Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Bayesian and Maximum Likelihood Estimation for Gaussian Processes on an Incomplete Lattice

Jonathan R. Stroud, Michael L. Stein & Shaun Lysen

To cite this article: Jonathan R. Stroud, Michael L. Stein & Shaun Lysen (2017) Bayesian and Maximum Likelihood Estimation for Gaussian Processes on an Incomplete Lattice, Journal of Computational and Graphical Statistics, 26:1, 108-120, DOI: 10.1080/10618600.2016.1152970

To link to this article: https://doi.org/10.1080/10618600.2016.1152970

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2017, VOL. 26, NO. 1, 108–120
http://dx.doi.org/10.1080/10618600.2016.1152970

Taylor & Francis
Taylor & Francis Group

Bayesian and Maximum Likelihood Estimation for Gaussian Processes on an Incomplete Lattice

Jonathan R. Stroud[a], Michael L. Stein[b], and Shaun Lysen[c]

[a]McDonough School of Business, Georgetown University, Washington, DC; [b]Department of Statistics, University of Chicago, Chicago, Illinois; [c]Quantitative Marketing, Google, Inc., Boulder, Colorado

ABSTRACT

This article proposes a new approach for Bayesian and maximum likelihood parameter estimation for stationary Gaussian processes observed on a large lattice with missing values. We propose a Markov chain Monte Carlo approach for Bayesian inference, and a Monte Carlo expectation-maximization algorithm for maximum likelihood inference. Our approach uses data augmentation and circulant embedding of the covariance matrix, and provides likelihood-based inference for the parameters and the missing data. Using simulated data and an application to satellite sea surface temperatures in the Pacific Ocean, we show that our method provides accurate inference on lattices of sizes up to 512 × 512, and is competitive with two popular methods: composite likelihood and spectral approximations.

ARTICLE HISTORY
Received February 2014
Revised November 2015

KEYWORDS
Circulant embedding; Data augmentation; EM algorithm; Markov chain Monte Carlo; Spatial statistics

1. Introduction

Spatial lattice data are common in many fields, including environmental science, medical imaging, and computer experiments. In these applications, a common approach is to model the data as a stationary Gaussian process, and estimate the mean and covariance parameters using maximum likelihood (ML) or Bayesian methods. However, spatial lattice datasets are often extremely large and have missing values. This makes likelihood inference infeasible, since calculation of the likelihood generally requires $O(n^3)$ operations, where n is the number of observations. The likelihood cost can be reduced to $O(n^{5/2})$ for stationary Gaussian processes on a complete two-dimensional lattice (Zimmerman 1989). However, if the lattice is incomplete or has irregular boundaries, the cost is cubic in the number of observations.

To deal with this problem, a number of likelihood approximations have been proposed for large spatial datasets. Whittle (1954) introduced a spectral approximation for stationary Gaussian processes observed on a lattice. Fuentes (2007) generalized the Whittle approximation to lattices with missing data. Vecchia (1988) proposed a composite likelihood method for unequally spaced data using nearest neighbors. Stein, Chi, and Welty (2004) extended the approach to restricted ML and provided asymptotic standard errors. Kaufman, Schervish, and Nychka (2008) proposed covariance tapering for approximate likelihood estimation. Other Gaussian process approaches for large datasets include fixed-rank kriging (Cressie and Johannesson 2008), predictive processes (Banerjee et al. 2008), predictive processes with tapering (Sang and Huang 2012), and block composite likelihood (Eidsvik et al. 2014).

Another popular class of models for lattice data are Gaussian Markov random fields (GMRFs; e.g., see Rue and Held 2005). These models allow for nonstationary processes and fast computation through specification of a local neighborhood structure. Unlike Gaussian processes, GMRFs are defined over a discrete graph, rather than a continuous spatial domain. Recent work by Lindgren, Rue, and Lindström (2011) has established a link between GRMFs and Gaussian processes that allows for inference in Gaussian process models using efficient GMRF computational techniques. While the method is ingenious, it applies only for the Matérn covariance with fixed values of the smoothness parameter. Hence, the method can only be applied for a limited number of Gaussian process models.

Stein, Chen, and Anitescu (2013) recently proposed a stochastic method for unbiased estimation of the score function for stationary Gaussian processes. The estimate converges to the true score function as the Monte Carlo sample size goes to infinity. However, at present there is no feasible Bayesian Markov chain Monte Carlo (MCMC) solution for this problem, that is, one that converges to samples from the true posterior distribution as the number of iterations increases.

In this article, we propose a new ML and Bayesian estimation approach for stationary Gaussian processes observed on a large lattice with missing values. The key idea is to view the observed data as a partial realization of a periodic Gaussian random field on a larger lattice. We then treat the unobserved values on the larger lattice as missing data, and impute them within a data augmentation procedure. Conditional on the parameters, the missing data are generated using conditional simulation techniques from the geostatistics literature. Conditional on

CONTACT Jonathan R. Stroud jrs390@georgetown.edu McDonough School of Business, Georgetown University, Washington, DC 20057.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

the imputed data, we have a complete realization of a periodic process, and the complete-data likelihood can be computed efficiently using the fast Fourier transform (FFT). This iterative procedure is implemented for Bayesian inference using MCMC, and for maximum likelihood estimation (MLE) using a Monte Carlo expectation-maximization (EM) algorithm.

