2.

Note that $\mathbf{E}$ is not used in the implementation of PCG; $\mathbf{G}$ is the only new matrix needed. Although each iteration of PCG is more costly than each iteration of CG, the total number of iterations needed for PCG is significantly lower with a good choice of a preconditioner. PCG is the method we will use in this thesis to solve for our MAP estimator.

An illustration of the CG and PCG methods is presented in Figure 3.1. The left side shows the way $\boldsymbol{x}_k$ converges to the solution in a small-scale example when using CG. For this example,

$$\mathbf{A} = \begin{bmatrix} 3.5 & -1 \\ -1 & 1.5 \end{bmatrix} \text{ and } \boldsymbol{b} = \begin{bmatrix} 2.75 \\ 2.25 \end{bmatrix}, \text{ which means } \boldsymbol{x}_* = \begin{bmatrix} 1.5 \\ 2.5 \end{bmatrix}.$$

Beginning with $\boldsymbol{x}_0 = (0,0)^T$, the first iteration of CG yields $\boldsymbol{x}_1 = (1.60, 1.30)^T$ and the second iteration $\boldsymbol{x}_2 = \boldsymbol{x}_*$ arrives at the true solution, as it should since $n = 2$. The advantage PCG has over CG is illustrated on the right side of

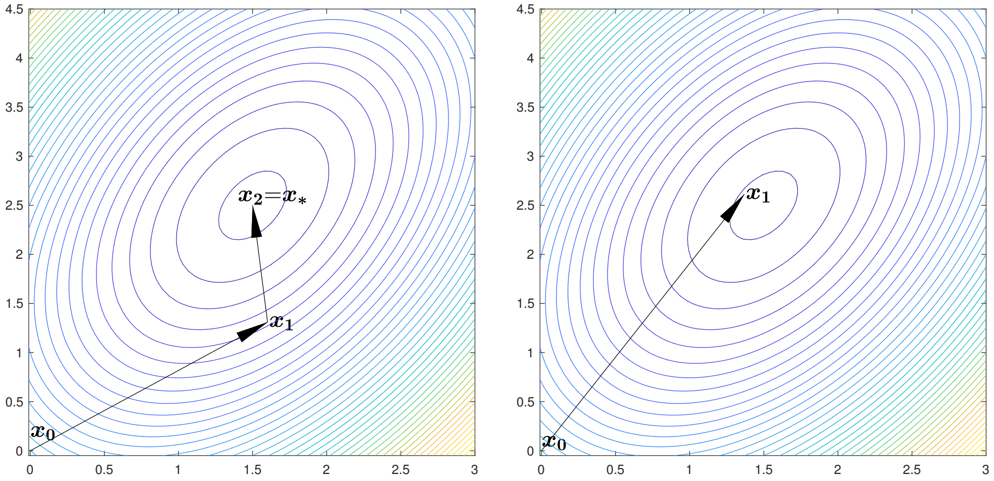

Figure 3.1: Conjugate gradient methods. The image on the left tracks the convergence of the CG algorithm and the image on the right shows PCG. Both methods converged in two iterations.

Figure 3.1. We now use the preconditioner

$$\mathbf{G} = \begin{bmatrix} 3.5 & 0 \\ 0 & 1.5 \end{bmatrix}$$

that has just the diagonal entries of $\mathbf{A}$ and is therefore easier to invert. This time, $\boldsymbol{x}_1 = (1.37, 2.61)^T$, which is substantially closer to $\boldsymbol{x}_*$ than $\boldsymbol{x}_1$ was in the CG case and $\boldsymbol{x}_2 = \boldsymbol{x}_* = (1.5, 2.5)^T$. For this example, both the CG and PCG converged to the true solution in two iterations, but it is clear that PCG has the potential to converge in fewer iterations for a problem with much larger $n$.

## 3.2 Discretizing the SPDE

In order to use the SPDE (2.1) or (2.12) to define the prior precision $\mathbf{P}$, it must be discretized so it is in the form of a matrix. To discretize the SPDEs, we use the finite-difference method [Bardsley, 2018]. To illustrate how the discretization is performed, first consider the two-dimensional case when $\nu = 1$ so that (2.1) reduces to

$$\left(1 - \ell^2 \left[\frac{\partial^2}{\partial u_1^2} + \frac{\partial^2}{\partial u_2^2}\right]\right) x(\boldsymbol{u}) = \mathcal{W}(\boldsymbol{u}), \quad \boldsymbol{u} = (u_1, u_2)^T \in \mathbb{R}^2, \quad \ell > 0.$$

for isotropic fields and (2.12) reduces to

$$\left(1 - \left[(a_\theta^2 + b_\theta^2)\frac{\partial^2}{\partial w_1^2} + (c_\theta^2 + d_\theta^2)\frac{\partial^2}{\partial w_2^2} - 2(a_\theta c_\theta - b_\theta d_\theta)\frac{\partial^2}{\partial w_1 \partial w_2}\right]\right) x(\boldsymbol{w}) = (\ell_2/\ell_1)^{1/2}\mathcal{W}(\boldsymbol{w})$$

for anisotropic ones.

In two dimensions, $x(\boldsymbol{u}) = x(u_1, u_2)$ or $x(\boldsymbol{w}) = x(w_1, w_2)$ is a function of two variables. To ease notation, we let $s$ refer to the first spatial coordinate ($u_1$ in the isotropic case or $w_1$ in the anisotropic case) and $t$ will refer to the second spatial coordinate. Let $h = 1/n$ be the step-size of the discretization, let $s_i = (i - 1/2)h$ for $i = 1, \ldots, n$, and let $t_j = (j - 1/2)h$ for $j = 1, \ldots, n$. In general, the Taylor series for a function $f(x)$ expanded about $a$ is

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)(x - a)^2}{2} + \mathcal{O}(x^3).$$

To discretize $\frac{\partial^2}{\partial s^2} x(s_i, t_j)$, we will first fix $t_i$ and expand $x(s_i + h, t_i)$ about $s_i$. This gives

$$\begin{aligned} x(s_i + h, t_j) &= x(s_i, t_j) + \frac{\partial}{\partial s} x(s_i, t_j)(s_i + h - s_i) + \frac{\frac{\partial^2}{\partial s^2}(s_i, t_j)(h)^2}{2} + \mathcal{O}(h^3) \\ &= x(s_i, t_j) + \frac{\partial}{\partial s} x(s_i, t_j)(h) + \frac{\frac{\partial^2}{\partial s^2} x(s_i, t_j)(h)^2}{2} + \mathcal{O}(h^3) \end{aligned}$$

and then, solving for $-\frac{\partial^2}{\partial s^2}x(s_i,t_j)$, we have

$$-\frac{\partial^2}{\partial s^2}x(s_i,t_j) = \frac{-2x(s_{i+1},t_j) + 2x(s_i,t_j) + 2\frac{\partial}{\partial s}x(s_i,t_j)(h) + \mathcal{O}(h^3)}{h^2}. \tag{3.8}$$

Now expanding $x(s_i - h, t_j)$ about $s_i$ gives:

$$\begin{aligned} x(s_i-h,t_j) &= x(s_i,t_j) + \frac{\partial}{\partial s}x(s_i,t_j)(s_i-h-s_i) + \frac{\frac{\partial^2}{\partial s^2}(s_i,t_j)(-h)^2}{2} + \mathcal{O}(h^3) \\ &= x(s_i,t_j) + \frac{\partial}{\partial s}x(s_i,t_j)(-h) + \frac{\frac{\partial^2}{\partial s^2}x(s_i,t_j)(-h)^2}{2} + \mathcal{O}(h^3), \end{aligned}$$

and so

$$-\frac{\partial^2}{\partial s^2}x(s_i,t_j) = \frac{-2x(s_{i-1},t_j) + 2x(s_i,t_j) - 2\frac{\partial}{\partial s}x(s_i,t_j)(h) + \mathcal{O}(h^3)}{h^2}. \tag{3.9}$$

Now adding the expansions (3.8) and (3.9) and solving for $-\frac{\partial^2}{\partial s^2}x(s_i,t_j)$, we have

$$-\frac{\partial^2}{\partial s^2}x(s_i,t_j) = \frac{-x(s_{i-1},t_j) + 2x(s_i,t_j) - x(s_{i+1},t_j)}{h^2} + E \tag{3.10}$$

with $E$ of order $h^2$ since the terms of order $h^3$ cancel in the numerator. By the same process, we can write

$$-\frac{\partial^2}{\partial t^2}x(s_i,t_j) = \frac{-x(s_i,t_{j-1}) + 2x(s_i,t_j) - x(s_i,t_{j+1})}{h^2} + E. \tag{3.11}$$

Thus, the Laplacian can be discretized in the two-dimensional case by writing

$$\begin{aligned} -\Delta x(s_i,t_j) &= \left(-\frac{\partial^2}{\partial s^2} - -\frac{\partial^2}{\partial t^2}\right)x(s_i,t_j) \\ &\approx \frac{-x(s_{i-1},t_j) + 2x(s_i,t_j) - x(s_{i+1},t_j)}{h^2} \end{aligned}$$

$$\begin{aligned} & + \frac{-x(s_i,t_{j-1})+2x(s_i,t_j)-x(s_i,t_{j+1})}{h^2} \\ = & \frac{-x(s_{i-1},t_j)-x(s_i,t_{j-1})+4x(s_i,t_j)-x(s_{i+1},t_j)-x(s_i,t_{j+1})}{h^2}. \end{aligned}$$

Notice that (3.10) and (3.11) each give rise to $n^2$ equations. To define the $n \times n$ array $\mathbf{X}$ correctly, we must let $[\mathbf{X}]_{ji} = x(s_i, t_{n+1-j}) = x_{i(n+1-j)}$ for $i, j = 1, \dots, n$. The indices are reversed since they refer to spatial location and all entries in first column, for example, have the same $s$ coordinate. Thus, $\mathbf{X}$ has the form

$$\mathbf{X} = \begin{bmatrix} x(s_1,t_n) & x(s_2,t_n) & \dots & x(s_n,t_n) \\ \vdots & \vdots & \vdots & \vdots \\ x(s_1,t_2) & x(s_2,t_2) & \dots & x(s_n,t_2) \\ x(s_1,t_1) & x(s_2,t_1) & \dots & x(s_n,t_1) \end{bmatrix} = \begin{bmatrix} x_{1n} & x_{2n} & \dots & x_{nn} \\ \vdots & \vdots & \vdots & \vdots \\ x_{12} & x_{22} & \dots & x_{n2} \\ x_{11} & x_{21} & \dots & x_{n1} \end{bmatrix}.$$

For the $n \times n$ matrix $\mathbf{L}$, which is equal to

$$\mathbf{L}_0 = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix} \quad \text{or} \quad \mathbf{L}_p = \begin{bmatrix} 2 & -1 & 0 & \dots & -1 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ -1 & \dots & 0 & -1 & 2 \end{bmatrix}$$

for zero or periodic boundary conditions, respectively, the $n^2$ equations defined by (3.10) can be written

$$\left[\frac{1}{h^2}\mathbf{X}\mathbf{L}\right]_{ji} = \frac{-x(s_{i-1},t_j)+2x(s_i,t_j)-x(s_{i+1},t_j)}{h^2} \approx -\frac{\partial^2}{\partial s^2}x(s_i,t_j)$$

and the equations defined by (3.11) can be expressed

$$\left[\frac{1}{h^2}\mathbf{L}\mathbf{X}\right]_{ji} = \frac{-x(s_i,t_{j-1})+2x(s_i,t_j)-x(s_i,t_{j+1})}{h^2} \approx -\frac{\partial^2}{\partial t^2}x(s_i,t_j).$$

This means the finite difference approximation of $\left(-\frac{\partial^2}{\partial s^2}-\frac{\partial^2}{\partial t^2}\right)x(s,t)$ is given by $(1/h^2)\mathrm{vec}(\mathbf{XL}+\mathbf{LX})$, which can be equivalently expressed as $(1/h^2)(\mathbf{L}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{L})\boldsymbol{x}$ where $\boldsymbol{x}=\mathrm{vec}(\mathbf{X})=(x_{1n},x_{1(n-1)},\dots,x_{11},x_{2n},\dots,x_{n1})^T$ using (3.1) and the fact that $\mathbf{L}$ is symmetric. Therefore,

$$-\Delta x(\boldsymbol{u}) = \left(-\frac{\partial^2}{\partial u_1^2}-\frac{\partial^2}{\partial u_2^2}\right)x(\boldsymbol{u}) \approx (1/h^2)(\mathbf{L}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{L})\boldsymbol{x}.$$

We now must discretize the $\frac{\partial^2}{\partial w_1\partial w_2}$ operator in the anisotropic SPDE. A similar process involving the Taylor series for functions of two variables and expanding $x(s_i+h,t_j+h)$, $x(s_i-h,t_j+h)$, $x(s_i+h,t_j-h)$, and $x(s_i-h,t_j-h)$ about the point $(s_i,t_j)$ will yield

$$\frac{\partial^2}{\partial s\partial t}x(s_i,t_j) \approx \frac{x(s_{i-1},t_{j-1})-x(s_{i-1},t_{j+1})-x(s_{i+1},t_{j-1})+x(s_{i+1},t_{j+1})}{4h^2}, \tag{3.12}$$

which also defines $n^2$ equations. For the matrix $\mathbf{K}$, which is equal to

$$\mathbf{K}_0 = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 0 \\ 0 & -1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & 0 & \dots & -1 & 0 \end{bmatrix} \quad \text{or} \quad \mathbf{K}_p = \begin{bmatrix} 0 & 1 & 0 & \dots & -1 \\ -1 & 0 & 1 & \dots & 0 \\ 0 & -1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 1 & 0 & \dots & -1 & 0 \end{bmatrix},$$

for zero or periodic boundary conditions, respectively, the $n^2$ equations defined by (3.12) can be expressed

$$\left[\frac{1}{4h^2}\mathbf{KXK}\right]_{ji} \approx \frac{\partial^2}{\partial s \partial t} x(s_i, t_j).$$

Writing this in vector notation, $\text{vec}(\mathbf{KXK}) = (\mathbf{K}^T \otimes \mathbf{K})\boldsymbol{x} = -(\mathbf{K} \otimes \mathbf{K})\boldsymbol{x}$.

Finally, it is well known that the discretization of Gaussian white noise with unit variance, as we have on the right-hand side of our SPDEs, yields a Gaussian random variable with mean zero and identity covariance matrix. That is, the discretization of $\mathcal{W}(s,t)$ is $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$ for $N = n^2$.

Generalizing all of the discretizations presented above and defining $\mathbf{L}_{\text{2D}} := \mathbf{L} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{L}$, the discretization of the isotropic SPDE given by (2.1) is

$$(\mathbf{I}_N + (\ell/h)^2 \mathbf{L}_{\text{2D}})^{(\nu + d/2)/2} \boldsymbol{x} = \delta^{-1/2} \boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N) \tag{3.13}$$

and the discretization of the anisotropic SPDE (2.12) is

$$\begin{aligned}\Big[\mathbf{I} + &\frac{1}{h^2}(a_\theta^2 + b_\theta^2)(\mathbf{L} \otimes \mathbf{I}) + \frac{1}{h^2}(c_\theta^2 + d_\theta^2)(\mathbf{I} \otimes \mathbf{L}) \\ &- \frac{2}{4h^2}(a_\theta c_\theta - b_\theta d_\theta)(\mathbf{K} \otimes \mathbf{K})\Big]^{(\nu + d/2)/2} \boldsymbol{x} = \delta^{-1/2} \boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N).\end{aligned}$$

In both cases, $\delta$ is the scaling parameter for the prior.

When discretizing the SPDEs, there is a scaling factor needed that guarantees that the variance scales systematically with respect to the change of the length-scaling parameters, $\ell$ or $\ell_1$ and $\ell_2$. The exact form of this scaling factor is unimportant for our purposes since we are ultimately only interested in a regularization parameter, $\alpha$, as will be seen in Section 3.4. To keep notation simpler, we use $\delta$ as a placeholder for this term. This is also the reason we are interested in whether the Matérn correlation rather than the covariance is preserved when restricting our Gaussian field to a finite domain, as we will discuss in Section 3.3.

### 3.2.1 Prior Precision Matrices

Now that we have the SPDE discretizations, we can solve each for the unknown $\boldsymbol{x}$ to get the priors we need to compute the Bayesian inverse problem solution. In the isotropic case,

$$\boldsymbol{x} = \delta^{-1/2}(\mathbf{I}_N + (\ell/h)^2\mathbf{L}_{\text{2D}})^{-(\nu+d/2)/2}\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$$

and thus

$$\boldsymbol{x}|\delta,\ell \sim \mathcal{N}\left(\mathbf{0}, \delta^{-1}(\mathbf{I} + (\ell/h)^2\mathbf{L}_{\text{2D}})^{-\nu-d/2}\right),$$

or equivalently,

$$p(\boldsymbol{x}|\delta,\nu,\ell) \propto \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}_{\nu,\ell}\boldsymbol{x}\right) \tag{3.14}$$

with $\mathbf{P}_{\nu,\ell} = (\mathbf{I} + (\ell/h)^2\mathbf{L}_{\text{2D}})^{\nu+d/2}$. In the anisotropic case,

$$p(\boldsymbol{x}|\delta,\nu,\ell_1,\ell_2,\theta) \propto \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}_{\nu,\ell_1,\ell_2,\theta}\boldsymbol{x}\right) \tag{3.15}$$

where

$$\begin{aligned}\mathbf{P}_{\nu,\ell_1,\ell_2,\theta} = \Big[\mathbf{I} &+ \frac{1}{h^2}(a_\theta^2 + b_\theta^2)(\mathbf{L}\otimes\mathbf{I}) + \frac{1}{h^2}(c_\theta^2 + d_\theta^2)(\mathbf{I}\otimes\mathbf{L}) \\ &- \frac{2}{4h^2}(a_\theta c_\theta - b_\theta d_\theta)(\mathbf{K}\otimes\mathbf{K})\Big]^{\nu+d/2}.\end{aligned} \tag{3.16}$$

In either of these expressions for $\mathbf{P}$ ($\mathbf{P}_{\nu,\ell}$ or $\mathbf{P}_{\nu,\ell_1,\ell_2,\theta}$), if $\nu + d/2$ is a non-integer, a fractional power of $\mathbf{P}$ must be computed, which is possible, generally speaking, if we have a diagonalization of $\mathbf{P}$ in hand, but the resulting precision matrix is typically full and dense. Such a diagonalization is typically computable in one-dimensional examples, even with dense matrices. In two dimensions, however, an efficient diagonalization is possible only if periodic boundary conditions are assumed. We will restrict the exponent $\nu+d/2$ to be an integer in this thesis to preserve the sparsity in the precision matrix, which will be especially useful in Chapter 5.

## 3.3 The Effect of a Finite Domain and Boundary Conditions

The proof of Theorem 2.2.1 assumed that the domain was all of $\mathbb{R}^d$, i.e. $\Omega = \mathbb{R}^d$. When solving inverse problems, however, as we mentioned before, $x(\boldsymbol{u})$ is restricted to a finite domain $\Omega \subset \mathbb{R}^d$. In such cases, boundary conditions that modify the Green's function must be assumed, and thus the equivalence between the Gaussian fields defined by the SPDE (2.1) and those defined by the Matérn covariance function may not hold.