We note that a similar approach was proposed by Kozintsev and Kedem (2000) for discretized Gaussian fields on a lattice. They used circulant embedding and a Monte Carlo EM algorithm for MLE. Due to the non-Gaussian observations, they impute the missing values one location at a time within a Gibbs sampler. This updating scheme is computationally expensive, requiring $O(n^2)$ operations per iteration, and may potentially suffer from slow MCMC convergence due to single-site updating (e.g., Liu, Wong, and Kong 1994). In contrast, our conditional simulation (MCMC) approach generates all of the missing data simultaneously, and thus should provide faster convergence for Gaussian or conditionally Gaussian models.

Using simulated data, we first show that our approach works well and compare it to existing methods. Under different sampling designs (complete lattices, missing at random, missing blocks), we find that the Bayesian approach provides accurate inference for the parameters and the missing data on lattices up to size 512 × 512 (262,144 observations). We also show that our ML approach is competitive with both composite likelihood and spectral approximations in terms of recovering the true ML estimate. Finally, we apply the MCMC method to a satellite image of sea surface temperatures, where data are unavailable over land locations. The method is shown to provide accurate inference in this real-data application.

The rest of the article is outlined as follows. In Section 2, we introduce the stationary Gaussian process model for lattice data and describe the circulant embedding approach. Section 3 provides an MCMC method for Bayesian estimation and a Monte Carlo EM algorithm for MLE. The methods are illustrated in Section 4 with an extensive simulation study and an analysis of satellite sea surface temperatures. Conclusions are given in Section 5.

2. Likelihood for Gaussian Processes

Let $\{Z(\mathbf{s}), \mathbf{s} \in D \subseteq \mathbb{R}^d\}$ be a stationary Gaussian process with mean μ and covariance $\mathrm{cov}(Z(\mathbf{s}), Z(\mathbf{s}+\mathbf{h})) = \sigma^2 K(|\mathbf{h}|; \boldsymbol{\theta})$, where $\mathbf{h}$ is the spatial lag, $|\cdot|$ is Euclidean distance, $K(h; \boldsymbol{\theta})$ is an isotropic correlation function, and $\boldsymbol{\theta}$ is a vector of unknown parameters. Let $\mathbf{Z} = (Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n))'$ denote the observed data at a set of locations $\mathbf{s}_1, \ldots, \mathbf{s}_n$, and let $\boldsymbol{\Theta} = (\mu, \sigma^2, \boldsymbol{\theta})$ denote the unknown parameters. The likelihood function is

$$p(\mathbf{Z}|\boldsymbol{\Theta}) = (2\pi\sigma^2)^{-\frac{n}{2}} |\boldsymbol{\Sigma}(\boldsymbol{\theta})|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{Z}-\mu\mathbf{1})'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}(\mathbf{Z}-\mu\mathbf{1})\right\}, \qquad (1)$$

where $\mathbf{1} = (1, \ldots, 1)' \in \mathbb{R}^n$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is the $n \times n$ correlation matrix with elements $\boldsymbol{\Sigma}_{ij}(\boldsymbol{\theta}) = K(|\mathbf{s}_i - \mathbf{s}_j|; \boldsymbol{\theta})$. For unequally spaced locations, the likelihood becomes computationally infeasible when n is large (say, more than a few thousand), because the determinant and quadratic form require $O(n^3)$ operations to compute. If the data are observed on a two-dimensional complete rectangular lattice, then $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is block Toeplitz with Toeplitz blocks, which reduces the cost of the likelihood to $O(n^{5/2})$ (Zimmerman 1989). However, if the lattice is incomplete due to missing values or irregular boundaries, then $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is no longer Toeplitz, and exact likelihood requires $O(n^3)$ operations.

Throughout the article, we assume that $Z(\mathbf{s})$ is defined over a continuous domain in $\mathbb{R}^2$, but the data are observed on a two-dimensional rectangular lattice of size $n_1 \times n_2$. The lattice may be incomplete due to missing data or nonrectangular boundaries. To simplify exposition, we will often assume a square lattice with $n_1 = n_2$. The methods presented in this article can also be extended to more general lattices including time series or spatio-temporal data ($d = 1$ or 3), and to geometrically anisotropic processes. The goal of our analysis is to perform likelihood-based parameter estimation, and prediction of the latent Gaussian process at missing locations on the observation lattice. To do this, we use the circulant embedding approach described below.