To see this, consider the case where $d = 2$ and $\Omega = [0,1] \times [0,1]$ with Dirichlet (zero) boundary conditions, $x(0,t) = x(1,t) = x(s,0) = x(s,1) = 0$, where $0 \leq s,t \leq 1$. Additionally, to simplify the process, we assume $\nu = 1$ so that the exponent of the differential operator is equal to one. In this case, the discretization of (2.1), given by (3.13), simplifies to

$$(\mathbf{I}_N + (\ell/h)^2\mathbf{L}_{2\text{D}})\boldsymbol{x} = \delta^{-1/2}\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$$

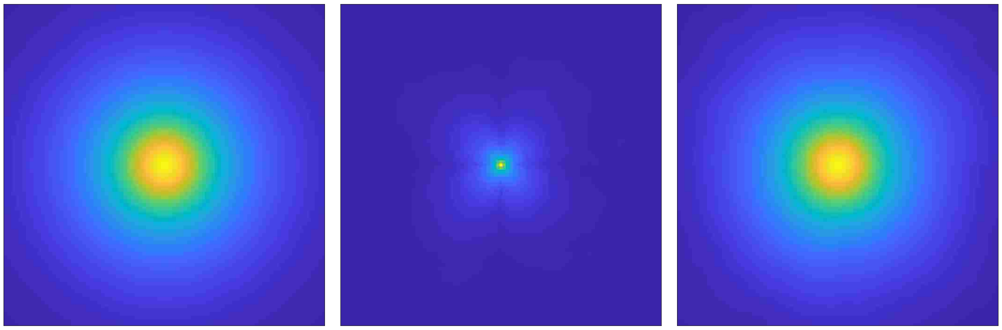

Figure 3.2: Isotropic correlation maps. Plots of the Matérn correlation map (left), the empirical correlation map with $n = 50$ computed on the domain $\Omega = [0, 1] \times [0, 1]$ (middle), and the empirical correlation map computed on the domain $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$ (right), computed from random draws from the prior (3.17) in 2D with $\nu = 1$ and $\ell = 1/4$.

and so the probability density for $\boldsymbol{x}$ is given by

$$p(\boldsymbol{x}|\delta, \ell) \propto \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T(\mathbf{I} + (\ell/h)^2\mathbf{L}_{2\mathrm{D}})^2\boldsymbol{x}\right). \tag{3.17}$$

We now let $n = 50$, so $N = 50^2 = 2500$, and generate 50 000 samples from (3.17) for each of $N$ $x_i$ values, calculate the empirical correlation between the samples, and compare this with the theoretical correlation defined by the Matérn covariance function. We do this for $\ell = 1/4$ and plot the results in the middle of Figure 3.2, together with the Matérn correlation map on the left. It is clear that there is a disconnection between the empirical correlation and the Matérn correlation.

*It is crucial that the connection between the Gaussian fields defined by the SPDE and those defined by the Matérn covariance function holds because then the parameters in the SPDE can be estimated using the semivariogram method described in Chapter 4.* Fortunately, we can restore this connection by extending the computational domain. In two dimensions, we define $\overline{\Omega} = [1-a, a] \times [1-a, a]$, for $a > 1$, e.g., if $a = 1.5$ then $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$. We then generate realizations for $((2a-1)n)^2 = (2n)^2 = 10\,000$ $x_i$ values on the extended domain and compute the empirical correlation only for the $x_i$ values that correspond to the original domain, $\Omega = [0, 1] \times [0, 1]$. The results are plotted on the right side of Figure 3.2, where it is clear that the empirical correlation map is nearly indistinguishable from those obtained using the Matérn correlation function.

To determine the $a$ value that extends the domain far enough to restore the Matérn/SPDE connection, but not so far as to introduce unnecessary computational cost, we look to the Matérn correlation function itself. We want to extend the domain far enough so that all $\boldsymbol{x}$ values in $[0, 1] \times [0, 1]$ have a sufficiently low correlation with the $\boldsymbol{x}$ values at the end of the extended domain. The criterion we used to determine if the connection was restored was based on relative error: $\|\boldsymbol{\rho} - \boldsymbol{\rho}_a\|_F / \|\boldsymbol{\rho}\|_F < 0.05$, where $\boldsymbol{\rho}$ is the true Matérn correlation matrix, $\boldsymbol{\rho}_a$ is the approximate correlation matrix obtained by discretizing the SPDE, and $\|\cdot\|_F$ denotes the Frobenius norm.

In tests, it was found that we should always extend the domain at least slightly. If we let $r_c$ be the distance for which the Matérn correlation is approximately equal to $c$, then our tests showed that setting $a = 1 + r_{0.30}$ restores the connection to the Matérn covariance for $\nu \geq 1/2$ when using zero boundary conditions and setting $a = 1 + r_{0.20}$ restores the connection to the Matérn covariance for $\nu \geq 1/2$ when periodic boundary conditions are used. For $\nu = 1$ and $\ell = 1/4$, $a$ should be set to 1.5 in the Dirichlet boundary condition case, which gives a relative error in the difference of the correlation matrices of 0.0375, and it should be set to 1.6 when using periodic boundary conditions.

We note that since $\ell$ is directly related to the degree of correlation in the prior, the extension necessary to preserve the connection rises sharply as $\ell$ increases. It is rare in practice, however, to have $\ell \geq 1/4$ when $\nu \geq 1$ since that implies the correlation persists across the entire region. Thus, it is uncommon to have to extend beyond a domain of $[-0.5, 1.5] \times [-0.5, 1.5]$.

For the above discussion, we focused on zero boundary conditions. Similar results hold if periodic boundary conditions are assumed, in which case $\mathbf{L}$, and thus $\mathbf{L}_{2\text{D}}$, can be diagonalized by the DFT, assuming $\boldsymbol{x}$ is defined on a regular grid. The DFT-based diagonalization of $\mathbf{L}_{2\text{D}}$ can be exploited to greatly reduce computational cost, thus when extending the domain in two dimensions, it is advantageous to use periodic boundary conditions and the extended domain $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$ so that $\mathbf{L}_{2\text{D}}$ defined on $\overline{\Omega}$ can be diagonalized by the DFT. A more thorough description of the effects of boundary artifacts with different boundary conditions can be found in [Khristenko et al., 2019].

The same results hold when dealing with anisotropy. Now we generate samples from (3.15), compute the correlation, and compare it to the anisotropic Matérn correlation (2.13). Figure 3.3 shows the disconnection on the domain $\Omega = [0, 1] \times [0, 1]$ and the reconnection when extending the domain to $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$. The $a$ values of $a = 1 + r_{0.30}$ when using zero boundary conditions or $a = 1 + r_{0.20}$ when using periodic boundary conditions are calculated using the isotropic Matérn correlation function with $\ell = \ell_1$.

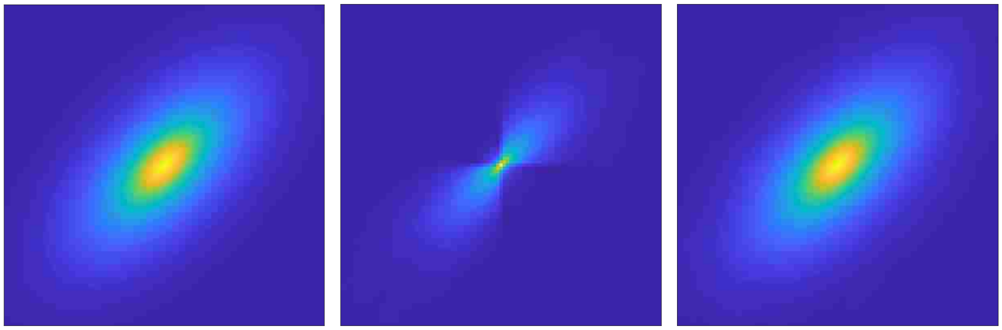

Figure 3.3: Anisotropic correlation maps. Plots of the anisotropic Matérn correlation map (left), the empirical correlation map with $n = 50$ computed on the domain $\Omega = [0,1] \times [0,1]$ (middle), and the empirical correlation map computed on the domain $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$ (right), computed from random draws from the prior (3.15) in 2D with $\nu = 1$, $\ell_1 = 1/4$, $\ell_2 = 1/8$, and $\theta = 45°$.

## 3.4 Computing MAP Estimators for Whittle-Matérn Priors

Typically, the sizes of the matrices and vectors we work with are as follows: $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{P} \in \mathbb{R}^{N \times N}$, $\boldsymbol{x} \in \mathbb{R}^{N}$, and $\boldsymbol{b} \in \mathbb{R}^{M}$ with $M = m$ and $N = n$ in one dimension and $M = m^2$ and $N = n^2$ for arrays $\mathbf{X} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times m}$ where $\boldsymbol{x} = \operatorname{vec}(\mathbf{X})$ and $\boldsymbol{b} = \operatorname{vec}(\mathbf{B})$ in two dimensions. When extending the computational domain, however, these sizes will change slightly. We define the size of the extended domain as an $n \times n$ square and we now introduce a matrix $\mathbf{M}$ that will restrict $\mathbf{A}\boldsymbol{x}$ to its central $m \times m$ elements. Thus, the setup will be

$$\boldsymbol{b} = \mathbf{M}\mathbf{A}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{3.18}$$

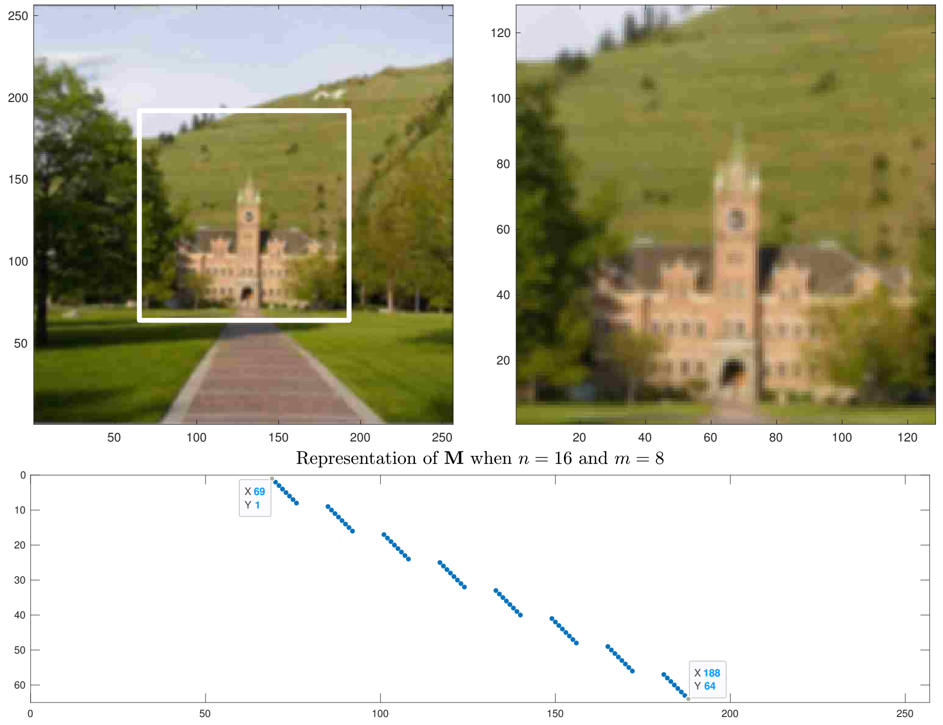
Representation of $\mathbf{M}$ when $n = 16$ and $m = 8$

Figure 3.4: Effect of the masking matrix. On the top left is the array $\mathbf{A}\boldsymbol{x}$ that is of size $n \times n$, on the top right is $\mathbf{M}\mathbf{A}\boldsymbol{x}$ in array form that is $m \times m$, and on the bottom is a representation of the matrix $\mathbf{M}$ when $n = 16$ and $m = 8$.

for $\mathbf{M} \in \mathbb{R}^{M \times N}$, $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\boldsymbol{x} \in \mathbb{R}^{N}$, and $\boldsymbol{b}, \boldsymbol{\epsilon} \in \mathbb{R}^{M}$ (so $\mathbf{X} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times m}$).

Figure 3.4 shows on the top left a representation of $\mathbf{A}\boldsymbol{x}$ reshaped into an $n \times n$ array. On the top right is $\mathbf{M}\mathbf{A}\boldsymbol{x}$ as an $m \times m$ array. Multiplying $\mathbf{A}\boldsymbol{x}$ by $\mathbf{M}$ removed all elements outside of the central box. On the bottom is a look at the components of the matrix $\mathbf{M}$ in a smaller case when $n = 16$ and $m = 8$. In general, when extending in such a way that $n = 2m$, which will be the case when $\Omega = [0, 1] \times [0, 1]$ and $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$, $\mathbf{M}$ has the following form:

$$\mathbf{M} = \left[\begin{array}{ccccccccccccc} \multicolumn{3}{c}{\overbrace{\qquad\qquad\qquad}^{m\text{ groups}}} & \multicolumn{7}{c}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{2m\text{ groups}}} & \multicolumn{3}{c}{\overbrace{\qquad\qquad\qquad}^{m\text{ groups}}} \\ \mathbf{0}_m & \dots & \mathbf{0}_m & \mathbf{I}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \dots & \mathbf{0}_m & \mathbf{0}_m & \dots & \mathbf{0}_m \\ \mathbf{0}_m & \dots & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{I}_m & \mathbf{0}_m & \mathbf{0}_m & \dots & \mathbf{0}_m & \mathbf{0}_m & \dots & \mathbf{0}_m \\ \mathbf{0}_m & \dots & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{I}_m & \dots & \mathbf{0}_m & \mathbf{0}_m & \dots & \mathbf{0}_m \\ \vdots & \dots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \dots & \vdots & & \\ \mathbf{0}_m & \dots & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m & \dots & \mathbf{I}_m & \mathbf{0}_m & \dots & \mathbf{0}_m \end{array}\right],$$

where $\mathbf{0}_m$ is an $m \times m$ matrix of all zeros. Therefore, $\mathbf{M}$ is of size $m^2 \times 4m^2 = m^2 \times n^2 = M \times N$.

Now we can obtain our MAP estimate by computing

$$\begin{aligned} \boldsymbol{x}_\alpha &= \arg\min_{\boldsymbol{x}} \left\{ \frac{\lambda}{2} \|\mathbf{M}\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \frac{\delta}{2} \boldsymbol{x}^T \mathbf{P} \boldsymbol{x} \right\} \\ &= \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|\mathbf{M}\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \frac{\alpha}{2} \boldsymbol{x}^T \mathbf{P} \boldsymbol{x} \right\} \\ &= \left( \mathbf{A}^T \mathbf{M}^T \mathbf{M} \mathbf{A} + \alpha \mathbf{P} \right)^{-1} \mathbf{A}^T \mathbf{M}^T \boldsymbol{b}, \end{aligned} \tag{3.19}$$

where $\alpha = \delta/\lambda$ and $\mathbf{P} = \mathbf{P}_{\nu,\ell} = (\mathbf{I} + (\ell/h)^2 \mathbf{L}_{2\mathrm{D}})^{\nu + d/2}$ when we have isotropy or $\mathbf{P} = \mathbf{P}_{\nu,\ell_1,\ell_2,\theta}$ as in (3.16) if we are dealing with anisotropy. Note that $\boldsymbol{x}_\alpha \in \mathbb{R}^N$ is defined on the extended domain $\overline{\Omega}$, so we will need to multiply $\boldsymbol{x}_\alpha$ by $\mathbf{M}$ to extract the values that correspond to the central $m \times m$ elements to obtain the solution on the original domain $\Omega$. The extracting matrix $\mathbf{M}$ from (3.18) and (3.19) is not circulant, so while we can multiply $\mathbf{A}^T \mathbf{M}^T \boldsymbol{b}$ and compute $(\mathbf{A}^T \mathbf{M}^T \mathbf{M} \mathbf{A} + \alpha \mathbf{P})\boldsymbol{v}$ for a vector $\boldsymbol{v} \in \mathbb{R}^N$ using DFTs, we cannot use DFTs to obtain or multiply by $\left( \mathbf{A}^T \mathbf{M}^T \mathbf{M} \mathbf{A} + \alpha \mathbf{P} \right)^{-1}$. Thus, we will use PCG to iteratively solve $\left( \mathbf{A}^T \mathbf{M}^T \mathbf{M} \mathbf{A} + \alpha \mathbf{P} \right) \boldsymbol{x}_\alpha = \mathbf{A}^T \mathbf{M}^T \boldsymbol{b}$ for $\boldsymbol{x}_\alpha$ with preconditioner $\mathbf{G} = \mathbf{A}^T \mathbf{A} + \alpha \mathbf{P}$, which is circulant when periodic boundary conditions are

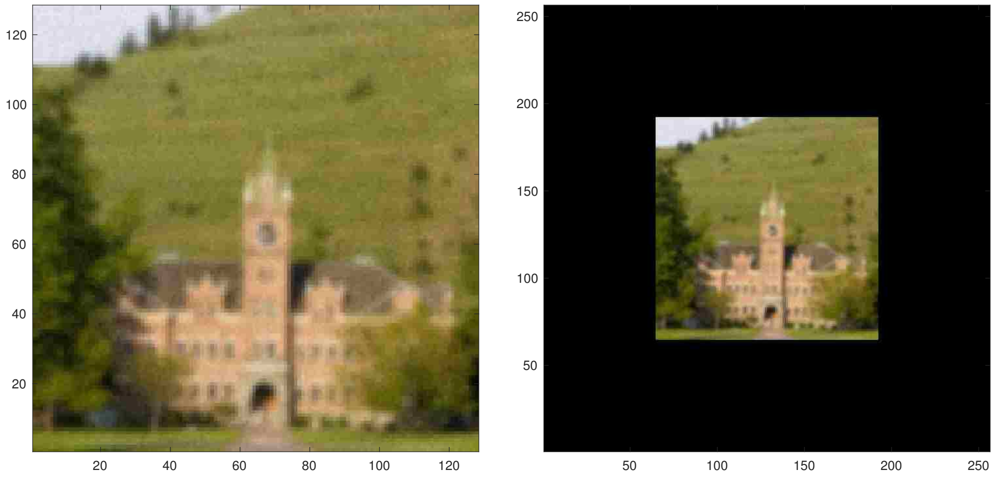

Figure 3.5: Extended domain images. On the left is an array $\mathbf{B}$ that is of size $m \times m$ and on the right is the zero-padded $\tilde{\mathbf{B}}$ that is of size $n \times n$ on the extended domain.

used. Assuming we know the hyperparameters in $\mathbf{P}$, $\alpha$ can be estimated using one of many regularization parameter selection methods (see, e.g.,[Vogel, 2002, Hansen, 2005, Bardsley, 2018]). One such method is generalized cross validation (GCV):

$$\alpha = \arg\min_{\eta>0} \left\{ \frac{\left\| \mathbf{MA}\left(\mathbf{A}^T\mathbf{M}^T\mathbf{MA} + \eta\mathbf{P}\right)^{-1}\mathbf{A}^T\mathbf{M}^T\boldsymbol{b} - \boldsymbol{b} \right\|^2}{\operatorname{tr}\left(\mathbf{I} - \mathbf{MA}\left(\mathbf{A}^T\mathbf{M}^T\mathbf{MA} + \eta\mathbf{P}\right)^{-1}\mathbf{A}^T\mathbf{M}^T\right)} \right\}. \tag{3.20}$$

PCG will also have to be used when finding $\alpha$.

When implementing the calculation of the MAP estimator using arrays and DFTs, we first zero-pad the array $\mathbf{B}$ so it is of size $n \times n$. The zero-padded array will be denoted $\tilde{\mathbf{B}}$. An example of this is illustrated in Figure 3.5 for $m = 128$ and $n = 256$. On the left, $\mathbf{B}$ is presented as an $m \times m$ array and on the right is the zero-padded $\tilde{\mathbf{B}}$ as an $n \times n$ array. Next we define the array

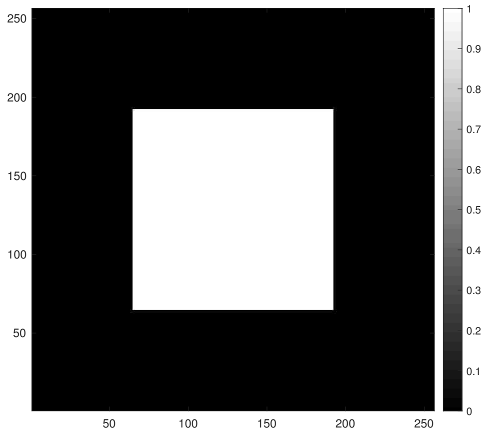

Figure 3.6: $\mathbf{M}_a$ array. A standard example when using DFTs and extending the computational domain that shows the central $m \times m$ elements of $\mathbf{M}_a$ being equal to one and all the rest being equal to zero.

$\mathbf{M}_a$ as an $n \times n$ square with all the array values being equal to zero outside of the central $m \times m$ region. The array values in the central $m \times m$ region of $\mathbf{M}_a$ are dependent on the values of $\mathbf{B}$. If we let $\mathbf{M}_a^{m \times m}$ denote the central $m \times m$ region of $\mathbf{M}_a$, then

$$[\mathbf{M}_a^{m \times m}]_{ij} = \begin{cases} 1 & \text{if } [\mathbf{B}]_{ij} \text{ is not missing} \\ 0 & \text{if } [\mathbf{B}]_{ij} \text{ is missing} \end{cases}$$

for $i, j = 1, \ldots, m$. An example of $\mathbf{M}_a$ is given in Figure 3.6, where the central $m \times m$ region of $\mathbf{M}_a$ are all ones, as will be the case when there are no missing values in $\mathbf{B}$.

Now our solution on the extended domain will be computed as an array, not a vector, which we denote $\mathbf{X}_\alpha$. Then, we can take the central $m \times m$ region of $\mathbf{X}_\alpha$ as our reconstruction. To compute $\mathbf{X}_\alpha$, we use PCG with $\mathbf{A}_{\text{PCG}} = \mathbf{A}^T\mathbf{M}^T\mathbf{M}\mathbf{A} + \alpha\mathbf{P}$ and $\boldsymbol{b}_{\text{PCG}} = \mathbf{A}^T\mathbf{M}^T\boldsymbol{b}$, where $\mathbf{A}_{\text{PCG}}$ and $\boldsymbol{b}_{\text{PCG}}$ correspond to the $\mathbf{A}$ and $\boldsymbol{b}$ in Algorithm 2. We can compute $\boldsymbol{b}_{\text{PCG}}$ using DFTs with

$$\boldsymbol{b}_{\text{PCG}} = \mathbf{A}^T\mathbf{M}^T\boldsymbol{b} = \text{vec}\left(\mathbf{F}^*\left(\text{conj}(\widehat{\boldsymbol{a}}_s)\odot[\mathbf{F}\tilde{\mathbf{B}}\mathbf{F}]\right)\mathbf{F}^*\right),$$

and in order to multiply $\mathbf{A}_{\text{PCG}}$ by some vector $\boldsymbol{v}$ of conformable size, as will need to be done for each iteration of PCG, we use the following:

$$\begin{aligned}&\mathbf{A}_{\text{PCG}}\boldsymbol{v} = \left(\mathbf{A}^T\mathbf{M}^T\mathbf{M}\mathbf{A}+\alpha\mathbf{P}\right)\boldsymbol{v}\\ &= \text{vec}\Big(\mathbf{F}^*\Big(\text{conj}(\widehat{\boldsymbol{a}}_s)\odot\Big[\mathbf{F}\Big(\mathbf{M}_a\odot[\mathbf{F}^*(\widehat{\boldsymbol{a}}_s\odot[\mathbf{FVF}])\mathbf{F}^*]\Big)\mathbf{F}\Big]+\alpha(\hat{\boldsymbol{p}}_s\odot[\mathbf{FVF}])\Big)\mathbf{F}^*\Big),\end{aligned}$$

where $\widehat{\boldsymbol{a}}_s$ and $\hat{\boldsymbol{p}}_s$ are $n\times n$ matrices of the eigenvalues of $\mathbf{A}$ and $\mathbf{P}$, respectively, and $\boldsymbol{v} = \text{vec}(\mathbf{V})$. Using MATLAB, $\mathbf{A}_{\text{PCG}}\boldsymbol{v}$ and $\boldsymbol{b}_{\text{PCG}}$ can be computed by typing

```
vec(ifft2(conj(ahat).*fft2(Marray.*ifft2(ahat.*fft2(V))))
          +alpha*ifft2(phat.*fft2(V)))
```

and vec(`ifft2(conj(ahat).*fft2(b_pad)`), respectively. Also, since $\mathbf{G} = \mathbf{A}^T\mathbf{A}+\alpha\mathbf{P}$ is circulant, $\mathbf{G}^{-1}\boldsymbol{v}$ can be easily computed with

$$\mathbf{G}^{-1}\boldsymbol{v} = \left(\mathbf{A}^T\mathbf{A}+\alpha\mathbf{P}\right)^{-1}\boldsymbol{v} = \text{vec}\left(\mathbf{F}^*\left(\frac{\mathbf{FVF}}{|\widehat{\boldsymbol{a}}_s|^2+\alpha\widehat{\boldsymbol{p}}_s}\right)\mathbf{F}^*\right),$$

where division is component wise.

An example of a reconstruction of $\mathbf{X}_\alpha$ is shown in Figure 3.7. $\mathbf{X}_\alpha$ is the $n\times n$ reconstruction on the extended domain and the solution on the original domain is obtained by taking the central $m\times m$ elements of $\mathbf{X}_\alpha$, which corresponds to the part of the image that is inside the rectangle. Notice that another benefit

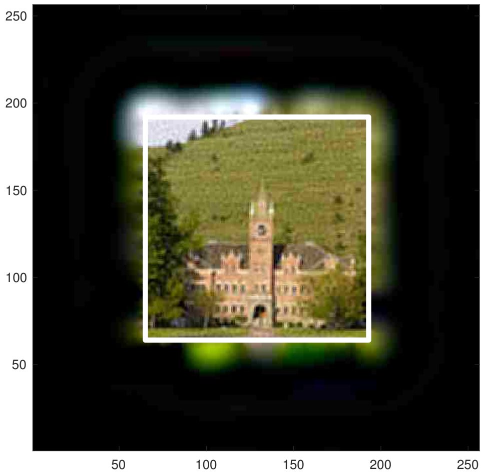

Figure 3.7: Example reconstruction. $\mathbf{X}_\alpha$ is the $n \times n$ reconstruction on the extended domain and the central $m \times m$ elements of $\mathbf{X}_\alpha$ give the reconstruction on the original domain.

of extending the domain is that any boundary artifacts that may have been present are moved outside of the central region so they do not tarnish the reconstruction.

Although we can now obtain the solution to the inverse problem when we know $\mathbf{P}$, the hyperparameters of $\mathbf{P}$ still need to be set. In practice, $\nu$ is often fixed [Khaledi and Rivaz, 2009, Roininen et al., 2016] and the other hyperparameters – $\ell$ in the isotropic case or $\ell_1$, $\ell_2$, and $\theta$ in the anisotropic case – are either estimated manually or by using the fully-Bayesian approach, which involves Markov chain Monte Carlo (MCMC) [Robert and Casella, 2013] sampling. For the sake of comparison, we take the fully-Bayesian approach in Section 5.2, but it requires setting up hyperprior distributions and can be time consuming, subjective, and unintuitive. Estimating these hyperparameters is the primary focus of this work and a new method to do so is presented in Chapter 4.

# Chapter 4

# Semivariogram Methods

In the inverse problem formulation in this dissertation, the components of the vector $\boldsymbol{x}$ correspond to values of an unknown function $x$ at numerical mesh points within a spatial region $\Omega$. This motivates using methods from spatial statistics to estimate the Whittle-Matérn prior hyperparameters – $\nu$ and $\ell$ in the isotropic case or $\nu$, $\ell_1$, $\ell_2$, and $\theta$ in the anisotropic case. One such method uses a variogram, and a corresponding semivariogram [Schabenberger and Gotway, 2017], which requires the assumption of *intrinsic stationarity*, i.e., that the elements of $\boldsymbol{x}$ have constant mean and the variance of the difference between the elements is constant throughout the region. This is a weaker assumption than is required by many other parameter estimation tools, which is one of the reasons variograms have become popular in spatial statistical applications [Cressie, 2015], and it is the reason we use semivariograms here. Although the use of semivariograms for estimating parameters to determine a covariance structure is commonly used in spatial statistics, this is, to our knowledge, the first time these tools have been used to estimate prior hyperparameters for use in inverse problems.

## 4.1 Isotropic Case

The semivariogram is defined by $\gamma(\boldsymbol{r}) = \frac{1}{2}\mathrm{Var}[Z(\boldsymbol{u}_i) - Z(\boldsymbol{u}_j)]$, where $\boldsymbol{r} = \boldsymbol{u}_i - \boldsymbol{u}_j$ and $\{Z(\boldsymbol{u}) : \boldsymbol{u} \in \Omega \subset \mathbb{R}^d\}$ is a spatial process. Due to our stationarity assumption, the variance is constant throughout the region and so $\mathrm{Var}[Z(\boldsymbol{u}_i)] = \mathrm{Var}[Z(\boldsymbol{u}_j)] = \sigma^2$, which we use to derive the following alternative expression for $\gamma(\boldsymbol{r})$:

$$\begin{aligned}\gamma(\boldsymbol{r}) &= \frac{1}{2}\Big(\mathrm{Var}[Z(\boldsymbol{u}_i)] + \mathrm{Var}[Z(\boldsymbol{u}_j)] - 2\mathrm{Cov}[Z(\boldsymbol{u}_i), Z(\boldsymbol{u}_j)]\Big)\\ &= \sigma^2 - \mathrm{Cov}[Z(\boldsymbol{u}_i), Z(\boldsymbol{u}_j)].\end{aligned}$$

Thus, the semivariogram simplifies to the difference between the variance in the region and the covariance between two points with a difference $\boldsymbol{r}$. The variogram is formally defined as $2\gamma(\boldsymbol{r})$, hence the terms variogram and semivariogram are often used interchangeably. To remain consistent, we will continue to refer to $\gamma(\boldsymbol{r})$ as a semivariogram throughout the dissertation.

We now need a way to estimate the semivariogram from given data. For this, we use what is known as the sample, or empirical, semivariogram. Assuming that $Z(\boldsymbol{u})$ is isotropic, so that $r = \|\boldsymbol{r}\| = \|\boldsymbol{u}_i - \boldsymbol{u}_j\|$, then the empirical semivariogram can be expressed

$$\hat{\gamma}(r) = \frac{1}{2n(r)} \sum_{(i,j)|\|\boldsymbol{u}_i - \boldsymbol{u}_j\| = r} [z(\boldsymbol{u}_i) - z(\boldsymbol{u}_j)]^2, \tag{4.1}$$

where $z(\boldsymbol{u})$ is a realization of $Z(\boldsymbol{u})$, and $n(r)$ is the number of points that are separated by a distance $r$. The $\hat{\gamma}(r)$ values are often referred to as the semivariance values. In a typical semivariogram, the semivariance values increase as $r$ increases since points tend to be less similar the further apart they are, which increases the variance of their differences.

Although the empirical semivariogram is useful in obtaining semivariance values from data, it is not ideal for modeling data for a few reasons: 1) the empirical semivariogram only gives estimates at particular lag distances, as opposed to being a continuous function of distance that we will need; 2) the resulting covariance matrix may not be positive definite, which is a requirement for all covariance matrices; and 3) the sample semivariogram values will not necessarily increase smoothly with distance due to random noise in our data. Due to these limitations of the empirical semivariogram, it is typical to fit a semivariogram model to the empirical semivariogram. Since our prior distribution for $\boldsymbol{x}$ has a Matérn covariance, we will use the theoretical Matérn semivariogram model [Matérn, 2013, Stein, 2012] given by

$$\gamma(r, \boldsymbol{\theta}) = \begin{cases} 0 & \text{if} \quad r = 0 \\ a_0 + (\sigma^2 - a_0)\left[1 - \frac{1}{2^{\nu-1}\Gamma(\nu)}(r/\ell)^\nu K_\nu(r/\ell)\right] & \text{if} \quad r > 0 \end{cases}$$

where $a_0 \geq 0$ is the nugget, $\sigma^2 \geq a_0$ is the sill, and $\boldsymbol{\theta} = (a_0, \sigma^2, \nu, \ell)$. The nugget is the term given to the semivariance value at a distance just greater than zero and the sill is the total variance contribution or the semivariance value where the model levels out. The sill, $\sigma^2$, is also the variance parameter in the Matérn covariance function (1.6). We can estimate $a_0$, $\sigma^2$, $\nu$, and $\ell$ by fitting semivariogram models to the empirical semivariogram.

There are a number of ways to fit the semivariogram model to the empirical semivariogram. We use weighted least squares, as is commonly done [Cressie, 2015], choosing the $\boldsymbol{\theta}$ that minimizes

$$W(\boldsymbol{\theta}) = \sum_r \frac{n(r)}{2[\gamma(r,\boldsymbol{\theta})]^2}[\hat{\gamma}(r) - \gamma(r,\boldsymbol{\theta})]^2.$$

To minimize $W(\boldsymbol{\theta})$, we adapt the MATLAB codes from [Schwanghart, a, Schwanghart, b]. More specifically, we adapt [Schwanghart, a] for computing the empirical semivariance $\hat{\gamma}(r)$ and we adapt [Schwanghart, b] for minimizing $W(\boldsymbol{\theta})$. Although it is possible to optimize both $\nu$ and $\ell$ continuously, we will require $\nu + d/2$ to be an integer. Weighted least squares, in general, performs well when finding optimal estimates for $a_0, \sigma^2$, and $\ell$ for given empirical semivariogram values when $\nu$ is fixed, but not when $\nu$ is also free to vary (most software requires a fixed $\nu$ value). To combat this issue, and to ensure $\nu + d/2$ is an integer, we cycle through various fixed values of $\nu$ to obtain estimates for the other parameters and their weighted least squares value. We then choose the $\boldsymbol{\theta}$ with the smallest $W(\boldsymbol{\theta})$.

For an illustration, we generated a random field, shown on the left side of Figure 4.1, and fit a semivariogram to the field. The optimized parameters of the model are $\nu = 2$ and $\ell = 0.019$, which corresponds to a practical range of 0.102. Thus, the values of the field are nearly independent a tenth of the way across the region. The sill and nugget are estimated to be $\sigma^2 = 1.003$ and $a_0 = 0.206$, respectively. A plot of the resulting fitted Matérn semivariogram model is given on the right side of Figure 4.1.

The values of $\nu$ and $\ell$ from $\boldsymbol{\theta} = (a_0, \sigma^2, \nu, \ell)$ obtained by fitting the Matérn semivariogram model to a spatial field, as described in the previous paragraphs, can be used to define the Whittle-Matérn prior (3.14). The sill, $\sigma^2$, and the nugget, $a_0$, are not especially useful outside of fitting the semivariogram model

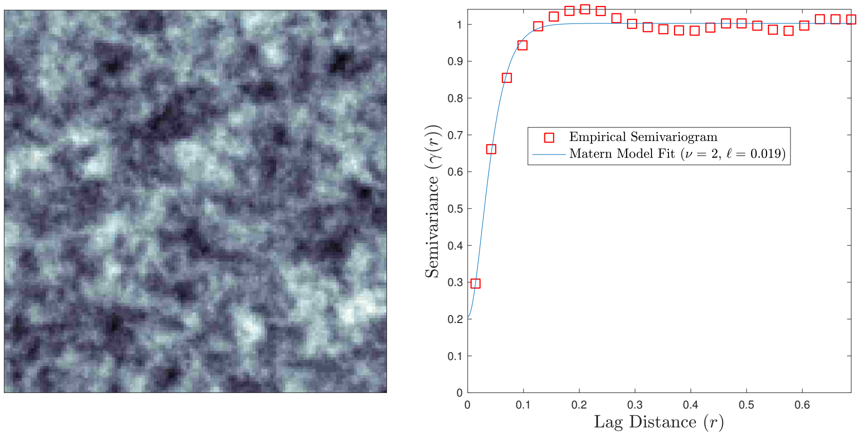

Figure 4.1: Semivariogram. A randomly generated spatial field is shown on the left and the empirical semivariogram, along with the Matérn model fit, is given on the right. The fitted hyperparameters are $\nu = 2$ and $\ell = 0.019$, which corresponds to a practical range of 0.102.

because they do not correspond to any hyperparameter in (3.14). They are helpful only in determining the best estimates for $\nu$ and $\ell$. Any contribution these parameters may have made to the prior distribution will be accounted for in the regularization parameter, $\alpha$. Therefore, after fitting the semivariogram models, $\sigma^2$ and $a_0$ are discarded.

### 4.1.1 Algorithm

With estimates for $\nu$ and $\ell$ in hand, the MAP estimator, $\boldsymbol{x}_\alpha$, can then be computed as in Section 3.4, from which we can recompute $\boldsymbol{\theta}$ by fitting the Matérn semivariogram model to the empirical semivariogram values of $\boldsymbol{x}_\alpha$. Repeating this process iteratively yields the following algorithm:

**Algorithm:** *The Semivariogram Method for MAP Estimation with Whittle-Matérn Prior:*

0. Estimate $\boldsymbol{\theta} = (a_0, \sigma^2, \nu, \ell)$ by fitting a Matérn semivariogram model to $\boldsymbol{b}$.

1. Define the prior (3.14) using $\nu$ and $\ell$, compute $\alpha$ using (3.20), and compute $\boldsymbol{x}_\alpha$ using (3.19).

2. Update $\boldsymbol{\theta} = (a_0, \sigma^2, \nu, \ell)$ by fitting a Matérn semivariogram model to $\boldsymbol{x}_\alpha$.

3. Return to step 1 and repeat until $\nu$ and $\ell$ stabilize.

Recall that $\boldsymbol{b}$ is a vector of measurements, which will usually be noisy or have some missing values, and each element of $\boldsymbol{b}$ has a corresponding spatial position. Since $\nu$ is being optimized discretely to ensure that $\beta = \nu + d/2$ is an integer, convergence will be met when $\nu_j - \nu_{j-1} = 0$ where $\nu_j$ is the $\nu$ value fit in the $j$th iteration. Then $\ell$ is said to have converged when $|\ell_j - \ell_{j-1}|/\ell_{j-1} < \varepsilon$ with $\varepsilon$ determined by the user. In this thesis, we will consider $\ell$ to have converged when the relative difference is less than 0.01, which usually takes fewer than three iterations to achieve.

### 4.1.2 Numerical Experiments

We now implement the semivariogram method on a two-dimensional deblurring and inpainting example. Recall that the connection between the Matérn covariance and the Whittle-Matérn prior depends on a stationarity assumption, which the following example may not exhibit. For simplicity, we will still assume stationarity and acknowledge that future work should be done in the case when no stationarity is present. Additionally, the numerical examples given in this thesis all use color images. In our analysis, we will assume independence in the color bands and obtain priors and reconstructions for each one individually.

### Results

In this example, we assume periodic boundary conditions on the extended domain, but due to the restriction from the extended domain $\overline{\Omega}$ to $\Omega$, as was mentioned in Section 3.4, circulant structure is lost in the forward model matrix, and hence, linear system solves must be done using an iterative method. As is done in [Bardsley, 2018, Section 3.1.3], we use preconditioned conjugate gradient (PCG) iteration, both for computing $\alpha$ and for computing $\boldsymbol{x}_\alpha$. We attempt to deblur and demask a $128 \times 128$ image of Main Hall on the University of Montana (UM) campus. To do this, we begin with a $256 \times 256$ image, given in Figure 4.2, and then restrict to the center $128 \times 128$ image. This smaller image in the middle will be thought of as being on a domain $\Omega = [0, 1] \times [0, 1]$

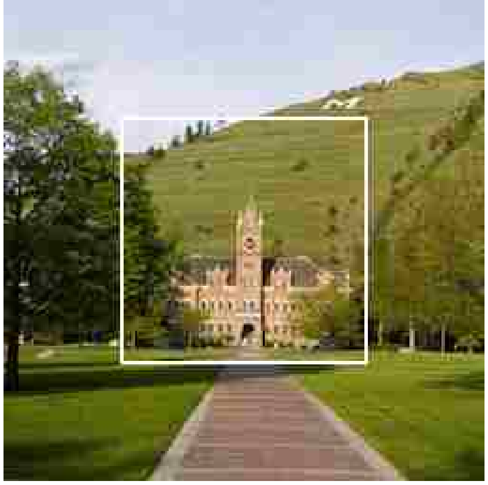

Figure 4.2: Full $256 \times 256$ image of Main Hall at the University of Montana with $128 \times 128$ subimage.