2.1 Circulant Embedding

Circulant embedding was proposed by Wood and Chan (1994) and Dietrich and Newsam (1997) as a method for simulating stationary Gaussian random fields on a large lattice. The main idea is to embed the original $n_1 \times n_2$ lattice in $[0, s]^2$ in a larger $N_1 \times N_2$ lattice in $[0, 2rs]^2$, where $N_1 = 2rn_1$, $N_2 = 2rn_2$, and $r \geq 1$ is chosen such that N_1 and N_2 are highly composite numbers (i.e., can be written as a product of small prime numbers: 2, 3, 5, and 7). Throughout the article, we assume that $s = 1/\sqrt{2} \approx 0.707$. Following the notation in Stein (2002) and Gneiting et al. (2006), define P_sK as the function on $\mathbb{R}^2$ that has period $2s$ in each coordinate such that $P_sK(\mathbf{s}) = K(|\mathbf{s}|)$, for $\mathbf{s} \in [-s, s]^2$, and let $\mathbf{C}$ denote the $N \times N$ covariance matrix obtained by evaluating P_sK over the $N = N_1N_2$ points on the embedding lattice ordered lexicographically. Since P_sK is periodic and the domain is a rectangular grid, $\mathbf{C}$ is a block circulant matrix with circulant blocks (BCCB). This allows the matrix to be diagonalized in $O(N \log N)$ operations using the FFT.

To simulate $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, we first compute the eigenvalues of $\mathbf{C}$, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$. We then generate N independent normal random variables with variances proportional to the eigenvalues. We then apply an FFT to this random vector to obtain the periodic Gaussian random field $\mathbf{Z}$ on the embedding lattice with the correct covariance structure. The key implementation issue is the choice of the constant r. For the standard embedding approach, we choose the smallest value of $r > 1$ for which $N_1 = 2rn_1$ and $N_2 = 2rn_2$ are highly composite numbers. For some values of r, however, this may result in a nonpositive-definite matrix $\mathbf{C}$. To avoid this problem, Wood and Chan (1994) proposed increasing the value of r until $\mathbf{C}$ is positive definite; however, this often requires a very large value of r, which makes computation prohibitive.

Stein (2002) proposed an alternative approach to ensure positive definiteness of $\mathbf{C}$. Gneiting et al. (2006) labeled the method *cutoff embedding* and explored the limits of when it could be used. For a given isotropic correlation function $K(\cdot)$, Stein

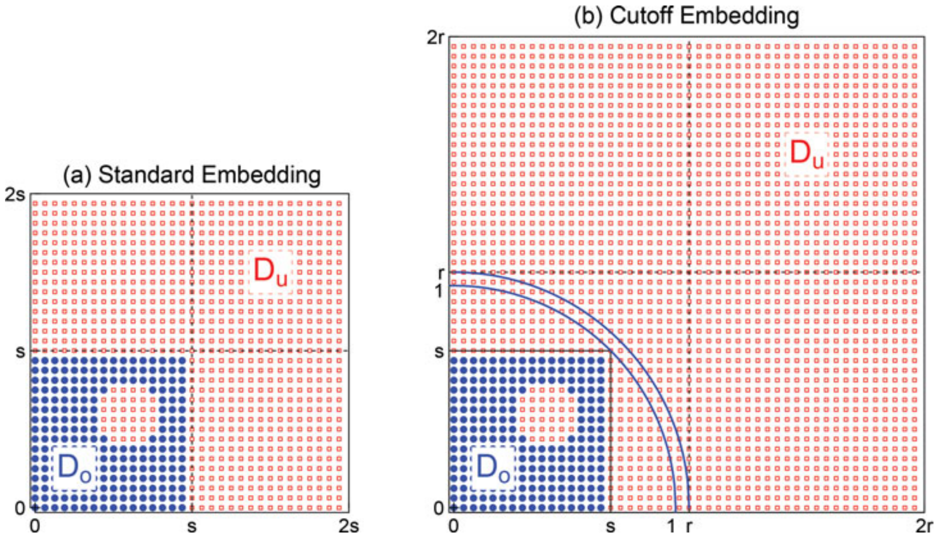

Figure 1. Circulant embedding schemes for a square lattice. Closed circles are observed locations. Open squares are unobserved locations. The original domain is 16 × 16, with missing data in a disk shape. (a) Standard embedding with a 32 × 32 embedding lattice. (b) Cutoff embedding with a 48 × 48 embedding lattice.

(2002) considered the modified correlation function

$$\tilde{K}(h) = \begin{cases} K(h) & \text{if } 0 \leq h < 1; \\ K_1(h) & \text{if } 1 \leq h < r; \\ 0 & \text{if } h \geq r, \end{cases}$$

where h is distance, $r > 1$ is the cutoff radius, and $K_1(h)$ is a function chosen to make $\tilde{K}(h)$ differentiable at r. Because $\tilde{K}(h)$ is compactly supported, the periodic function $P_r\tilde{K}$ is ensured to be positive definite, providing that $\tilde{K}(h)$ satisfies certain conditions (Gneiting et al. 2006). Stein (2002) and Gneiting et al. (2006) chose $K_1(h)$ to be either a quadratic or square root function. The circulant embedding approach is then applied using the modified covariance function $P_r\tilde{K}$, and the resulting covariance matrix $\mathbf{C}$ is guaranteed to be positive definite.

Figure 1 illustrates the spatial domains for the standard embedding and cutoff embedding schemes for a square lattice with missing observations. Here, the original lattice is 16 × 16, with data missing in a disk shape. Panel (a) illustrates the standard embedding scheme ($r = 1$), where the embedding lattice is 32 × 32. Panel (b) shows a cutoff embedding scheme with a cutoff radius of $r = 1.5s \approx 1.06$ and an embedding lattice of size 48 × 48.