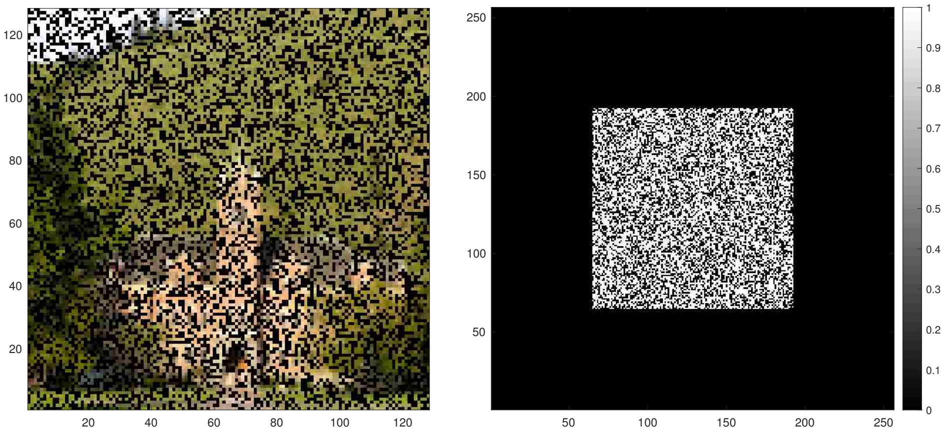

Figure 4.3: Inpainting setup. On the left is $\mathbf{B}$, the blurred, masked, and noisy data, in array form. On the right is the $\mathbf{M}_a$ array. All array values of $\mathbf{M}_a$ outside of the central $m \times m$ region and those that correspond to the missing elements in $\mathbf{B}$ are zero. All other values are equal to one.

and the larger, full image will then be defined on $\overline{\Omega} = [-0.5, 1.5] \times [-0.5, 1.5]$.

To obtain $\mathbf{B}$, and the corresponding $\boldsymbol{b} = \text{vec}(\mathbf{B})$, we first perform a slight blurring operation on the full 256×256 true image plotted in Figure 4.2. Since this is a color image, the blurring process is done individually for the red, green, and blue intensity arrays. We then restrict to the central $128 \times 128$ pixels (with boundaries marked in Figure 4.2) and randomly remove 40% of the pixels to obtain the masked and moderately blurry image on the left in Figure 4.3. The array $\mathbf{M}_a$ that we need when computing the solution is given on the right in Figure 4.3. In addition to the zeros for all array values outside of the central $128 \times 128$ region, all elements inside the central region that correspond to the missing values in $\mathbf{B}$ will be zeros as well. The other values of $\mathbf{M}_a$ are all equal to one.

We seek an estimate of $\boldsymbol{x}$ in the central subregion. Semivariograms with 25 approximately equally spaced grid points in $0 < r < \sqrt{2}/10$ are used. We chose $\sqrt{2}/10$ as a cutoff because it balances the need to capture the covariance structure at short distances, which are well-known to be the most important [Cressie, 2015], with those at longer distances. When fitting semivariograms to the masked image, the removed entries will not be considered or else the correlation would be strongly influenced by those entries.

The semivariogram method is used to obtain $\nu = 1$ for each color band, $\ell = 0.0339, 0.0292$ and $0.0530$ for the red, green, and blue intensities, respectively, and $\alpha = 2.24 \times 10^{-6}, 3.84 \times 10^{-6}$ and $1.27 \times 10^{-7}$. For reference, when fitting semivariograms to the true image, the values for $\ell$ are $0.0380, 0.0324$, and $0.0559$ for the red, green, and blue color bands, respectively. Convergence was met in two iterations for each color intensity. We also computed the Tikhonov solution, as defined in [Bardsley, 2018, Section 3.1.3], for which the prior covariance is equal to a scalar multiple of the identity matrix (i.e. $\mathbf{P} = \mathbf{I}_N$). The Tikhonov $\alpha$ values for all three color bands were around 0.0004. Note that for both of these reconstructions, the regularization parameter, $\alpha$, was optimized using the highest correlation between the solution and the true image rather than chosen by GCV to ensure that any differences in the solutions is due to the method and not a poorly-chosen regularization parameter.

The two solutions are plotted in Figure 4.4. It is clear that the solution that used the Whittle-Matérn prior is the superior reconstruction. The correlation between $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}$, the true image, is 0.982. While the Tikhonov solution is able to remove the blur, it performs inpainting poorly since each pixel value is assumed independent of one another due to the identity covariance matrix.

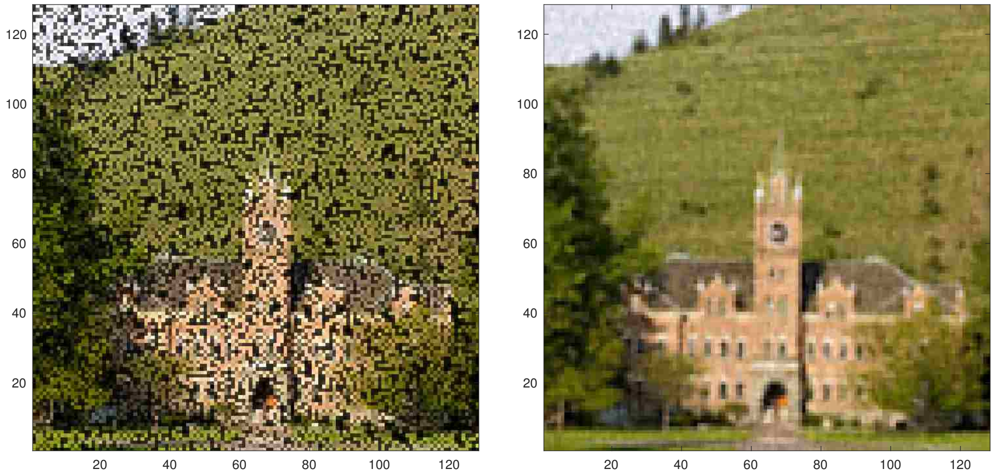

Figure 4.4: Two-dimensional image deblurring test case solutions. On the left is a plot of the Tikhonov solution and on the right is a plot of the solution obtained using the Whittle-Matérn prior with $\nu = 1$ and $\ell = 0.0339, 0.0292$ and 0.0530 for red, green and blue intensities, respectively.

### 4.1.3 Discussion

Compared to other hyperparameter selection methods, the semivariogram method produces competitive solutions and clearer interpretations of the hyperparameters $\nu$ and $\ell$, and it can inform how far to extend the domain to maintain a connection with the Matérn covariance. A shortcoming of the semivariogram method as described in this section, however, is the fact that it requires the field or image to be isotropic. In the next section, we extend these results to anisotropic fields.

## 4.2 Anisotropic Case

When fitting semivariograms to a spatial field, intrinsic stationarity and isotropy are assumed. In our case, we are still assuming intrinsic stationarity, but our field is anisotropic. Thus, a change must be made to our field before fitting a semivariogram to obtain an estimate for $\ell_1$ and $\ell_2$. We again use the change of variables (2.11), the inverse of the change used in (2.10). Using the same argument that was used when transforming the Green's function PDE from anisotropic coordinates in (2.16) to isotropic coordinates in (2.17), it is not difficult to show that (2.12) is transformed to

$$(1 - \ell_1^2 \Delta)^{(\nu + d/2)/2} x(\boldsymbol{u}) = \mathcal{W}(\boldsymbol{u}),$$

which is equivalent to (2.1) with $\ell = \ell_1$.

We can apply this same change of variables (2.11) to any two-dimensional spatial field that exhibits geometric anisotropy to achieve isotropy. For example, if we begin with a spatial field that exhibits its larger correlation length in the 45° direction with $\tau = \ell_1/\ell_2 = 3$, the change of variables will rotate the field so the direction of maximum correlation length is in the 0° direction and will then stretch the field along the new $y$-axis to remove the geometric anisotropy and create a new, isotropic field. This is shown in Figure 4.5. Once the spatial field has been adjusted in this way, a semivariogram can be fit to the transformed field as in the usual, isotropic case.

In order to adjust the spatial field to satisfy the isotropy assumptions in the way described above, we must ascertain $\theta$, the direction of maximum correlation length measured from the $x$-axis, and $\tau$, the ratio of the correlation length in the direction of $\theta$ to the correlation length in the direction orthogonal to $\theta$. Both of these parameters can be estimated using *directional empirical semivariograms*. Directional semivariograms are fit in a similar way as the semivariograms in (4.1), which are also referred to as omnidirectional semivar-

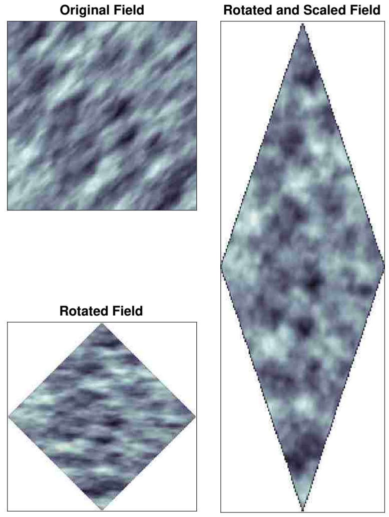

Figure 4.5: Transforming field from anisotropic to isotropic. The top-left image shows the original field that is anisotropic, the bottom-left image shows the rotated field so the direction of maximum anisotropy is aligned with the $x$-axis, and on the right is the rotated and scaled field that is isotropic.

iograms, but instead of taking all points separated by a distance $r$, we restrict the pairs of points to a certain angle, $\psi$. If we think of $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ as vectors, then $\psi$ is equivalent to the angle between $\boldsymbol{w}_i - \boldsymbol{w}_j$ and the $x$-axis. For example, if $\psi = 0$, we restrict to all pairs of locations $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ on the same horizontal line, i.e., with the same $y$-coordinate. Formally, the empirical directional semivariogram can be defined as

$$\hat{\gamma}_\psi(r) = \frac{1}{2n(r,\psi)} \sum_{(i,j)|\|\boldsymbol{w}_i - \boldsymbol{w}_j\| = r\,, \phi_{ij} = \psi} [z(\boldsymbol{w}_i) - z(\boldsymbol{w}_j)]^2, \tag{4.2}$$

where $\phi_{ij}$ denotes the angle between $\boldsymbol{w}_i - \boldsymbol{w}_j$ and the $x$-axis, and $n(r,\psi)$ is the number of points that are separated by a distance $r$ with angle of separation equal to $\psi$. It is common to calculate a directional semivariogram for $-90° < \psi \leq 90°$ in steps of either 15° or 30°. We take a step size of 15° here, which will result in 12 directional semivariograms.

Once the directional semivariograms have been calculated for each of the 12 different $\psi$ angles, we fit a common scatterplot smoother, the loess curve [Jacoby, 2000], to the semivariogram values in each direction to achieve continuous curves. Then, to determine the ratio of correlation lengths, we can select a constant $\gamma_{\text{crit}}$ value between the nugget and sill and observe the distance required for the loess curve to surpass the height of $\gamma_{\text{crit}}$. The direction of maximum correlation, $\theta$, will require a larger distance to reach $\gamma_{\text{crit}}$ than other directions since the variance of the differences between values in that direction is expected to be smaller. The anisotropy ratio, $\tau$, can then be computed as the ratio between the distance in the direction of $\theta$ and the distance in the direction perpendicular to $\theta$.

This process is illustrated in Figure 4.6. The directional semivariograms for the original, anisotropic field from Figure 4.5 are shown on the left. We can see that the correlation length is largest in the 45° direction since the distance of 0.1684 that it takes for the curve to pass $\gamma_{\text{crit}} = 0.9$ is the largest of any direction. The range distance in the $-45°$ direction is 0.0561 and so the ratio of those ranges is $\tau = 0.1684/0.0561 = 3$.

We can then rotate the field clockwise by 45° and stretch it in the direction of the new $y$-axis by a factor of $\tau = 3$ to achieve an isotropic field, as was done in Figure 4.5. The directional semivariograms for the new rotated and scaled field, which is isotropic, are shown on the right in Figure 4.6. It now takes a distance of 0.1684 for the semivariance values to pass $\gamma_{\text{crit}}$ for each $\psi$ angle, which means the ratio has been reduced to one, as it should be for an isotropic field. It is not always the case that we can reduce the ratio of these range values down to one, but we can reduce it enough for the field to be considered

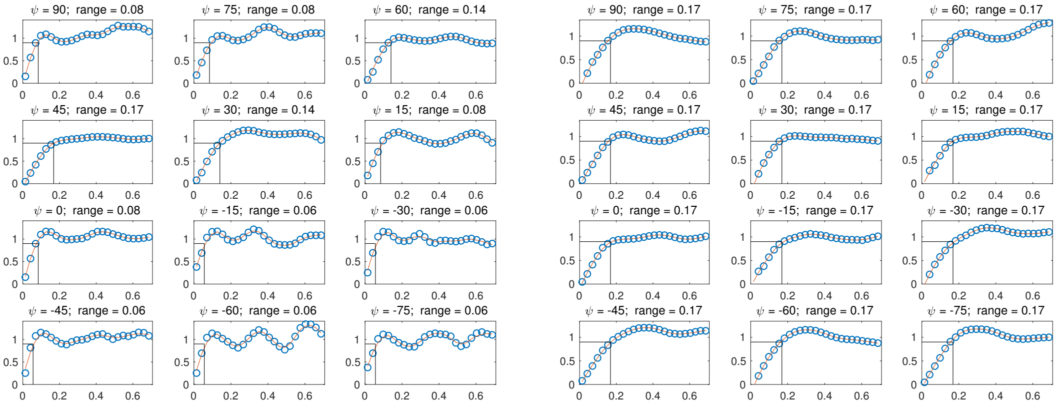

Figure 4.6: Directional semivariograms for fields in Figure 4.5. The 12 directional semivariograms for the original, anisotropic field are shown on the left. For each of the 12 plots, semivariance value is plotted against lag distance. The direction of maximum correlation is determined to be 45° with a ratio of 3 since the distance required to pass $\gamma_{\text{crit}} = 0.9$ was largest in that direction and that distance is 3 times greater than the distance needed in the $-45°$ direction. The 12 directional semivariograms for the rotated and scaled field are shown on the right with a ratio of 1.

approximately isotropic.

In practice, more than one $\gamma_{\text{crit}}$ value is used. We used 10 different $\gamma_{\text{crit}}$ values that varied between 48% below the smallest maximum $\gamma_\psi(r)$ value for the 12 $\psi$ directions to 6% above the smallest maximum $\gamma_\psi(r)$ value. For each of the 10 $\gamma_{\text{crit}}$ values, we found the distance at which the loess curve surpassed $\gamma_{\text{crit}}$ and then took the median of those 10 distances to use for the range.

### 4.2.1 Algorithm

Once we have obtained $\theta$ and $\tau$ and have changed the coordinates of the field, we can fit an isotropic omnidirectional semivariogram to estimate $\nu$ and $\ell_1$.

Then we let $\ell_2 = \ell_1/\tau$. All hyperparameters for use in (3.15) will have been estimated and we can update these estimates iteratively.

**Algorithm:** *The Semivariogram Method for MAP Estimation with Anisotropic Whittle-Matérn Prior:*

0. Set $\boldsymbol{x}_\alpha = \boldsymbol{b}$.
1. Estimate $\theta$ and $\tau$ by computing directional semivariograms for $\boldsymbol{x}_\alpha$.
2. Transform the anisotropic spatial field coordinates, $\boldsymbol{w}$, to isotropic spatial field coordinates, $\boldsymbol{u}$, using (2.11).
3. Estimate $\boldsymbol{\theta} = (a_0, \sigma^2, \nu, \ell_1)$ by fitting an isotropic Matérn semivariogram model to the transformed field. Then compute $\ell_2 = \ell_1/\tau$.
4. Define the prior precision matrix, $\mathbf{P} = \mathbf{P}_{\nu,\ell_1,\ell_2,\theta}$, by (3.16) using $\nu$, $\ell_1$, $\ell_2$, and $\theta$; compute $\alpha$ using (3.20); and compute $\boldsymbol{x}_\alpha$ using (3.19).
5. Return to step 1 and repeat until $\theta$, $\tau$, $\nu$, $\ell_1$, and $\ell_2$ stabilize.

The convergence criteria for these hyperparameters are as follows: $\theta_j - \theta_{j-1} = 0$, $\nu_j - \nu_{j-1} = 0$, $|\ell_1^j - \ell_1^{j-1}|/\ell_1^{j-1} < 0.01$, and $|\ell_2^j - \ell_2^{j-1}|/\ell_2^{j-1} < 0.01$ where $\theta_j, \nu_j, \ell_1^j$ and $\ell_2^j$ denotes the $j$th iteration of the respective hyperparameter.

### 4.2.2 Numerical Experiments

We will illustrate the semivariogram method in the anisotropic case with a two-dimensional inpainting example. The original image, given on the left in Figure 4.7, shows a rock formation in Northern Arizona known as the Wave [Gb11111, ] where the layers of sandstone strata are clearly visible. We selected a subsection in the lower-middle of the image, shown in the middle of Figure

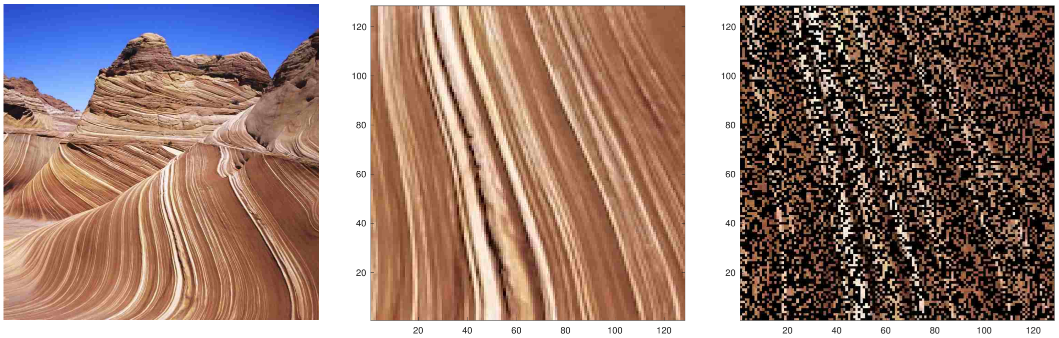

Figure 4.7: Inpainting example. The original image showing the rock layers of the Wave in northern Arizona is given on the left. The true image used in the inpainting example is given in the middle. The masked image is given on the right.

4.7, to illustrate our method. This will be the true image. We then added some noise and masked 60% of the image. This is shown on the right in Figure 4.7.

Like we saw in Section 4.1.2, the prior will play a large role in the inpainting process since much of the image is missing. We will directly compare the solution using the anisotropic Whittle-Matérn prior to the solution using the isotropic Whittle-Matérn prior, both of which will have hyperparameters determined using semivariograms. Like before, the regularization parameter, $\alpha$, will be optimized using the highest correlation between the solution and the true image.