2.2 BCCB Matrices

Suppose $\mathbf{Z}$ is a periodic, stationary Gaussian random field on a two-dimensional complete lattice of size $N_1 \times N_2$, with locations ordered lexicographically. Then the covariance matrix of $\mathbf{Z}$, $\mathbf{C}$, is block circulant with circulant blocks (BCCB). It follows that $\mathbf{C} = \mathbf{F}\boldsymbol{\Lambda}\mathbf{F}^*$, where $\mathbf{F}$ is the two-dimensional Fourier transform matrix, $\mathbf{F}^*$ is the corresponding inverse Fourier transform matrix, and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of eigenvalues. BCCB matrices have a number of computational advantages, namely that eigenvalues, matrix-vector multiplications, and quadratic forms can be computed efficiently in $O(N \log N)$ operations using the FFT, and they have a storage cost of $O(N)$. Properties of BCCB matrices are summarized in Appendix A. Kozintsev (1999) provided an excellent summary of BCCB matrices for Gaussian random fields.

2.3 Unconditional Simulation

Exact simulation of periodic, stationary Gaussian random fields on a grid can be performed efficiently using the circulant embedding approach of Wood and Chan (1994), Dietrich and Newsam (1997), and Stein (2002). Because the covariance matrix $\mathbf{C}$ is BCCB, unconditional simulations can be obtained in $O(N \log N)$ operations by exploiting the FFT. Specifically, to generate draws $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, we set $\mathbf{Z} = \mathbf{F}\boldsymbol{\Lambda}^{1/2}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ is a complex normal random vector, generated as $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_1 + i\boldsymbol{\epsilon}_2$, with $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The vector $\mathbf{Z} = \mathbf{Z}_1 + i\mathbf{Z}_2$ yields two independent draws $\mathbf{Z}_1$ and $\mathbf{Z}_2$ from $\mathcal{N}(\mathbf{0}, \mathbf{C})$.

2.4 Likelihood Function

Let $\mathbf{Z}$ be a random vector representing a stationary, periodic random field on a lattice; then it has distribution $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{C}(\boldsymbol{\theta}))$, where $\mathbf{C}(\boldsymbol{\theta})$ is BCCB. Let $\boldsymbol{\Theta} = (\mu, \sigma^2, \boldsymbol{\theta})$ denote the set of unknown parameters. The log-likelihood function for the complete data is (ignoring constants),

$$\begin{aligned} \log p(\mathbf{Z}|\boldsymbol{\Theta}) &= -\frac{N}{2}\log\sigma^2 - \frac{1}{2}\log|\mathbf{C}(\boldsymbol{\theta})| \\ &\quad - \frac{1}{2\sigma^2}(\mathbf{Z}-\boldsymbol{\mu})'\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{Z}-\boldsymbol{\mu}), \\ &= -\frac{N}{2}\log\sigma^2 - \frac{1}{2}\sum_{i=1}^{N}\log\lambda_i \\ &\quad - \frac{1}{2\sigma^2}\left(\boldsymbol{\Lambda}^{-1/2}\mathbf{F}^*\boldsymbol{\epsilon}\right)'\left(\boldsymbol{\Lambda}^{-1/2}\mathbf{F}^*\boldsymbol{\epsilon}\right), \end{aligned}$$

where $\boldsymbol{\epsilon} = \mathbf{Z} - \boldsymbol{\mu}$. The log-likelihood can be computed efficiently using FFTs. We first compute the eigenvalues of $\mathbf{C}(\boldsymbol{\theta})$ using a two-dimensional FFT. The determinant is then computed as the product of eigenvalues. The quadratic form is computed using an FFT followed by a vector-vector multiplication. Thus, the overall cost to compute the complete-data log-likelihood is $O(N \log N)$ operations.

2.5 Conditional Simulation

Our estimation approach requires an efficient method for generating conditional simulations of the missing data given the observed data and the parameters. Let D_o and D_u denote the observed and unobserved locations on the embedding lattice, and $D = D_o \cup D_u$ denote all locations on the embedding lattice. Define $\mathbf{Z}_o = \{Z(\mathbf{s}) : \mathbf{s} \in D_o\}$, $\mathbf{Z}_u = \{Z(\mathbf{s}) : \mathbf{s} \in D_u\}$, and $\mathbf{Z} = \{Z(\mathbf{s}) : \mathbf{s} \in D\}$ as the observed, unobserved, and complete data on the embedding lattice. Suppose that $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ and can be partitioned as

$$\begin{pmatrix} \mathbf{Z}_o \\ \mathbf{Z}_u \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_o \\ \boldsymbol{\mu}_u \end{pmatrix}, \begin{pmatrix} \mathbf{C}_{oo} & \mathbf{C}_{ou} \\ \mathbf{C}_{uo} & \mathbf{C}_{uu} \end{pmatrix}\right), \tag{2}$$

where n is the number of observed data, $N - n$ the number of unobserved data, and N is the total number of points on the embedding lattice. The conditional distribution for the unobserved data given the observed data is

$$\mathbf{Z}_u | \mathbf{Z}_o \sim \mathcal{N}\left(\boldsymbol{\mu}_u + \mathbf{C}_{uo}\mathbf{C}_{oo}^{-1}(\mathbf{Z}_o - \boldsymbol{\mu}_o), \mathbf{C}_{u|o}\right), \tag{3}$$

where $\mathbf{C}_{u|o} = \mathbf{C}_{uu} - \mathbf{C}_{uo}\mathbf{C}_{oo}^{-1}\mathbf{C}_{ou}$. Direct simulation from this distribution is infeasible when N is large, due to the cost of computing and storing the conditional covariance matrix $\mathbf{C}_{u|o}$, and its Cholesky decomposition, which require $O(N^3)$ and $O(N^2)$ operations, respectively.

We generate conditional simulations using *substitution sampling* (Matheron 1976). This method avoids the conditional covariance matrix and its Cholesky decomposition, and thus is more efficient than direct simulation. The method proceeds in two steps. First, we simulate the complete random field from its unconditional distribution, $\widetilde{\mathbf{Z}} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, using the methods from Section 2.3. The conditional simulation $\mathbf{Z}_u^*$ is then obtained by the formula

$$\mathbf{Z}_u^* = \widetilde{\mathbf{Z}}_u + \mathbf{C}_{uo}\mathbf{C}_{oo}^{-1}(\mathbf{Z}_o - \widetilde{\mathbf{Z}}_o). \tag{4}$$

It is straightforward to show that $\mathbf{Z}_u^* \sim p(\mathbf{Z}_u | \mathbf{Z}_o, \boldsymbol{\Theta})$; that is, it has the mean and covariance given in (3). For a proof, see Chilès and Delfiner (2012). Note that $\mathbf{Z}_u^*$ has the same form as the conditional mean given in (3), but with the simulated field $\widetilde{\mathbf{Z}}$ substituted for $\boldsymbol{\mu}$. To obtain the conditional simulation, we first solve the system

$$\mathbf{C}_{oo}\mathbf{x} = \mathbf{b}, \text{ where } \mathbf{b} = \mathbf{Z}_o - \widetilde{\mathbf{Z}}_o. \tag{5}$$

When n is large, it is infeasible to solve this system directly, as it requires $O(n^3)$ operations. Thus, we use a preconditioned conjugate gradient method to solve the system, which is described below. After solving the system, we obtain the conditional simulation by computing $\mathbf{w}_u = \mathbf{C}_{uo}\mathbf{x}$, which is done efficiently by exploiting the form of $\mathbf{C}$, and then setting $\mathbf{Z}_u^* = \widetilde{\mathbf{Z}}_u + \mathbf{w}_u$.

2.6 Preconditioned Conjugate Gradient

We use the preconditioned conjugate gradient algorithm (PCG; Golub and Van Loan 1996) to solve the system (5). The PCG is an iterative method that solves the modified system

$$\mathbf{M}\mathbf{C}_{oo}\mathbf{x} = \mathbf{M}\mathbf{b} \text{ where } \mathbf{b} = \mathbf{Z}_o - \widetilde{\mathbf{Z}}_o, \tag{6}$$

where $\mathbf{M}$ is an $n \times n$ preconditioner matrix. The solution to the modified system (6) is the same as the original one, but the preconditioner generally speeds up convergence (see Appendix B). The exact solution is guaranteed within n iterations; however, a good approximation can usually be obtained in far fewer iterations. The algorithm is stopped at the iteration k when the norm of the residual vector $\mathbf{r}_k = \mathbf{b} - \mathbf{C}_{oo}\mathbf{x}_k$ is less than a specified tolerance. We use the stopping criterion $|\mathbf{r}_k|/|\mathbf{b}| < \epsilon$, where ϵ is the error tolerance.

The PCG algorithm requires only matrix-vector multiplications of the form $\mathbf{C}_{oo}\mathbf{x}$ and $\mathbf{M}\mathbf{x}$. The former can be computed efficiently using submatrix-vector multiplications and exploiting the block circulant structure of $\mathbf{C}$. Suppose $\mathbf{C}$ is partitioned as in (2). To compute $\mathbf{C}_{oo}\mathbf{x}$, we pad the vector $\mathbf{x}$ with zeros, that is, $\mathbf{x}^* = (\mathbf{x}', \mathbf{0}')'$, then multiply $\mathbf{w} = \mathbf{C}\mathbf{x}^*$, and the result is obtained in the first n elements of $\mathbf{w}$:

$$\mathbf{w} = \mathbf{C}\mathbf{x}^* = \begin{pmatrix} \mathbf{C}_{oo} & \mathbf{C}_{ou} \\ \mathbf{C}_{uo} & \mathbf{C}_{uu} \end{pmatrix}\begin{pmatrix} \mathbf{x} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{C}_{oo}\mathbf{x} \\ \mathbf{C}_{uo}\mathbf{x} \end{pmatrix}.$$

This procedure is also used to compute $\mathbf{C}_{uo}\mathbf{x}$, but the result is obtained in the last $N - n$ elements of $\mathbf{w}$. This step is needed in the conditional simulations after solving the system. Each multiplication of the form $\mathbf{C}\mathbf{x}$ requires two FFTs, which require $N \log N$ operations. Each PCG iteration requires one $\mathbf{C}\mathbf{x}$ multiplication. Hence, the computational cost for one conditional simulation is $O(IN \log N)$, where I is the number of PCG iterations.