After calculating the directional semivariograms for the image, the direction of maximum correlation was determined to be $-75°$ for each color intensity. For the blue color-band, the correlation length in that direction was $\ell_1 = 0.1517$ and the correlation in the $15°$ direction was $\ell_2 = 0.0101$, which gives a ratio of $\tau = 15$. $\nu$ was determined to be 1, all of these hyperparameters

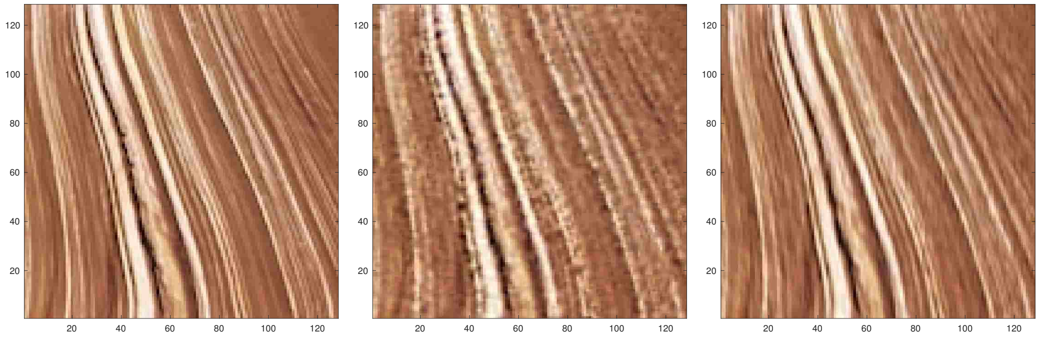

Figure 4.8: Inpainting solutions. The true image (left) is given along with the the isotropic solution (middle) and the anisotropic solution (right).

converged in at most four iterations for each color, and the initial $\theta$ estimate of $-75°$ given in the first iteration remained unchanged throughout the process. When fitting an omnidirectional semivariogram to the masked image for the isotropic case, $\nu = 1$ and $\ell = 0.0141$.

The reconstructions are given in Figure 4.8. With the isotropic solution, the masking is removed, but since the prior assigns a very small correlation between each pixel, the reconstruction is noticeably spotty. The anisotropic solution, however, does a good job of removing the masking completely. The reconstruction is a bit smoother than the true image, but the original sandstone layers can be seen nicely.

Some statistics of the reconstructions are given in Table 4.1. Although the isotropic solution is still relatively competitive, the anisotropic prior gives the reconstruction that most closely aligns with the true image. The isotropic solution has a mean absolute error (MAE) more than 57% larger and a mean squared error (MSE) more than 180% higher than those respective measures in the anisotropic case.

Table 4.1: Statistics for inpainting MAP estimates.

| | True Image | Isotropic Covariance | Anisotropic Covariance |
|---|---|---|---|
| $\bar{x}$ | 0.530 | 0.530 | 0.530 |
| s | 0.207 | 0.202 | 0.206 |
| Min | 0.000 | $-0.053$ | $-0.065$ |
| $Q_1$ | 0.357 | 0.364 | 0.360 |
| Median | 0.522 | 0.520 | 0.520 |
| $Q_3$ | 0.678 | 0.676 | 0.678 |
| Max | 1.000 | 1.112 | 1.093 |
| $\rho_{\boldsymbol{x}_\alpha,\boldsymbol{x}_{\text{true}}}$ | | 0.944 | 0.981 |
| Residual MAE | | 0.045 | 0.029 |
| Residual MSE | | 0.005 | 0.002 |

### 4.2.3 Discussion

Although the reconstruction with the anisotropic prior covariance matrix is better here, there are still some improvements that can be made. This example had a constant angle of maximum correlation length throughout the image and the ratio between maximum and minimum correlation was rather high, that is, greater than five. If either of these features fail to hold, the anisotropic prior often produces a reconstruction that performs slightly worse or offers no benefit over using an isotropic prior. We focus on the case when the angle of maximum anisotropy is not constant in the next chapter.

# Chapter 5

# Extensions

## 5.1 Regional Anisotropy

We have a way to define priors for isotropic and anisotropic spatial fields as long as that covariance structure is consistent for the entire field. In the case where the correlation length or angle of maximum anisotropy change throughout the image, we will want to model each of these regions with a different prior. An example of such a field is given in Figure 5.1. On the left side of the image, the field is isotropic and on the right side it is anisotropic.

### 5.1.1 Preliminary – Cholesky Factorization

The implementation of the method for dealing with regional anisotropy involves the Cholesky factorization or decomposition. The Cholesky factorization provides a way to obtain a unique "square root" of a Hermitian ($\mathbf{A}^* = \mathbf{A}$), positive-definite matrix $\mathbf{A}$ in the sense that it gives a viable, cost effective way of computing $\mathbf{R}_c$ where $\mathbf{A} = \mathbf{R}_c^T \mathbf{R}_c$ [Watkins, 2004]. With this representation of the Cholesky factorization, $\mathbf{R}_c$ is upper triangular (and thus $\mathbf{R}_c^T$ is lower

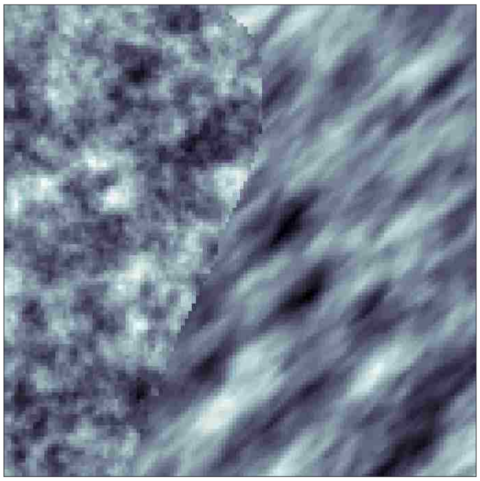

Figure 5.1: Regional field. On the left side of the image, the field is isotropic and on the right side it is anisotropic.

triangular). This decomposition also allows us to solve systems of equations $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ for $\boldsymbol{x}$ very quickly. Since $\mathbf{A} = \mathbf{R}_c^T\mathbf{R}_c$, we can write $\mathbf{R}_c^T\mathbf{R}_c\boldsymbol{x} = \boldsymbol{b}$. Letting $\boldsymbol{y} = \mathbf{R}_c\boldsymbol{x}$, we have $\mathbf{R}_c^T\boldsymbol{y} = \boldsymbol{b}$, which can be solved for $\boldsymbol{y}$ using forward substitution and then $\mathbf{R}_c\boldsymbol{x} = \boldsymbol{y}$ can be solved for $\boldsymbol{x}$ using backward substitution.

To obtain the algorithm for finding all entries of $\mathbf{R}_c$, first we let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad \text{and} \quad \mathbf{R}_c = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \end{bmatrix}.$$

Then we can obtain the diagonal entries of $\mathbf{R}_c$ using

$$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}$$

and the other entries of $\mathbf{R}_c$ can be obtained using

$$r_{ij} = r_{ii}^{-1}\left(a_{ij} - \sum_{k=1}^{i-1} r_{ki}r_{kj}\right), \quad j = i+1, \ldots, n.$$

### 5.1.2 Regional Precision Matrix

Suppose we have $n_r$ different regions in our image, each of which has a different covariance structure. We define $\mathbf{D}_i$, $i = 1, \ldots, n_r$, as a masking matrix such that the only non-zero elements of $\mathbf{D}_i\boldsymbol{x}$ are those in region $i$. We will not allow for overlapping regions so that $\sum_{i=1}^{n_r} \mathbf{D}_{n_r} = \mathbf{I}$, the identity matrix. Now, to establish a prior for $\boldsymbol{x}$ in this regional case, we take $\text{Cov}(\boldsymbol{x}) = \text{Cov}(\mathbf{D}_1\boldsymbol{x} + \cdots + \mathbf{D}_{n_r}\boldsymbol{x}) = \text{Cov}(\mathbf{D}_1\boldsymbol{x}) + \cdots + \text{Cov}(\mathbf{D}_{n_r}\boldsymbol{x})$, since each region is assumed independent due to not having any elements of $\boldsymbol{x}$ in common. Define the best Whittle-Matérn covariance structure, as chosen by a semivariogram, for region $i$ as $\mathbf{C}_i$ with corresponding precision matrix $\mathbf{P}_i = \mathbf{C}_i^{-1}$. Then $\text{Cov}(\mathbf{D}_i\boldsymbol{x}) := \mathbf{D}_i\mathbf{C}_i\mathbf{D}_i$. Thus, the prior for $\boldsymbol{x}$ in this regional case has pdf

$$p(\boldsymbol{x}|\delta) \propto \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T(\mathbf{D}_1\mathbf{C}_1\mathbf{D}_1 + \cdots + \mathbf{D}_{n_r}\mathbf{C}_{n_r}\mathbf{D}_{n_r})^{-1}\boldsymbol{x}\right), \tag{5.1}$$

which means our precision matrix is given by

$$\mathbf{P} = (\mathbf{D}_1\mathbf{C}_1\mathbf{D}_1 + \cdots + \mathbf{D}_{n_r}\mathbf{C}_{n_r}\mathbf{D}_{n_r})^{-1}.$$

Note that (5.1) reduces to (1.4) with $\mathbf{P} = \mathbf{P}_1$ when $n_r = 1$. In general, the $\mathbf{C}_i$ matrices and $\mathbf{P}$ are dense, so actually constructing this precision matrix is infeasible for large problems. Additionally, DFTs cannot be used since $\mathbf{D}_1\mathbf{C}_1\mathbf{D}_1 + \cdots + \mathbf{D}_{n_r}\mathbf{C}_{n_r}\mathbf{D}_{n_r}$ is not circulant even if each $\mathbf{C}_i$ is. Thus, we seek an alternative expression such that matrix-vector multiplication $\mathbf{P}\boldsymbol{v}$ is achievable for a vector $\boldsymbol{v} \in \mathbb{R}^N$.

Without loss of generality, let $n_r = 2$. Define

$$\mathbf{C}_1 = \begin{bmatrix} C_{1A} & C_{1B} \\ C_{1C} & C_{1D} \end{bmatrix} = \mathbf{P}_1^{-1} = \begin{bmatrix} P_{1A} & P_{1B} \\ P_{1C} & P_{1D} \end{bmatrix}^{-1}$$

and

$$\mathbf{C}_2 = \begin{bmatrix} C_{2A} & C_{2B} \\ C_{2C} & C_{2D} \end{bmatrix} = \mathbf{P}_2^{-1} = \begin{bmatrix} P_{2A} & P_{2B} \\ P_{2C} & P_{2D} \end{bmatrix}^{-1}.$$

Also assume that the regions are defined in a way that divides the region vertically (an assumption we will drop later) so that

$$\mathbf{C} = \mathrm{Cov}(\boldsymbol{x}) = \mathrm{Cov}(\mathbf{D}_1\boldsymbol{x} + \mathbf{D}_2\boldsymbol{x}) = \mathrm{Cov}(\mathbf{D}_1\boldsymbol{x}) + \mathrm{Cov}(\mathbf{D}_2\boldsymbol{x}) = \begin{bmatrix} C_{1A} & 0 \\ 0 & C_{2D} \end{bmatrix},$$

which means our precision matrix is

$$\mathbf{P} = \mathbf{C}^{-1} = \begin{bmatrix} C_{1A}^{-1} & 0 \\ 0 & C_{2D}^{-1} \end{bmatrix}.$$

Using the block matrix inversion identity, it can be shown that $C_{1A}^{-1} = P_{1A} - P_{1B}P_{1D}^{-1}P_{1C}$ and $C_{2D}^{-1} = P_{2D} - P_{2C}P_{2A}^{-1}P_{2B}$ and thus

$$\mathbf{P} = \mathbf{C}^{-1} = \begin{bmatrix} C_{1A}^{-1} & 0 \\ 0 & C_{2D}^{-1} \end{bmatrix} = \begin{bmatrix} P_{1A} - P_{1B}P_{1D}^{-1}P_{1C} & 0 \\ 0 & P_{2D} - P_{2C}P_{2A}^{-1}P_{2B} \end{bmatrix},$$

which can be equivalently written as

$$\mathbf{P} = \mathbf{D}_1\mathbf{P}_1\mathbf{D}_1 - \mathbf{D}_1\mathbf{P}_1(\mathbf{D}_2\mathbf{P}_1\mathbf{D}_2)^\dagger\mathbf{P}_1\mathbf{D}_1 + \mathbf{D}_2\mathbf{P}_2\mathbf{D}_2 - \mathbf{D}_2\mathbf{P}_2(\mathbf{D}_1\mathbf{P}_2\mathbf{D}_1)^\dagger\mathbf{P}_2\mathbf{D}_2.$$

In the expression above, '†' denotes the Moore-Penrose inverse, which is also referred to as the generalized inverse or the pseudoinverse, of a matrix [Ben-Israel and Greville, 2003]. In general, for $n_r \geq 2$,

$$\begin{aligned}\mathbf{P} = \mathbf{C}^{-1} &= (\mathbf{D}_1\mathbf{C}_1\mathbf{D}_1 + \mathbf{D}_2\mathbf{C}_2\mathbf{D}_2 + \cdots + \mathbf{D}_{n_r}\mathbf{C}_{n_r}\mathbf{D}_{n_r})^{-1} \\ &= \sum_{i=1}^{n_r}\left(\mathbf{D}_i\mathbf{P}_i\mathbf{D}_i - \mathbf{D}_i\mathbf{P}_i\Big[(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)\Big]^\dagger\mathbf{P}_i\mathbf{D}_i\right), \qquad (5.2)\end{aligned}$$

which, since each $\mathbf{P}_i$ is sparse, involves only sparse matrices.

In the case where the masking matrices $\mathbf{D}_i$, $i = 1, \ldots, n_r$, do not split up the image vertically, we can reorder to achieve a block-diagonal covariance matrix. Let $\tilde{\mathbf{I}}$ be the permutation matrix that corresponds to a reordering of $\boldsymbol{x}$ such that $\tilde{\mathbf{D}}_i = \tilde{\mathbf{I}}\mathbf{D}_i\tilde{\mathbf{I}}^T$ divides $\tilde{\mathbf{I}}\boldsymbol{x}$ vertically (the minimum degree ordering [Tinney and Walker, 1967] will do this). Note that it is a property of permutation matrices that $\tilde{\mathbf{I}}^{-1} = \tilde{\mathbf{I}}^T$. Now, for any matrix $\mathbf{A}$, define $\tilde{\mathbf{A}} := \tilde{\mathbf{I}}\mathbf{A}\tilde{\mathbf{I}}^T$. Then $\tilde{\mathbf{P}} = \tilde{\mathbf{I}}\mathbf{P}\tilde{\mathbf{I}}^T = \tilde{\mathbf{C}}^{-1} = (\tilde{\mathbf{D}}_1\tilde{\mathbf{C}}_1\tilde{\mathbf{D}}_1 + \cdots + \tilde{\mathbf{D}}_{n_r}\tilde{\mathbf{C}}_{n_r}\tilde{\mathbf{D}}_{n_r})^{-1}$. Since this is block-diagonal, we can use the result from above, with $\tilde{\mathbf{P}}_i = \tilde{\mathbf{C}}_i^{-1}$, to say

$$\tilde{\mathbf{P}} = \sum_{i=1}^{n_r}\left(\tilde{\mathbf{D}}_i\tilde{\mathbf{P}}_i\tilde{\mathbf{D}}_i - \tilde{\mathbf{D}}_i\tilde{\mathbf{P}}_i\Big((\mathbf{I}_N - \tilde{\mathbf{D}}_i)\tilde{\mathbf{P}}_i(\mathbf{I}_N - \tilde{\mathbf{D}}_i)\Big)^\dagger\tilde{\mathbf{P}}_i\tilde{\mathbf{D}}_i\right).$$

So, then writing $\tilde{\mathbf{P}} = \tilde{\mathbf{I}}\mathbf{P}\tilde{\mathbf{I}}^T$, we have

$$\tilde{\mathbf{I}}\mathbf{P}\tilde{\mathbf{I}}^T = \sum_{i=1}^{n_r}\left(\tilde{\mathbf{D}}_i\tilde{\mathbf{P}}_i\tilde{\mathbf{D}}_i - \tilde{\mathbf{D}}_i\tilde{\mathbf{P}}_i\Big((\mathbf{I}_N - \tilde{\mathbf{D}}_i)\tilde{\mathbf{P}}_i(\mathbf{I}_N - \tilde{\mathbf{D}}_i)\Big)^\dagger\tilde{\mathbf{P}}_i\tilde{\mathbf{D}}_i\right),$$

which implies

$$\begin{aligned}
\mathbf{P} &= \tilde{\mathbf{I}}^T \left( \sum_{i=1}^{n_r} \left( \tilde{\mathbf{D}}_i \tilde{\mathbf{P}}_i \tilde{\mathbf{D}}_i - \tilde{\mathbf{D}}_i \tilde{\mathbf{P}}_i \Big( (\mathbf{I}_N - \tilde{\mathbf{D}}_i) \tilde{\mathbf{P}}_i (\mathbf{I}_N - \tilde{\mathbf{D}}_i) \Big)^\dagger \tilde{\mathbf{P}}_i \tilde{\mathbf{D}}_i \right) \right) \tilde{\mathbf{I}} \\
&= \sum_{i=1}^{n_r} \left( \tilde{\mathbf{I}}^T \tilde{\mathbf{D}}_i \tilde{\mathbf{P}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{I}} - \tilde{\mathbf{I}}^T \tilde{\mathbf{D}}_i \tilde{\mathbf{P}}_i \Big( (\mathbf{I}_N - \tilde{\mathbf{D}}_i) \tilde{\mathbf{P}}_i (\mathbf{I}_N - \tilde{\mathbf{D}}_i) \Big)^\dagger \tilde{\mathbf{P}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{I}} \right) \\
&= \sum_{i=1}^{n_r} \Bigg( \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \mathbf{D}_i \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \mathbf{P}_i \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \mathbf{D}_i \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} - \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \mathbf{D}_i \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \mathbf{P}_i \tilde{\mathbf{I}}^T \\
&\qquad\qquad \times \Big( (\mathbf{I}_N - \tilde{\mathbf{I}} \mathbf{D}_i \tilde{\mathbf{I}}^T) \tilde{\mathbf{I}} \mathbf{P}_i \tilde{\mathbf{I}}^T (\mathbf{I}_N - \tilde{\mathbf{I}} \mathbf{D}_i \tilde{\mathbf{I}}^T) \Big)^\dagger \tilde{\mathbf{I}} \mathbf{P}_i \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \mathbf{D}_i \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} \Bigg) \\
&= \sum_{i=1}^{n_r} \left( \mathbf{D}_i \mathbf{P}_i \mathbf{D}_i - \mathbf{D}_i \mathbf{P}_i \Big( (\mathbf{I}_N - \mathbf{D}_i) \mathbf{P}_i (\mathbf{I}_N - \mathbf{D}_i) \Big)^\dagger \mathbf{P}_i \mathbf{D}_i \right) \text{ since } \tilde{\mathbf{I}}^T \tilde{\mathbf{I}} = \mathbf{I}_N,
\end{aligned}$$

which is the same expression for $\mathbf{P}$ as in the block-diagonal case. Thus, for any masking matrices $\mathbf{D}_1, \ldots, \mathbf{D}_{n_r}$, so long as $\sum_{i=1}^{n_r} \mathbf{D}_i = \mathbf{I}_N$, this expression for $\mathbf{P}$ holds.

### 5.1.3 Implementation

Since we have an expression for $\mathbf{P}$, we can now discuss how to perform matrix-vector multiplication $\mathbf{P}\boldsymbol{v}$ for some vector $\boldsymbol{v}$. This will be needed to perform an iterative inverse method such as preconditioned conjugate gradient to obtain the MAP estimator. Since each $\mathbf{D}_i$ and $\mathbf{P}_i$ is sparse, each matrix vector multiplication in (5.2) is efficient except the ones involving pseudoinverses. We can, however, take advantage of the lower-rank structure of $\Big[(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)\Big]^\dagger$, which has rank $N - r_i$ where $r_i$ the rank of $\mathbf{D}_i$. Let $\mathbf{P}_{i,\text{nz}}$ be the square matrix that consists of all rows and columns of $(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)$ that have any nonzero elements. That is, keep row and column $j$ of $(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)$ if $[\mathbf{I}_N - \mathbf{D}_i]_{j,j} = 1$. Then let $\mathbf{R}_i = \text{chol}(\mathbf{P}_{i,\text{nz}})$ such that $\mathbf{P}_{i,\text{nz}} = \mathbf{R}_i^T \mathbf{R}_i$ where chol denotes the Cholesky factorization and $\mathbf{R}_i$ is upper triangular. The Cholesky decomposition is known to be efficient for sparse, symmetric, positive definite matrices such as $\mathbf{P}_{i,\text{nz}}$ [Watkins, 2004]. Then we can perform the multiplication of $\mathbf{D}_i\mathbf{P}_i[(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)]^\dagger\mathbf{P}_i\mathbf{D}_i\boldsymbol{v}$ in the following way:

1. Multiply $\boldsymbol{y}_i = \mathbf{P}_i(\mathbf{D}_i\boldsymbol{v})$.

2. Extract the $N - r_i$ elements of $\boldsymbol{y}_i$ that correspond to the nonzero diagonal elements of $\mathbf{I}_N - \mathbf{D}_i$: $\boldsymbol{y}_i(ind)$.