2.7 Preconditioners

The performance of the PCG algorithm depends critically on the choice of preconditioner $\mathbf{M}$. Ideally, $\mathbf{M}$ should satisfy three criteria: (1) it should be a good approximation to the precision matrix, that is, $\mathbf{M} \approx (\mathbf{C}_{oo})^{-1}$; (2) it should allow fast matrix-vector multiplications; and (3) it should have a low storage cost. Common choices for preconditioners include circulant or BCCB matrices, block diagonal matrices, incomplete LU or Cholesky decompositions, and sparse matrices (see Golub and Van Loan 1996).

In this article, we propose a new preconditioner based on the approximate likelihood methods of Vecchia (1988) and Stein, Chi, and Welty (2004). Here, the observed data $\mathbf{Z}$ are partitioned into b blocks and the likelihood is approximated by a product of conditional normal densities $p(\mathbf{A}_j\mathbf{Z}|\mathbf{B}_j\mathbf{Z})$, $j = 1, \ldots, b$, where $\mathbf{A}_j$ and $\mathbf{B}_j$ are matrices of zeros and ones that define the prediction and conditioning sets for block j. The sets are chosen to be small so that the conditional moments can be computed and stored efficiently. Since each conditional density is normal, the joint distribution is normal, and the likelihood approximation corresponds to the multivariate normal density $\mathcal{N}(\mathbf{Z}|\mathbf{0}, \mathbf{V})$, where $\mathbf{V}^{-1} = \mathbf{L}'\mathbf{D}\mathbf{L}$, where $\mathbf{L}$ and $\mathbf{D}$ are sparse $n \times n$ matrices containing the regression coefficients and precision matrices for the conditional distributions (see Appendix C). The approximation implies that $\mathcal{N}(\mathbf{0}, \mathbf{V}) \approx \mathcal{N}(\mathbf{0}, \mathbf{C}_{oo})$, or equivalently, $\mathbf{V} \approx \mathbf{C}_{oo}$. Therefore, we choose $\mathbf{V}^{-1}$ as our preconditioner. We then only need to specify the ordering of the points and our choice of prediction and conditioning sets. Here, we choose prediction sets of size 4 and conditioning sets of either 18 or 52 nearest neighbors, as illustrated in Figure 2.

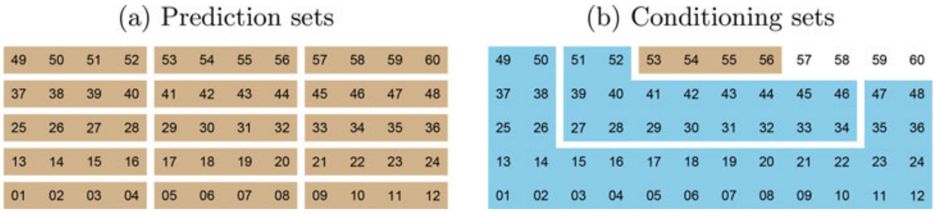

Figure 2. Illustration of the Vecchia preconditioner for a complete lattice with prediction sets of size 4 and prediction sets of size 18 or 52. The lattice points are ordered lexicographically. (a) Prediction sets. (b) Conditioning sets for the prediction set in brown. The conditioning set includes either the nearest two rows or four rows (18 or 52 neighbors, respectively).

We have also developed a number of other preconditioners, including BCCB, block diagonal, and sparse covariance or precision matrices based on Whittle's approximation, covariance tapering, and Markov random fields, respectively. One preconditioner that works quite well is the observed block of the complete-data precision matrix, that is, $(\mathbf{C}^{-1})_{oo}$. Since the inverse of a BCCB matrix is also BCCB, it has a storage cost of $O(N)$ and matrix-vector multiplications are computed efficiently using FFTs. We found that this preconditioner works well for complete or nearly complete lattices, but less well when the proportion of missing data is high and smoother covariance functions. However, we found that all of these choices were generally outperformed by the Vecchia preconditioner in terms of convergence rate and run time.

In the next section, we propose two estimation algorithms based on the ideas of circulant embedding and conditional simulation. First, we propose an MCMC algorithm for Bayesian inference. Second, we introduce a Monte Carlo EM algorithm for MLE.

3. Parameter Estimation

3.1 Bayesian Estimation

For the Bayesian analysis, we specify a prior distribution for the unknown parameters, $\pi(\boldsymbol{\Theta})$, and make inference based on the joint posterior distribution

$$\pi(\boldsymbol{\Theta}, \mathbf{Z_u}|\mathbf{Z_o}) \propto \mathbf{p}(\mathbf{Z}|\boldsymbol{\Theta})\,\pi(\boldsymbol{\Theta}),$$

where $\mathbf{Z} = (\mathbf{Z}_o, \mathbf{Z}_u)$ denotes the complete data. This joint posterior distribution is typically unavailable in closed form. Therefore, we propose an MCMC algorithm to sample from it. Specifically, we propose a two-block Gibbs sampler that alternates between updating the missing data and the parameters. Given initial parameter values, $\boldsymbol{\Theta}^0$, the MCMC algorithm proceeds as follows for $i = 1, \dots, M$:

1. Generate $\mathbf{Z}_u^i \sim p(\mathbf{Z}_u|\mathbf{Z}_o, \Theta^{i-1})$.
2. Generate $\boldsymbol{\Theta}^i \sim \pi(\Theta|\mathbf{Z}^i)$.

The missing data are updated using conditional simulation methods described in Section 2.5. The parameters are updated using a block Metropolis-Hastings step, described below.

Given the complete data, the parameters are generated from their conditional distribution $\pi(\boldsymbol{\Theta}|\mathbf{Z}) \propto p(\mathbf{Z}|\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})$. This distribution is easy to evaluate, but is generally unavailable in closed form due to the nonlinearity of $\boldsymbol{\theta}$ in the determinant and the quadratic form. Bayesian MCMC approaches to estimate covariance parameters in spatial Gaussian processes include Ecker and Gelfand (1997), who proposed a Metropolis algorithm, and Agarwal and Gelfand (2005), who proposed a slice sampling approach. Here, we use a Metropolis-Hastings scheme to update the parameters, which proceeds as follows.

1. Generate $\boldsymbol{\Theta}^*$ from a proposal distribution $q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^i)$.
2. Accept $\boldsymbol{\Theta}^*$ with probability

$$\min\left\{1, \frac{p(\mathbf{Z}|\boldsymbol{\Theta}^*)\,\pi(\boldsymbol{\Theta}^*)}{p(\mathbf{Z}|\boldsymbol{\Theta}^i)\,\pi(\boldsymbol{\Theta}^i)}\frac{q(\boldsymbol{\Theta}^i|\boldsymbol{\Theta}^*)}{q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}^i)}\right\}.$$

The computational cost to generate the missing data is $O(IN \log N)$, and the cost to update the parameters is $O(N \log N)$. Hence, the cost for each iteration of the Gibbs sampler is $O(IN \log N)$ operations, and the total cost for M iterations of the sampler is $O(MIN \log N)$.

The Bayesian approach also provides inference for the random field at missing locations via the posterior predictive distribution (see Handcock and Stein 1993). If the missing data lie on the original lattice or the embedding lattice, the MCMC algorithm automatically generates samples from their distribution as part of the imputation step.

3.2 Maximum Likelihood Estimation

For MLE, we propose an EM algorithm (Dempster, Laird, and Rubin 1977) to obtain the ML estimate $\widehat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} p(\mathbf{Z}_o|\boldsymbol{\Theta})$. The algorithm iterates between the E-step and M-step until convergence. In the E-step, we calculate the expected complete-data log-likelihood given the observed data and the current parameter, $\boldsymbol{\Theta}^t$:

$$Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t) = \int \log p(\mathbf{Z}|\boldsymbol{\Theta})p(\mathbf{Z}|\mathbf{Z}_o, \boldsymbol{\Theta}^t)d\mathbf{Z}. \quad (7)$$

In the M-step, we maximize this function to obtain the next parameter value, $\boldsymbol{\Theta}^{t+1}$. Under the Gaussian model (2), the distribution for the complete data is $\mathbf{Z}|\boldsymbol{\Theta} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\Theta}), \mathbf{C}(\boldsymbol{\Theta}))$, and the conditional distribution for the complete data is $\mathbf{Z}|\mathbf{Z}_o, \boldsymbol{\Theta}^t \sim \mathcal{N}(\widetilde{\boldsymbol{\mu}}(\boldsymbol{\Theta}^t), \widetilde{\mathbf{C}}(\boldsymbol{\Theta}^t))$. Suppressing dependence on parameters, and ignoring constants, the expectation (7) is

$$Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t) = -\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}E\left\{(\mathbf{Z}-\boldsymbol{\mu})'\mathbf{C}^{-1}(\mathbf{Z}-\boldsymbol{\mu})|\mathbf{Z}_o, \boldsymbol{\Theta}^t\right\} \quad (8)$$

$$= -\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\left\{\mathrm{tr}(\mathbf{C}^{-1}\widetilde{\mathbf{C}}) + (\widetilde{\boldsymbol{\mu}}-\boldsymbol{\mu})'\mathbf{C}^{-1}(\widetilde{\boldsymbol{\mu}}-\boldsymbol{\mu})\right\} \quad (9)$$