3. Define a variable $\boldsymbol{z}_i$ as an $N \times 1$ vector of zeros.

4. Multiply by $\Big((\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)\Big)^\dagger$ by taking $\boldsymbol{z}_i(ind) = \mathbf{R}_i\backslash(\mathbf{R}_i^T\backslash\boldsymbol{y}_i(ind))$.

5. Complete the multiplication $\boldsymbol{t}_i = \mathbf{D}_i(\mathbf{P}_i\boldsymbol{z}_i)$.

6. Repeat for $1 \leq i \leq n_r$.

Then $\mathbf{P}\boldsymbol{v} = \sum_{i=1}^{n_r}\Big(\mathbf{D}_i\boldsymbol{y}_i - \boldsymbol{t}_i\Big)$.

Step 4 is the most costly since it requires both a forward and a backward substitution. This can be performed more efficiently for large regions since the rank of $(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)$ is inversely related to the size of region $i$. Sparse reorderings, such as the symmetric approximate minimum degree permutation that can be computed using the `symamd` function in MATLAB, can also be used so $\mathbf{R}_i$ has fewer nonzero entries. The multiplication of $\mathbf{P}\boldsymbol{v}$ must be performed for each iteration of PCG, but each $\mathbf{R}_i$ can be stored ahead of time so the Cholesky decompositions need only be performed once.

### 5.1.4 Preconditioner

To speed up the CG convergence, we use the preconditioner $\mathbf{A}^T\mathbf{A} + \frac{\alpha}{n_r}\sum_{i=1}^{n_r}\mathbf{P}_i$, which uses the average of the precision matrices for the $n_r$ regions and is

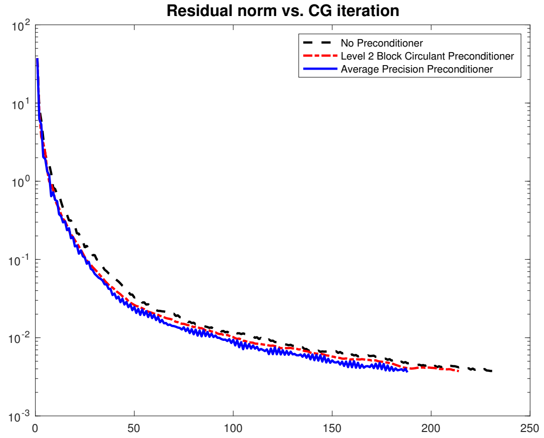

Figure 5.2: CG Iteration History. The CG iteration history is shown for no preconditioner, the level 2 block circulant preconditioner, and the average precision preconditioner. The average precision preconditioner has the fastest convergence rate.

circulant when using periodic boundary conditions, so DFTs can be used. We also tried using the block circulant preconditioner suggested in [Chan and Olkin, 1994], referred to as the Level 2 Block Circulant Preconditioner in [Vogel, 2002], instead of the average of the precision matrices, but performance was slightly worse and it required constructing a new preconditioner. The iteration history results for the two preconditioners, as well as when using no preconditioner, are shown in Figure 5.2. The total number of iterations was approximately 21% lower when using the average precision preconditioner over not using a preconditioner, which decreased computation time by about 15%.

### 5.1.5 Region Selection

Selecting the various regions throughout the image or field is the next point of discussion. For this thesis, a program was written that takes user-defined

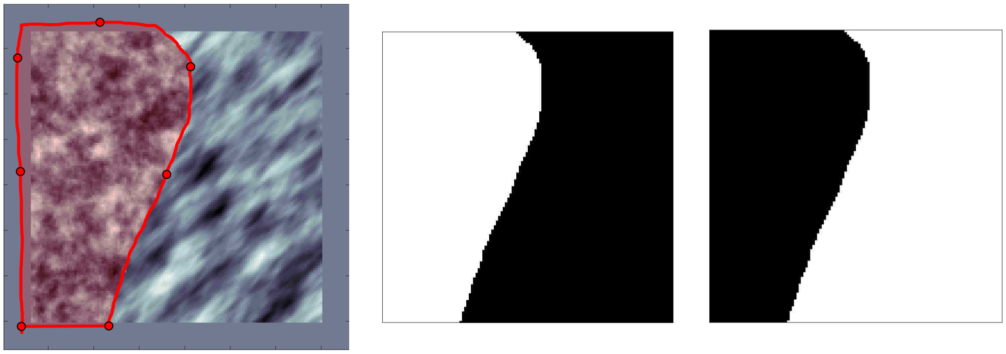

Figure 5.3: Making arrays. The image on the left shows a regional field with regions selected by the user. The first region has the red overlay and the second region is the remainder of the image. The image in the middle is the masking array for region one and shown on the right is the masking array for region two.

regions and creates the masking arrays. With the cursor, the user can draw the borders of each region. For the field shown in Figure 5.1, the user-defined regions are shown on the left in Figure 5.3. The first region is under the red overlay and the second region is the remainder of the image. The two masking arrays are given in the middle and on the right of Figure 5.3. The matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ that restrict only to the elements in region one and two, respectively, can then be obtained by allowing

$$[\mathbf{D}_i]_{jj} = \begin{cases} 1 & \text{if element } j \text{ is in region } i \\ \\ 0 & \text{if element } j \text{ is not in region } i \end{cases}$$

for $i = 1, \ldots, n_r$ and $j = 1, \ldots, N$.

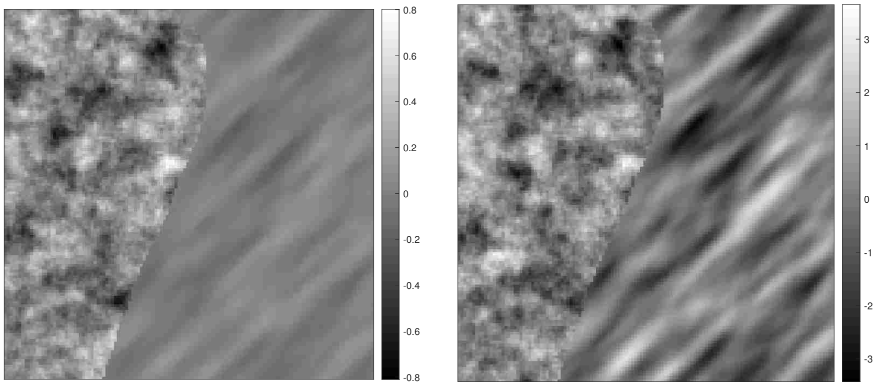

Figure 5.4: Prior weights. The image on the left shows a sample from the prior with unequal variances in the two regions. The image on the right shows a sample from the prior when weighting the regional precision matrices such that the variances in the two regions are approximately equal.

### 5.1.6 Prior Weights

An issue that can arise when defining the prior as we do in this section with $\mathbf{P} = \sum_{i=1}^{n_r} \left( \mathbf{D}_i \mathbf{P}_i \mathbf{D}_i - \mathbf{D}_i \mathbf{P}_i \Big[ (\mathbf{I}_N - \mathbf{D}_i) \mathbf{P}_i (\mathbf{I}_N - \mathbf{D}_i) \Big]^{\dagger} \mathbf{P}_i \mathbf{D}_i \right)$ is the fact that the magnitude of the values in $\mathbf{P}_1, \ldots, \mathbf{P}_{n_r}$ depend on $\nu$ and $\ell$ when the region is isotropic or on $\nu$, $\ell_1$, $\ell_2$, and $\theta$ when the region is anisotropic. Therefore, using a single regularization parameter across each region can assign too much or too little weight to the prior of a region if the variance contribution of each region is substantially different. An example is shown in Figure 5.4. On the left we have a sample from the prior with unequal variances. The total variance of the region on the left is 0.0444 while the variance on the right is approximately only 10% of that value at 0.0044.

To rectify this problem, we will assign weights to the precision matrices in such a way that the variance of all values in each region is approximately equal to one. This is done by taking samples from each region independently, calculating the variance of each region for those samples, and assigning weights as follows:

$$w_i = \frac{s_i^2}{\sum_{j=1}^{n_r} s_j^2}, \quad i = 1, \ldots, n_r,$$

where $s_i^2$ is the empirical variance calculated with all values from region $i$. Then we will multiply each of the regional precision matrices $\mathbf{P}_i$ by $w_i$ so our precision matrix becomes

$$\begin{aligned}\mathbf{P} &= \sum_{i=1}^{n_r} \Big(\mathbf{D}_i(w_i\mathbf{P}_i)\mathbf{D}_i - \mathbf{D}_i(w_i\mathbf{P}_i)\Big[(\mathbf{I}_N - \mathbf{D}_i)(w_i\mathbf{P}_i)(\mathbf{I}_N - \mathbf{D}_i)\Big]^{\dagger}(w_i\mathbf{P}_i)\mathbf{D}_i\Big) \\ &= \sum_{i=1}^{n_r} w_i \Big(\mathbf{D}_i\mathbf{P}_i\mathbf{D}_i - \mathbf{D}_i\mathbf{P}_i\Big[(\mathbf{I}_N - \mathbf{D}_i)\mathbf{P}_i(\mathbf{I}_N - \mathbf{D}_i)\Big]^{\dagger}\mathbf{P}_i\mathbf{D}_i\Big).\end{aligned}$$

Alternatively, one can simply let $\mathbf{P}_i = w_i\mathbf{P}_i$ once the weights have been established. A sample from the prior using these weights is shown on the right in Figure 5.4. The weight for the region on the left is approximately $0.0444/(0.0444 + 0.0044) = 0.9103$ and the weight for the region on the right is $0.0044/(0.0444 + 0.0044) = 0.0897$. Using these weights to generate a new sample led to variances of 1.01 and 0.93 for the left and right region, respectively.

### 5.1.7 Numerical Experiments

We now consider an example where the angle of maximum anisotropy changes throughout the image. We take the central portion of the Wave image from Figure 4.7 and again mask it so that 60% of the image is blank. Then we attempt to inpaint the image using an isotropic prior, an anisotropic prior, and a regional anisotropic prior. The results are shown in Figure 5.5. The top-right image shows the masked picture as well as how the regions were

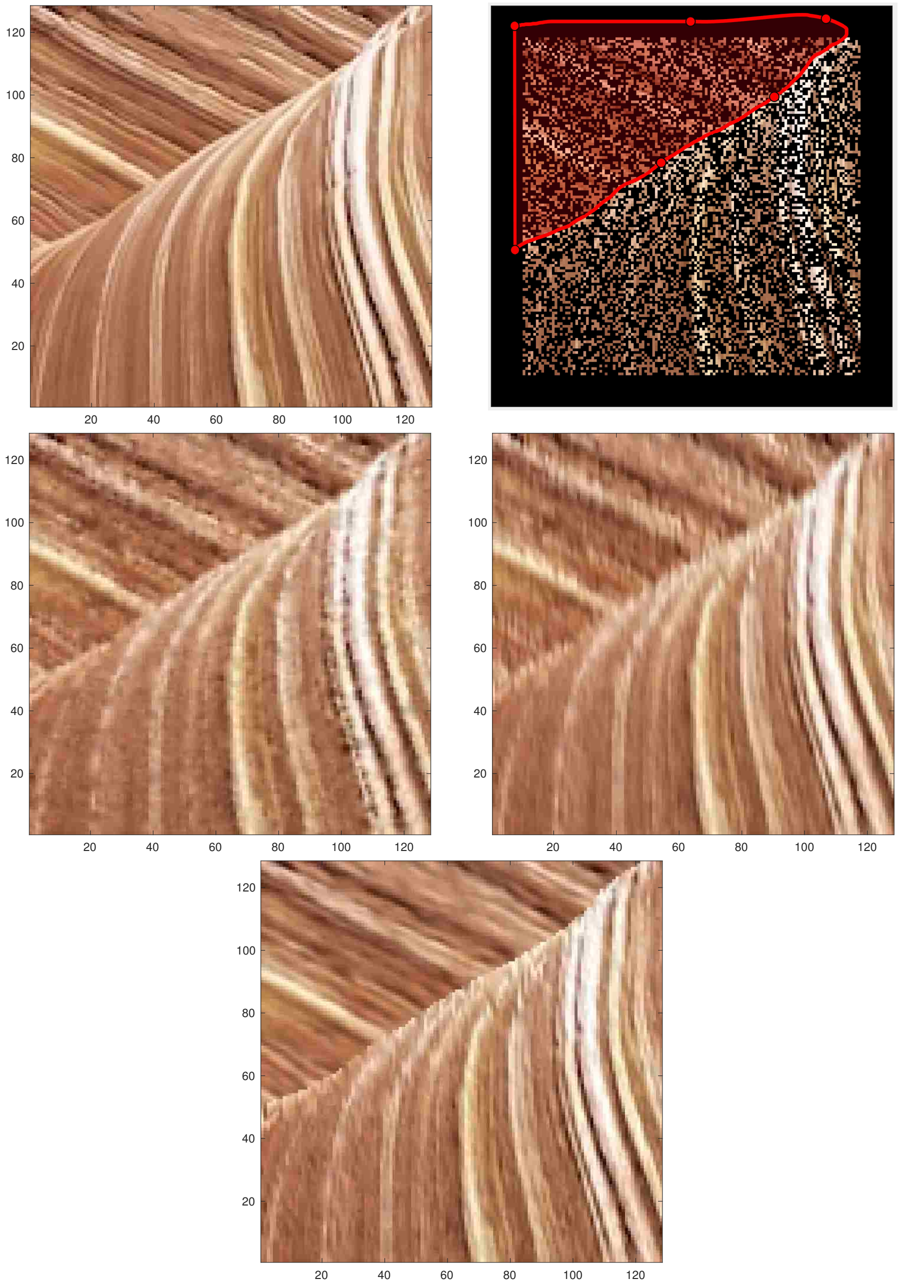

Figure 5.5: Inpainting solutions. The true image (top-left) is given along with the masked image (top-right), the isotropic solution (center-left) the anisotropic solution (center-right), and the regional solution (bottom).

Table 5.1: Statistics for regional inpainting MAP estimates.

| | True Image | Isotropic Covar. | Anisotropic Covar. | Regional Covar. |
|---|---|---|---|---|
| $\bar{x}$ | 0.567 | 0.565 | 0.566 | 0.566 |
| s | 0.207 | 0.200 | 0.202 | 0.206 |
| Min | 0.000 | $-0.014$ | $-0.082$ | $-0.035$ |
| $Q_1$ | 0.400 | 0.402 | 0.402 | 0.398 |
| Median | 0.565 | 0.564 | 0.564 | 0.562 |
| $Q_3$ | 0.722 | 0.717 | 0.718 | 0.720 |
| Max | 1.000 | 1.085 | 1.077 | 1.160 |
| $\rho_{\boldsymbol{x}_\alpha, \boldsymbol{x}_{\text{true}}}$ | | 0.954 | 0.954 | 0.969 |
| Residual MAE | | 0.042 | 0.041 | 0.035 |
| Residual MSE | | 0.004 | 0.004 | 0.003 |

chosen. The first region is shown with the red overlay while the second region is the remainder of the image. Semivariograms were fit to both regions and the top region was given a prior with an angle of maximum anisotropy of $-30°$ while $\theta = -75°$ for the bottom region. In the anisotropic solution given in the bottom-middle of the figure, $\theta = -75°$ throughout the image. Qualitatively, the regional solution in the bottom-right of the figure looks best.

Turning to Table 5.1, we can see the statistics comparing the different reconstructions. The isotropic and anisotropic solutions were similar in terms of the correlation and mean errors, but the regional solution is better in each of those categories and is similar in the others.

### 5.1.8 Discussion

The regional covariance solution performed better in this example, but it does have some shortcomings. Firstly, it is best used when the distinction between regions is high. This is because the transition between regions when using this prior is abrupt, rather than smooth. Smoothing the transition between regions is something we leave to future work. Additionally, since multiplying $\mathbf{P}$ by a vector requires inverting a matrix, this method can be slow when that matrix is large, which corresponds to a small region. Therefore, we suggest using small regions only when necessary. Alternatively, it is possible to solve a different inverse problem for each region independently and then combine the results. This will allow DFTs to be used since the precision matrix for each inverse problem will be in the form of (3.16).

## 5.2 Markov Chain Monte Carlo Methods

In the previous examples, we calculated point estimates of $\nu$ and $\ell$ in the isotropic case or $\nu$, $\ell_1$, $\ell_2$, and $\theta$ in the anisotropic case using semivariograms and suggested estimating $\alpha = \delta/\lambda$ using a regularization method such as GCV. Although these estimates are useful by themselves, an alternative approach that allows for the quantification of uncertainty in these parameter estimates is occasionally desirable.

We will focus here on the isotropic case. Sampling $\lambda, \delta, \nu$, and $\ell$ simultaneously with a fully-Bayesian approach can be done using the Metropolis algorithm or, as is suggested in [Agarwal and Gelfand, 2005], via slice sampling. The problem is that these variables, especially $\nu$ and $\ell$, are so highly correlated that the sampling can be wildly inefficient [Shaby and Wells, 2010] without careful selection of the prior distributions and considerable adjusting of the proposal distributions [Khaledi and Rivaz, 2009]. In our tests, the essential sample size (ESS) [Bardsley, 2018] was more than 100 times smaller than the number of samples generated.

Due to the inefficiency of sampling from this posterior, we will fix $\nu$, as is commonly done [Khaledi and Rivaz, 2009, Roininen et al., 2016], setting it equal to the estimated value determined by the semivariogram method. We now focus on quantifying the uncertainty in $\boldsymbol{\theta} = (\lambda, \delta, \ell)$. To this end, we employ Markov chain Monte Carlo (MCMC) methods to obtain samples from $p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{b})$.

First, we write down the full likelihood function and prior for $\boldsymbol{x}$:

$$p(\boldsymbol{b}|\boldsymbol{x}, \lambda) = \frac{\lambda^{\overline{M}/2}}{(2\pi)^{\overline{M}/2} N^{1/2}} \exp\left(-\frac{\lambda}{2}\|\mathbf{M}\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2\right),$$

where $\overline{M}$ is the rank of $\mathbf{M}$ and

$$p(\boldsymbol{x}|\delta, \nu, \ell) = \frac{\delta^{N/2} \det(\mathbf{P}_{\nu,\ell})^{1/2}}{(2\pi)^{N/2}} \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T \mathbf{P}_{\nu,\ell}\boldsymbol{x}\right),$$

where $\mathbf{P}_{\nu,\ell} = (\mathbf{I} + (\ell/h)^2\mathbf{L})^{\nu+d/2}$. Now we assume $\lambda$ and $\delta$ are random variables with Gamma-distributed hyperpriors, as in [Bardsley, 2018]:

$$p(\lambda) \propto \lambda^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda)$$
$$p(\delta) \propto \delta^{\alpha_\delta - 1} \exp(-\beta_\delta \delta),$$

where $\alpha_\lambda = \alpha_\delta = 1$ and $\beta_\lambda = \beta_\delta = 10^{-4}$. Setting the hyperparameters as we did will lead to the priors being fairly uninformative. That is, the priors will not influence the samples nearly as much as the data will. Gamma hyperpriors lead to Gamma conditional posterior densities for $\lambda$ and $\delta$ (a property known as *conjugacy*), which can be sampled from directly. Since no such conjugacy exists for $\ell$, we assign it a uniform hyperprior, $\ell \sim U(0, \ell_{\max})$, where $\ell_{\max}$ is a selected upper bound for $\ell$. We assign the uniform hyperprior in an effort to be as data-driven as possible. By Bayes' law, the posterior density function is given by

$$\begin{aligned} p(\boldsymbol{x}, &\lambda, \delta, \ell|\boldsymbol{b}, \nu) \propto p(\boldsymbol{b}|\boldsymbol{x}, \lambda)p(\boldsymbol{x}|\delta, \nu, \ell)p(\lambda)p(\delta)p(\ell) \\ &\propto \lambda^{\overline{M}/2+\alpha_\lambda-1}\delta^{N/2+\alpha_\delta-1}\text{det}\,(\mathbf{P}_{\nu,\ell})^{1/2} \\ &\quad\times \exp\left(-\frac{\lambda}{2}\|\mathbf{MA}\boldsymbol{x}-\boldsymbol{b}\|^2 - \frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}_{\nu,\ell}\boldsymbol{x} - \beta_\lambda\lambda - \beta_\delta\delta\right), \quad 0<\ell<\ell_{\max}. \end{aligned}$$

Collecting all of the terms of $p(\boldsymbol{x}, \lambda, \delta, \ell|\boldsymbol{b}, \nu)$ that depend on $\lambda$ and holding all other variables constant yields

$$p(\lambda|\boldsymbol{b}, \boldsymbol{x}, \delta, \nu, \ell) \propto \lambda^{\overline{M}/2+\alpha_\lambda-1}\exp\left(-\lambda\left[\frac{1}{2}\|\mathbf{MA}\boldsymbol{x}-\boldsymbol{b}\|^2+\beta_\lambda\right]\right),$$

which is a Gamma distribution with shape parameter $\overline{M}+\alpha_\lambda$ and rate parameter $\frac{1}{2}\|\mathbf{MA}\boldsymbol{x}-\boldsymbol{b}\|^2+\beta_\lambda$. Doing the same for $\delta$ gives us

$$p(\delta|\boldsymbol{b}, \boldsymbol{x}, \lambda, \nu, \ell) \propto \delta^{N/2+\alpha_\delta-1}\exp\left(-\delta\left[\frac{1}{2}\boldsymbol{x}^T\mathbf{P}_{\nu,\ell}\boldsymbol{x}+\beta_\delta\right]\right),$$

which is another a Gamma distribution. Therefore, the conditional probability distributions for $\lambda$ and $\delta$ are given by

$$\begin{aligned} \lambda|\boldsymbol{b}, \boldsymbol{x}, \delta, \nu, \ell &\sim \Gamma\left(\overline{M}/2+\alpha_\lambda, \frac{1}{2}\|\mathbf{MA}\boldsymbol{x}-\boldsymbol{b}\|^2+\beta_\lambda\right), \\ \delta|\boldsymbol{b}, \boldsymbol{x}, \lambda, \nu, \ell &\sim \Gamma\left(N/2+\alpha_\delta, \frac{1}{2}\boldsymbol{x}^T\mathbf{P}_{\nu,\ell}\boldsymbol{x}+\beta_\delta\right), \end{aligned}$$

and it can be shown that the conditional probability distribution for $\boldsymbol{x}$ is

$$\boldsymbol{x}|\boldsymbol{b}, \lambda, \delta, \nu, \ell \sim \mathcal{N}\Big(\left(\lambda\mathbf{A}^T\mathbf{M}^T\mathbf{MA}+\delta\mathbf{P}_{\nu,\ell}\right)^{-1}\lambda\mathbf{A}^T\mathbf{M}^T\boldsymbol{b}, \left(\lambda\mathbf{A}^T\mathbf{M}^T\mathbf{MA}+\delta\mathbf{P}_{\nu,\ell}\right)^{-1}\Big).$$

When collecting terms for $\ell$, we obtain

$$p(\ell|\boldsymbol{x}, \lambda, \delta, \nu, \boldsymbol{b}) \propto \det\left(\mathbf{P}_{\nu,\ell}\right)^{1/2} \exp\left(-\frac{\delta}{2}\boldsymbol{x}^T\mathbf{P}_{\nu,\ell}\boldsymbol{x}\right), \quad 0 < \ell < \ell_{\max},$$

which is not a known probability distribution.

Since no conjugacy exists for $\ell$, we will use the Metropolis-Hastings (MH) algorithm to sample from $p(\ell|\boldsymbol{x}, \lambda, \delta, \nu, \boldsymbol{b})$. MH is an iterative method in which each sample is dependent on the last. To implement MH, we randomly generate a proposed sample, $\ell^*$, from a proposal density, $q$. To keep all $\ell^*$ values positive, we use a log-normal proposal density, i.e.,

$$q(\ell|\ell_k) \propto \frac{1}{\ell}\exp\left(-\frac{1}{2\rho^2}(\ln \ell - \ln \ell_k)^2\right), \tag{5.3}$$

where $\rho^2$ is chosen offline to keep the acceptance rate at an acceptable level, which is often said to be between 0.2 and 0.5 [Gelman et al., 1996]. A proposed sample $\ell^* \sim q(\ell|\ell_k)$ is then accepted with probability

$$\begin{aligned}\beta(\ell^*, \ell_k) &= \min\left\{1, \frac{p(\ell^*|\boldsymbol{x}^k, \lambda_{k+1}, \delta_{k+1}, \nu, \boldsymbol{b})q(\ell_k|\ell^*)}{p(\ell_k|\boldsymbol{x}^k, \lambda_{k+1}, \delta_{k+1}, \nu, \boldsymbol{b})q(\ell^*|\ell_k)}\right\} \\ &= \min\left\{1, \frac{\ell^*\det\left(\mathbf{P}_{\nu,\ell^*}\right)^{1/2}\exp\left(-\frac{\delta}{2}(\boldsymbol{x}^k)^T\mathbf{P}_{\nu,\ell^*}\boldsymbol{x}^k\right)}{\ell_k\det\left(\mathbf{P}_{\nu,\ell_k}\right)^{1/2}\exp\left(-\frac{\delta_{k+1}}{2}(\boldsymbol{x}^k)^T\mathbf{P}_{\nu,\ell_k}\boldsymbol{x}^k\right)}\right\}.\end{aligned}$$

That is, we set $\ell_{k+1} = \ell^*$ with probability $\beta(\ell^*, \ell_k)$, and otherwise we set $\ell_{k+1} = \ell_k$. To obtain samples of $\boldsymbol{x}, \lambda, \delta$, and $\ell$, we use the following Metropolis-within-Gibbs sampler [Metropolis et al., 1953, Tierney, 1994]. Note that we denote a sample of $\boldsymbol{x}$ as $\boldsymbol{x}^k$ rather than $\boldsymbol{x}_k$ to differentiate it from an iteration of the CG or PCG methods.

*The Metropolis-within-Gibbs Sampler for Posterior Sampling*

0. Initialize $\lambda_0, \delta_0$, and $\ell_0$ and set $k = 0$;

1. Compute a sample

$$\boldsymbol{x}^k \sim \mathcal{N}\left(\left(\lambda_k \mathbf{A}^T \mathbf{M}^T \mathbf{M} \mathbf{A} + \delta_k \mathbf{P}_{\nu,\ell_k}\right)^{-1} \lambda_k \mathbf{A}^T \mathbf{M}^T \boldsymbol{b}, \left(\lambda_k \mathbf{A}^T \mathbf{M}^T \mathbf{M} \mathbf{A} + \delta_k \mathbf{P}_{\nu,\ell_k}\right)^{-1}\right).$$

2. Compute samples

$$\begin{aligned} \lambda_{k+1} &\sim \Gamma\left(\overline{M}/2 + \alpha_\lambda, \frac{1}{2}\|\mathbf{M}\mathbf{A}\boldsymbol{x}^k - \boldsymbol{b}\|^2 + \beta_\lambda\right); \\ \delta_{k+1} &\sim \Gamma\left(N/2 + \alpha_\delta, \frac{1}{2}(\boldsymbol{x}^k)^T \mathbf{P}_{\nu,\ell} \boldsymbol{x}^k + \beta_\delta\right). \end{aligned}$$

3. Apply the Metropolis-Hastings algorithm to compute a sample

$$\ell_{k+1} \sim p(\ell | \boldsymbol{x}^k, \lambda_{k+1}, \delta_{k+1}, \nu, \boldsymbol{b}).$$

4. Set $k = k + 1$ and return to Step 1.

The $\lambda$ and $\delta$ samples can be directly computed in both one and two dimensions, but we need to apply PCG to obtain a sample of $\boldsymbol{x}$ in two-dimensional cases [Bardsley and Luttman, 2015].

### 5.2.1 MCMC Diagnostics

#### Sampling Chains

The primary diagnostic tool for evaluating MCMC sampling is observing the sampling chains. Ideally, each sample in the chain will be independent from all

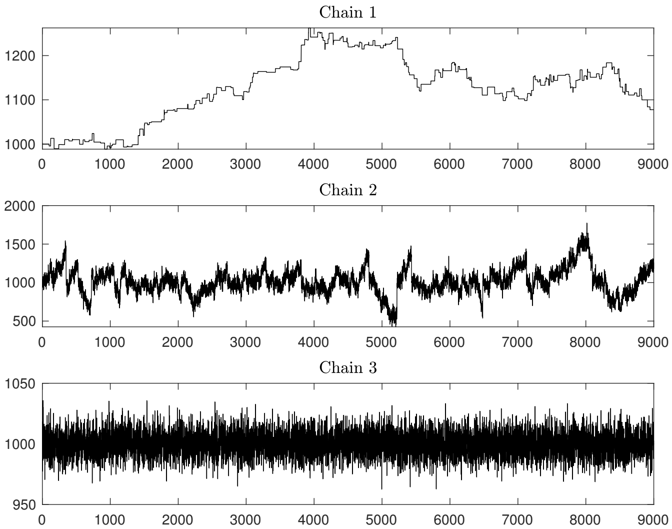

Figure 5.6: MCMC chain examples. Chain 1 on top has a low acceptance rate. Chain 2 in the middle is dependent on many previous iterations. Chain 3 on the bottom exhibits good behavior.

others. Since the posterior distributions rely on the previous iteration, that is not going to be the case, but we would like to minimize the number of previous iterations on which the current one depends. A good-looking sampling chain is one that essentially looks like noise, with no discernible patterns or paths. Figure 5.6 shows three different types of sampling chains. The chain on top is one that has a low acceptance rate, and therefore remains at specific values for many iterations. The chain in the middle is moving, but is correlated with many previous iterations. The chain on the bottom exhibits good behavior and is dependent on very few previous iterations.

### Autocorrelation Functions

The autocorrelation function (ACF) gives an idea of how correlated a chain is with itself at various lags [Venables and Ripley, 2001]. For a sampling chain

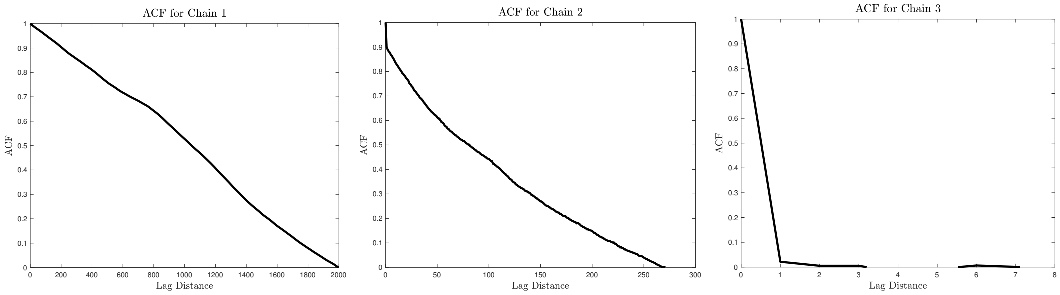

Figure 5.7: MCMC ACF examples. The ACFs for the top, middle, and bottom chains in Figure 5.6 are given on the left, middle, and right, respectively.

$\{y_k\}_{k=1}^K$, one can calculate the ACF using

$$\hat{\rho}(j) = C(j)/C(0), \quad \text{where} \quad C(j) = \frac{1}{K}\sum_{k=1}^{K-j}(y_k - \bar{y})(y_{k+|j|} - \bar{y}),$$

and $K$ is the number of MCMC samples in the chain. Note that we divide by $K$ and not $K-j$ in $C(j)$ in order to keep the $C(j)$ covariance sequence positive definite. The ACFs for the top, middle, and bottom chains in Figure 5.6 are given in Figure 5.7 on the left, middle, and right, respectively. The ACFs show that Chain 1 is autocorrelated out to a lag of around 2,000, Chain 2 is autocorrelated to a lag of approximately 270, and Chain 3 is only dependent on the last iteration.

### Integrated Autocorrelation Time

The integrated autocorrelation time (IACT) estimates the number of iterations needed, on average, for an independent sample to be drawn [Fang et al., 2017]. This is denoted by $\tau_{\text{int}}(y)$. There are various ways to calculate this statistic. We use Sokal's adaptive truncated periodogram estimator presented in [Sokal, 1997]. The calculation is given by

$$\hat{\tau}_{\text{int}} = 1 + 2\sum_{j=1}^{\overline{K}} \frac{K}{K-j}\hat{\rho}(j),$$

where $\overline{K}$ is taken to be the smallest integer such that such that $\overline{K} \geq 3\hat{\tau}_{\text{int}}$. Since the value of $\overline{K}$ depends on $\hat{\tau}_{\text{int}}$, it is usually necessary to compute this statistics multiple times for various $\overline{K}$ values before obtaining one that fits the required criteria.

Since $\hat{\tau}_{\text{int}}$ gives the estimated number of iterations needed to obtain an independent sample, one can calculate the *essential sample size* (ESS) by dividing the number of iterations in the chain by $\hat{\tau}_{\text{int}}$:

$$K_{\text{ESS}} = K/\hat{\tau}_{\text{int}}.$$

This measure gives a good idea of the number of independent samples in the chain. The IACTs for the chains in Figure 5.6 are 1,144, 154.4, and 1.065 for the top, middle, and bottom chains, respectively. That corresponds to an ESS of 7.87, 58.3, and 8,451 in a chain length of 9,000. Notice that the IACT does not calculate the lag distance needed for the autocorrelation to be equal to zero. Some autocorrelation is acceptable even in samples that are said to be independent.

### Geweke Test

The final MCMC diagnositc we will discuss is the Geweke Test for equality in means that can be used to identify if the sampling chain is in equilibrium

[Geweke, 1992]. The Geweke test performs the following hypothesis test:

$$H_0 : \mu_a = \mu_b$$
$$H_a : \mu_a \neq \mu_b,$$

where $\mu_a$ is the true mean of the first $a$% of the sampling chain and $\mu_b$ is the mean of the last $b$% of the chain. We use $a = 10$ and $b = 50$ so that we test if the mean of the first 10% of the sampling chain is equal to the last 50%. Then the test statistic

$$R_{\text{Geweke}} = \frac{\bar{y}_a - \bar{y}_b}{\sqrt{\frac{\hat{S}_a}{K_a} + \frac{\hat{S}_b}{K_b}}}$$

can be used, where $\bar{y}_a$ is the sample mean, $K_a$ is the sample size, and $\hat{S}_a$ is consistent spectral density estimates for the variance of the first $a$% of the sampling chain; and $\bar{y}_b$ is the sample mean, $K_b$ is the sample size, and $\hat{S}_b$ is consistent spectral density estimates for the variance of the last $b$% of the sampling chain. The spectral density estimates for the variances are used since the sample variances are not independent. These estimates will not be discussed here but can be found in [Geweke, 1992]. When the number of samples in the chain is large, this test can be approximated using the Central Limit Theorem. That is, $R_{\text{Geweke}} \overset{dist}{\to} \mathcal{N}(0, 1)$ as $K \to \infty$. A large p-value (greater than 0.95) indicates strong evidence in favor of the null hypothesis that $\mu_a = \mu_b$ and, therefore, that the chain is stable.

### 5.2.2 MCMC Results

We implement the MCMC procedure described above for the two-dimensional example from Section 4.1.2. For the log-normal proposal (5.3), we use the

Table 5.2: MCMC results – Red color band.

| Hyper-Parameter | Mean | 95% Credible Interval | ESS | Geweke p-value |
|---|---|---|---|---|
| $\lambda$ | 1,122 | (1,020, 1,229) | 152 | 0.98 |
| $\delta$ | 0.0202 | (0.014, 0.027) | 524 | 0.99 |
| $\alpha = \delta/\lambda$ | $1.80 \times 10^{-5}$ | $(1.28, 2.39) \times 10^{-5}$ | 800 | 0.99 |
| $\ell$ | 0.0228 | (0.021, 0.025) | 313 | 0.99 |

variance $\rho^2 = 0.025$, which gave an acceptance rate of approximately 0.40 for $\ell$. We compute 10,000 samples and discard the first 1,000 as burn-in. The results for the red intensities are given in Figure 5.8 and in Table 5.2. We omit the MCMC chains for the green and blue intensities, but the results are similar except for the fact that the IACTs are larger for the blue color band. The high p-values for the Geweke tests indicate that the chains are well mixed and have reach stability. Although the IACT values are high, the chains and histograms otherwise look good as does the reconstruction given in the bottom-right of Figure 5.8. The middle-right image shows the difference between the 97.5th and 2.5th percentile for every pixel. The pixels with larger uncertainty correspond to the ones that were missing in the masked image.

The 95% credible intervals for $\lambda$, $\delta$, $\alpha$, and $\ell$ are given for each of the color arrays in Table 5.3. The $\alpha$ values for the red and green intensities are nearly identical, while the $\alpha$ values for the blue color band are a bit larger. The $\alpha$ values of $2.24 \times 10^{-6}$, $3.84 \times 10^{-6}$ and $1.27 \times 10^{-7}$ that we obtained using

Table 5.3: MCMC 95% credible intervals.

| Hyper-Parameter | Red Intensity | Green Intensity | Blue Intensity |
|---|---|---|---|
| $\lambda$ | (1,020, 1,229) | (1,071, 1,278) | (2,202, 2,960) |
| $\delta$ | (0.014, 0.027) | (0.015, 0.029) | (0.072, 0.119) |
| $\alpha = \delta/\lambda$ ($\times 10^{-5}$) | (1.28, 2.39) | (1.29, 2.47) | (2.84, 4.73) |
| $\ell$ | (0.021, 0.025) | ( 0.022, 0.026) | (0.015. 0.017) |

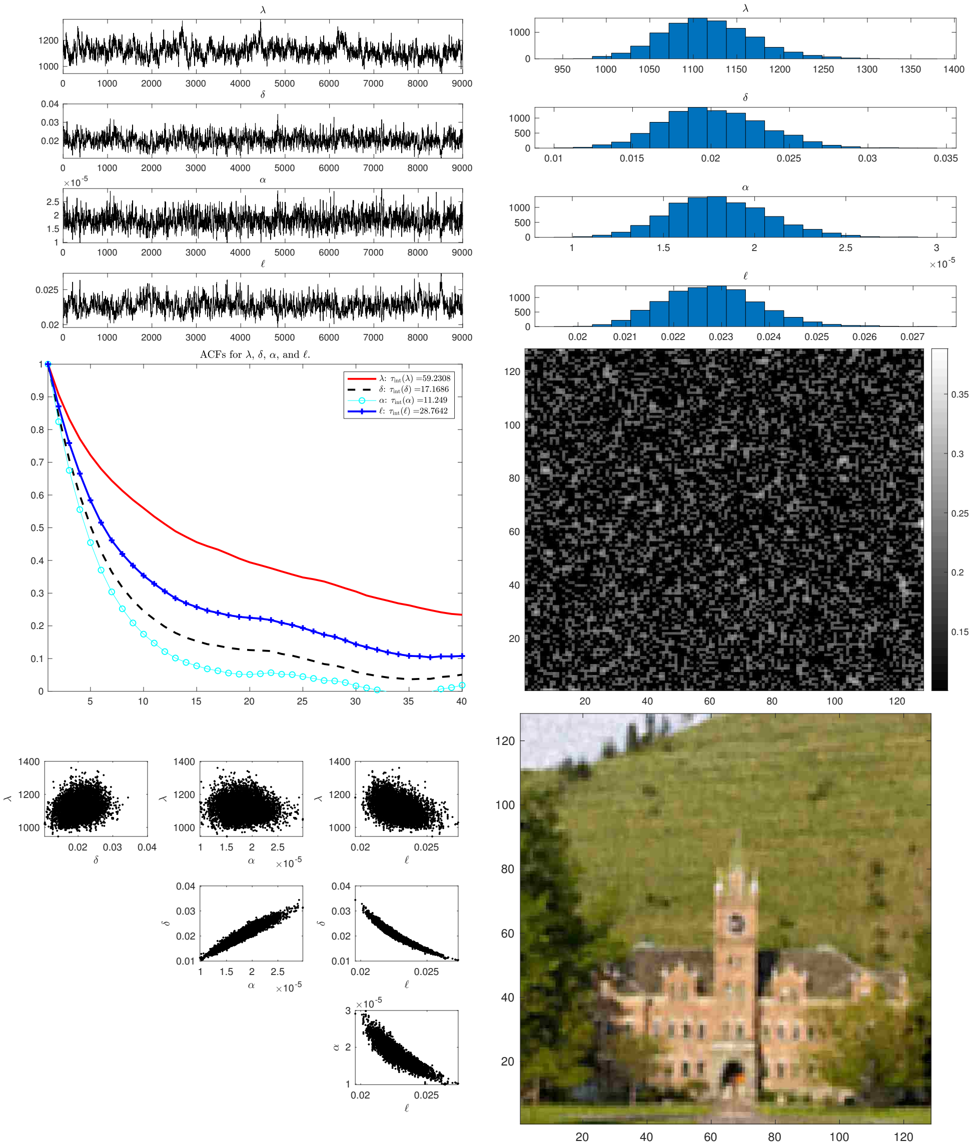

Figure 5.8: 2-D MCMC results. The upper-left shows the $\lambda$, $\delta$, $\alpha = \delta/\lambda$, and $\ell$ sampling chains for the red intensity. The upper-right shows the $\lambda$, $\delta$, $\alpha = \delta/\lambda$, and $\ell$ histograms for the red intensity. The middle-left shows the autocorrelation functions for $\lambda$, $\delta$, $\alpha = \delta/\lambda$, and $\ell$. The middle-right shows the difference between the 97.5th and 2.5th percentiles for each pixel in the image. The bottom-left image shows pair-wise scatter plots. The MCMC solution taken as the mean of the sampling chain of $\boldsymbol{x}$ is on the bottom right.

Table 5.4: Statistics for MCMC and MAP estimates.

| | True Image | MCMC | Whittle-Matérn Covariance |
|---|---|---|---|
| $\bar{x}$ | 0.418 | 0.418 | 0.418 |
| s | 0.216 | 0.211 | 0.213 |
| Min | 0.000 | $-0.057$ | $-0.062$ |
| $Q_1$ | 0.243 | 0.246 | 0.245 |
| Median | 0.455 | 0.454 | 0.452 |
| $Q_3$ | 0.545 | 0.545 | 0.546 |
| Max | 1.000 | 1.014 | 1.027 |
| $\rho_{\boldsymbol{x}_{\text{estimate}},\boldsymbol{x}_{\text{true}}}$ | | 0.981 | 0.982 |
| Residual MAE | | 0.029 | 0.029 |
| Residual MSE | | 0.002 | 0.002 |

the best correlation criteria from Section 4.1.2 fall below of the intervals for all three color bands, which contributed to a slightly smoother image in the MCMC solution.

Figure 5.9 contains additional images to make a direct visual comparison between the true image, the masked image, and the reconstructions from the different methods. Although the solutions look similar, the semivariogram image is slightly sharper, while the MCMC image is smoother, which especially looks a bit better in the grassy regions. Table 5.4 contains a numerical comparison between the two solutions. These measures indicate that while the two methods give different reconstructions, they are very similar and both appear to be good.

### 5.2.3 Discussion

Although the MCMC method provided a good solution and gave uncertainty quantification for the model parameters, the semivariogram technique still produced a competitive solution and has some practical advantages: the semi-

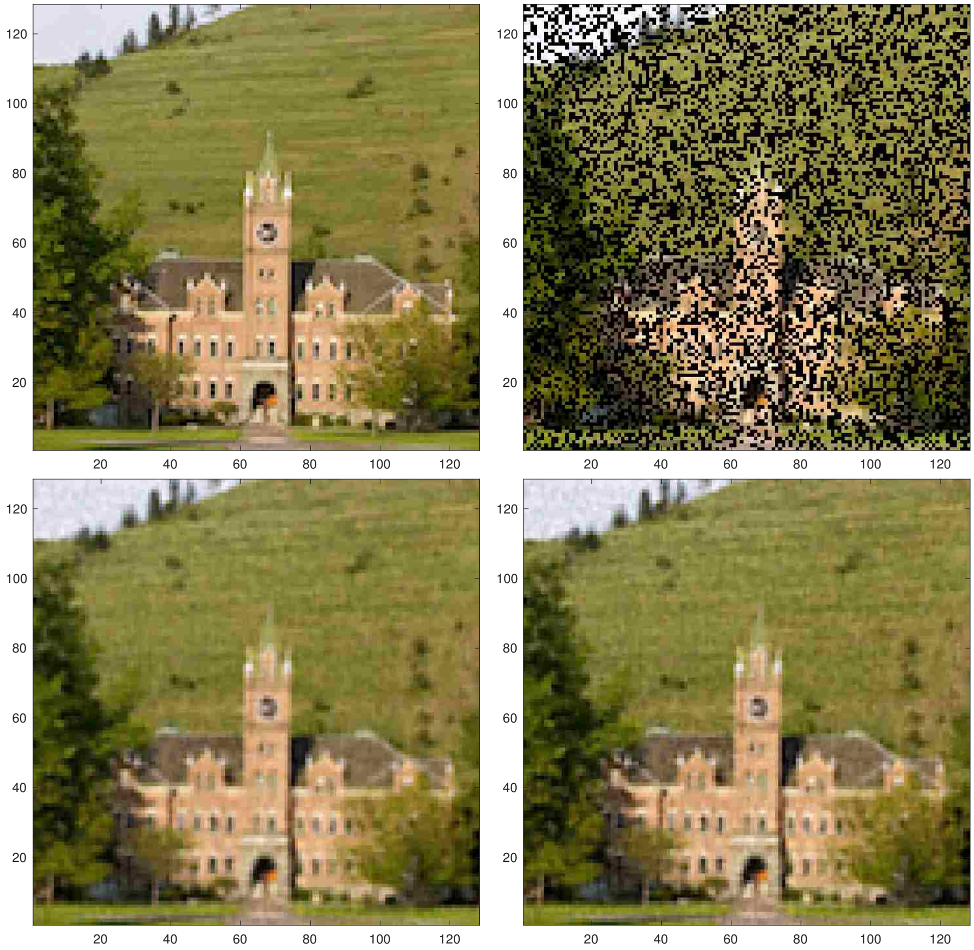

Figure 5.9: 2-D deblurring images. The upper-left plot shows the true Image; the upper-right shows the masked image; the lower-left image shows the MCMC sample mean reconstruction fixing $\nu$; and the lower-right image shows the MAP reconstruction using variograms.

variogram procedure provides clearer interpretations of the parameters $\nu$ and $\ell$; it gives an objective, intuitive way to select $\nu$ for use in the MCMC; and it can inform how far to extend the domain to maintain a connection with the Matérn covariance. Additionally, the computation time is only a fraction of what is needed to compute an adequate number of MCMC samples. In our implementation of the example above, the semivariogram method was approximately 80 times faster than the MCMC approach.

Extending the MCMC methods to the anisotropic case can be accomplished in one of two ways. First, one can assign hyperprior distributions to $\ell_1$, $\ell_2$, and $\theta$ and then use the prior (3.15) and Bayes' law to achieve a new Metropolis-within-Gibbs sampler. With that many hyperparameters, however, convergence may be difficult to achieve. Alternatively, it may be possible to transform the image to an isotropic one before applying MCMC methods using (2.10) and then transform back using (2.11) once the sampling is complete. When doing this, however, DFTs may not be able to be used to obtain a sample of $\boldsymbol{x}^k$ since the image may no longer be on a grid. Applying MCMC to the anisotropic case is something we leave to future work.

# Chapter 6

# Conclusion

In this thesis, we introduced a method for selecting hyperparameters for use in the prior distribution of $\boldsymbol{x}$ based on semivariogram modeling. We think of the noisy data as a spatial field and fit semivariograms to the noisy data and then iteratively to the MAP estimates to obtain point estimates for the prior hyperparameters. This method relies on the fact that the solution of the SPDE (2.1) is a Gaussian process with zero mean and Matérn covariance operator, which we have shown in detail. However, this connection requires an infinite domain. For a finite domain, which is typically required for computations, the connection is broken, i.e., the SPDE solution is a zero mean Gaussian process without a Matérn covariance operator. Fortunately, the connection can be restored by extending the finite computational domain. We showed how to systematically choose the extended domain using the Matérn parameters.

We generalized the isotropic results to the anisotropic case and showed the semivariogram method can be applied as well by using directional semivariograms and the anisotropic SPDE (2.12). An inpainting example comparing reconstructions using isotropic and anisotropic priors was presented. We also extended the results to an even more general case when the image has regions with differing correlation lengths or angles of maximum correlation, which requires a sparse precision matrix that can be obtained via a discretized SPDE for each region. One more example was shown that yielded good solutions. Additionally, we compared the results of the semivariogram method to the fully-Bayesian method using MCMC. The solutions were similar and while uncertainty quantification is lost, the semivariogram method has the benefits of giving point estimates with a more intuitive interpretation while providing an objective way to choose an extension of the computational domain that is adequate for restoring the SPDE/Matérn connection.

Future work related to this topic could focus on incorporating MCMC methods to the anisotropic case to quantify uncertainty in the hyperparameter estimates. Also, work to smooth out the transition between regions in the regional case could be done to increase its usefulness for practical problems. This may be accomplished by dropping the assumption of independence between regions as well as allowing for regions to overlap. Another topic of study is to investigate whether it is possible to use semivariograms to estimate the regularization parameter so that reliance on a regularization parameter selection technique like GCV is not required. Finally, applying spatial statistical methods for estimating hyperparameters when the assumption of stationarity in the field is dropped can be investigated.

## 6.1 MATLAB Code

All of the methods discussed in this dissertation where coded in MATLAB [MAT, 2019]. We used MATLAB R2019a, but most of the code will work in older versions. For the function that will not work in an earlier version, an alternate is provided. All code can be accessed at the following URL: https://github.com/rbrown53/DissertationCodes.

# Bibliography

[MAT, 2019] (2019). *MATLAB version 9.6.0.1099231 (R2019a).* The Mathworks, Inc., Natick, Massachusetts.

[Agarwal and Gelfand, 2005] Agarwal, D. K. and Gelfand, A. E. (2005). Slice sampling for simulation based fitting of spatial data models. *Statistics and Computing*, 15(1):61–69.

[Andrews, 1992] Andrews, L. C. (1992). *Special Functions of Mathematics for Engineers.* McGraw-Hill New York.

[Bak et al., 2010] Bak, J., Newman, D. J., and Newman, D. J. (2010). *Complex Analysis.* Springer.

[Bardsley, 2018] Bardsley, J. M. (2018). *Computational Uncertainty Quantification for Inverse Problems.* SIAM.

[Bardsley and Luttman, 2015] Bardsley, J. M. and Luttman, A. (2015). Dealing with boundary artifacts in MCMC-based deconvolution. *Linear Algebra and Its Applications*, 473:339–358.

[Bateman, 1954] Bateman, H. (1954). *Tables of Integral Transforms [Volumes I & II].* McGraw-Hill.

[Ben-Israel and Greville, 2003] Ben-Israel, A. and Greville, T. N. (2003). *Generalized Inverses: Theory and Applications*, volume 15. Springer Science & Business Media.

[Bracewell, 1986] Bracewell, R. N. (1986). *The Fourier Transform and its Applications*, volume 31999. McGraw-Hill New York.

[Chan and Olkin, 1994] Chan, T. F. and Olkin, J. A. (1994). Circulant preconditioners for toeplitz-block matrices. *Numerical Algorithms*, 6(1):89–101.

[Cressie, 2015] Cressie, N. (2015). *Statistics for Spatial Data*. John Wiley & Sons.

[D'Angelo, 2013] D'Angelo, J. P. (2013). *Hermitian Analysis: From Fourier Series to Cauchy-Riemann Geometry*. Birkhäuser Basel.

[Dietrich and Newsam, 1997] Dietrich, C. R. and Newsam, G. N. (1997). Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107.

[Fang et al., 2017] Fang, Y., Cao, Y., and Skeel, R. D. (2017). Quasi-reliable estimates of effective sample size. *arXiv preprint arXiv:1705.03831*.

[Gb11111, ] Gb11111. Arizona – the wave. Flickr. Retrieved 22 May 2019.

[Gelman et al., 1996] Gelman, A., Roberts, G., and Gilks, W. (1996). Bayesian statistics 5. *Oxford University Press*, 16:17.

[Geweke, 1992] Geweke, J. (1992). Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. In *Bayesian Statistics*, pages 169–193. University Press.

[Gneiting et al., 2010] Gneiting, T., Kleiber, W., and Schlather, M. (2010). Matérn Cross-Covariance Functions for Multivariate Random Fields. *Journal of the American Statistical Association*, 105(491):1167–1177.

[Gockenbach, 2005] Gockenbach, M. S. (2005). *Partial Differential Equations: Analytical and Numerical Methods*, volume 122. SIAM.

[Guttorp and Gneiting, 2006] Guttorp, P. and Gneiting, T. (2006). Studies in the history of probability and statistics XLIX On the Matérn correlation family. *Biometrika*, 93(4):989–995.

[Hale, 2013] Hale, D. (2013). Implementing an anisotropic and spatially varying Matérn model covariance with smoothing filters.

[Hansen, 2005] Hansen, P. C. (2005). *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, volume 4. Siam.

[Hansen et al., 2006] Hansen, P. C., Nagy, J. G., and O'Leary, D. P. (2006). *Deburring Images: Matrices, Spectra, and Filtering.* Society for Industrial and Applied Mathematics.

[Haskard, 2007] Haskard, K. A. (2007). *An anisotropic Matérn spatial covariance model: REML estimation and properties.* PhD thesis, University of Adelaide.

[Hassani, 2000] Hassani, S. (2000). Dirac Delta Function. In *Mathematical Methods*, pages 289–319. Springer.

[Jacoby, 2000] Jacoby, W. G. (2000). Loess: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, 19(4):577–613.

[Kaipio and Somersalo, 2005] Kaipio, J. and Somersalo, E. (2005). *Statistical and Computational Methods for Inverse Problems*. Springer.

[Khaledi and Rivaz, 2009] Khaledi, M. J. and Rivaz, F. (2009). Empirical Bayes spatial prediction using a Monte Carlo EM algorithm. *Statistical Methods and Applications*, 18(1):35–47.

[Khristenko et al., 2019] Khristenko, U., Scarabosio, L., Swierczynski, P., Ullmann, E., and Wohlmuth, B. (2019). Analysis of Boundary Effects on PDE-Based Sampling of Whittle–Matérn Random Fields. *SIAM/ASA Journal on Uncertainty Quantification*, 7(3):948–974.

[Kwaśnicki, 2017] Kwaśnicki, M. (2017). Ten equivalent definitions of the fractional Laplace operator. *Fractional Calculus and Applied Analysis*, 20(1):7–51.

[Lindgren et al., 2011] 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.

[Lord et al., 2014] Lord, G. J., Powell, C. E., and Shardlow, T. (2014). *An Introduction to Computational Stochastic PDEs*. Number 50 in Cambridge texts in applied mathematics. Cambridge University Press.

[Matérn, 2013] Matérn, B. (2013). *Spatial Variation*, volume 36. Springer Science & Business Media.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

[Minasny and McBratney, 2005] Minasny, B. and McBratney, A. B. (2005). The Matérn function as a general model for soil variograms. *Geoderma*, 128(3-4):192–207.

[Monterrubio-Gómez et al., 2018] Monterrubio-Gómez, K., Roininen, L., Wade, S., Damoulas, T., and Girolami, M. (2018). Posterior Inference for Sparse Hierarchical Non-stationary Models.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media.

[Piessens, 2000] Piessens, R. (2000). The Hankel Transform. In Poularikas, A. D., editor, *Transforms and Applications Handbook*, chapter 9. CRC Press, Boca Raton, FL.

[Robert and Casella, 2013] Robert, C. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer Science & Business Media.

[Roininen et al., 2016] Roininen, L., Girolami, M., Lasanen, S., and Markkanen, M. (2016). Hyperpriors for Matérn fields with applications in Bayesian inversion. *Inverse Problems & Imaging*, 13.

[Roininen et al., 2014] Roininen, L., Huttunen, J. M., and Lasanen, S. (2014). Whittle-Matérn priors for Bayesian statistical inversion with applications in electrical impedance tomography. *Inverse Problems & Imaging*, 8(2):561–586.

[Roininen et al., 2013] Roininen, L., Piiroinen, P., Lehtinen, M., et al. (2013). Constructing continuous stationary covariances as limits of the second-order stochastic difference equations. *Inverse Problems & Imaging*, 7(2):611–647.

[Rue and Held, 2005] Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. CRC press.

[Saks, 1937] Saks, S. (1937). *Theory of the integral*. Hafner Publishing Company.

[Schabenberger and Gotway, 2017] Schabenberger, O. and Gotway, C. A. (2017). *Statistical Methods for Spatial Data Analysis*. CRC press.

[Schwanghart, a] Schwanghart, W. Experimental (Semi-) Variogram, 09 Jan 2013. MATLAB Central File Exchange. Retrieved 21 May 2018.

[Schwanghart, b] Schwanghart, W. variogramfit, 14 Oct 2010. MATLAB Central File Exchange. Retrieved 21 May 2018.

[Shaby and Wells, 2010] Shaby, B. and Wells, M. T. (2010). Exploring an adaptive metropolis algorithm. *Currently under review*, 1:1–17.

[Sneddon, 1995] Sneddon, I. N. (1995). *Fourier Transforms*. Courier Corporation.

[Sokal, 1997] Sokal, A. (1997). Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms. In *Functional Integration*, pages 131–192. Springer.

[Stakgold and Holst, 2011] Stakgold, I. and Holst, M. J. (2011). *Green's Functions and Boundary Value Problems*, volume 99. John Wiley & Sons.

[Stein, 2012] Stein, M. L. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media.

[Tierney, 1994] Tierney, L. (1994). Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728.

[Tinney and Walker, 1967] Tinney, W. F. and Walker, J. W. (1967). Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization. *Proceedings of the IEEE*, 55(11):1801–1809.

[Van Loan, 2000] Van Loan, C. F. (2000). The ubiquitous Kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100.

[Venables and Ripley, 2001] Venables, W. N. and Ripley, B. D. (2001). *Modern Applied Statistics with S-PLUS*. Springer.

[Vogel, 2002] Vogel, C. R. (2002). *Computational Methods for Inverse Problems*. Siam.

[Walsh, 1986] Walsh, J. B. (1986). An introduction to stochastic partial differential equations. In *École d'Été de Probabilités de Saint Flour XIV - 1984*, pages 265–439. Springer Berlin Heidelberg.

[Watkins, 2004] Watkins, D. S. (2004). *Fundamentals of Matrix Computations*, volume 64. John Wiley & Sons.

[Whittle, 1954] Whittle, P. (1954). On Stationary Processes in the Plane. *Biometrika*, 41(3/4):434–449.

[Wood and Chan, 1994] Wood, A. T. and Chan, G. (1994). Simulation of Stationary Gaussian Processes in $[0, 1]^d$. *Journal of Computational and Graphical Statistics*, 3(4):409–432.