Note that the trace term in (9) involves $\widetilde{\mathbf{C}}$, the conditional covariance matrix for the complete data. This matrix consists of $\widetilde{\mathbf{C}}_{u|o}$ in the lower diagonal block, and zeros elsewhere, and does not have a BCCB form. Thus, it is infeasible to compute $\widetilde{\mathbf{C}}$ when n is large and therefore Q cannot be evaluated exactly. Instead, we propose a Monte Carlo approach, where we approximate the expected log-likelihood (8) by

$$\widehat{Q}(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t) = -\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\left\{\frac{1}{M}\sum_{i=1}^{M}(\mathbf{Z}^{(i)}-\boldsymbol{\mu})'\mathbf{C}^{-1}(\mathbf{Z}^{(i)}-\boldsymbol{\mu})\right\}, \tag{10}$$

where $\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(M)} \sim p(\mathbf{Z}|\mathbf{Z}_o, \boldsymbol{\Theta}^t)$ are conditional simulations of the complete data generated using the current parameter value $\boldsymbol{\Theta}^t$. We then maximize $\hat{Q}$ to obtain the new parameter value, $\boldsymbol{\Theta}^{t+1}$. The Monte Carlo expectation avoids computing the conditional covariance matrix, and requires only a determinant and M quadratic forms involving the matrix $\mathbf{C}$. Therefore, $\hat{Q}$ can be computed for large n. Given an initial parameter $\boldsymbol{\Theta}^0$, the EM algorithm proceeds as follows for $t = 0, 1, \ldots, T$.

1. (E-step) Generate $\mathbf{Z}_u^{(1)}, \ldots, \mathbf{Z}_u^{(M)} \sim p(\mathbf{Z}_u|\mathbf{Z}_o, \boldsymbol{\Theta}^t)$.
2. (M-step) Update $\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} \hat{Q}(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$.

In Step 1 of the EM algorithm, we generate M conditional simulations using the current parameter $\boldsymbol{\Theta}^t$ using the approach described in Section 2.5. In Step 2, we maximize the expected complete-data log-likelihood using numerical optimization methods such as Quasi-Newton or Powell's method.

4. Examples

4.1 Simulation Study

To study the performance of our algorithm, we first carry out a detailed simulation study using different lattice sizes and missingness patterns. We generate data $Z(\mathbf{s})$ from a stationary, isotropic Gaussian process, with mean μ and covariance $\operatorname{cov}(Z(\mathbf{s}), Z(\mathbf{s}+\mathbf{h})) = \sigma^2 K(|\mathbf{h}|)$, where $K(\cdot)$ is a powered exponential correlation with microscale noise Cressie (1993):

$$K(h) = \exp\{-(h/\lambda)^{\alpha}\} + c\mathbb{1}_{(h=0)}. \tag{11}$$

Here, σ^2 is the partial sill parameter, $\lambda > 0$ is the range parameter, $\alpha \in (0, 2]$ is the shape parameter, $c \geq 0$ is the noise-to-signal ratio, and $\mathbb{1}_A$ is the indicator function. The parameter $\tau^2 = c\sigma^2$ represents the variance of the microscale noise. This covariance model is quite flexible and includes the exponential ($\alpha = 1$) and squared exponential ($\alpha = 2$) models as special cases. Throughout the simulation study, we fix c at its true value and use the methods from Section 3 to estimate the parameters μ, σ^2, and $\boldsymbol{\theta} = (\lambda, \alpha)$. This is done to make the simulation study feasible while still allowing a nugget effect, which is often present in practice.

For circulant embedding, we use a modified version of the cutoff embedding approach described in Section 2, with the modified correlation function

$$\tilde{K}(h) = \begin{cases} \exp\{-(h/\lambda)^{\alpha}\} + c\mathbb{1}_{(h=0)}, & 0 \leq h < 1; \\ a + b(h-r)^2, & 1 \leq h < r; \\ a, & h \geq r, \end{cases} \tag{12}$$

where $r > 1$, and $a = \exp\{-(1/\lambda)^{\alpha}\}/\{1 - (r-1)/2\lambda\}$ and $b = \exp\{-(1/\lambda)^{\alpha}\}/\{2\lambda(r-1)\}$ are chosen to make $\tilde{K}(h)$ differentiable at 1 and $\tilde{K}'(r) = 0$. This approach is similar to cutoff embedding with a quadratic function $K_1(h)$, but here r is selected by the user, and $\tilde{K}(h)$ is set to a constant rather than zero for $h \geq r$. While this approach does not guarantee positive definite embeddings, it leads to fewer violations than standard embedding, while allowing for a much smaller value of r than required for cutoff x embedding.

The value of r required for a positive definite embedding depends on the parameters λ, α, c. If these parameters were known, the value of r could be obtained by trial and error. However, in the context of an MCMC or EM algorithm, the parameters are unknown and changing at each iteration. One possible solution is to adaptively update r (and the size of the embedding grid) along with the parameters. However, this implies a variable-dimensional state space, which requires reversible jump or other transdimensional MCMC methods, which are difficult to use in high-dimensional settings. Thus, to simplify estimation, we hold r fixed throughout the estimation algorithm, and choose its value based on prior information, with subsequent modifications based on a few trial runs of the algorithm.

For the simulation study, we generate data on a $n_1 \times n_1$ lattice on $[0, s]^2$, where $s = 1/\sqrt{2}$. The true parameter values are $\sigma^2 = 4$, $\lambda = 0.1$, $\alpha = 1$, $c = 0.01$, and $\mu = 10$, which corresponds to an exponential covariance with small microscale noise. We use the modified correlation function (12) with $r = 1.5s \approx 1.06$, and an embedding lattice of size $3n_1 \times 3n_1$ on $[0, 3s]^2$. The "cutoff" radius r is about 10 times larger than the spatial range parameter λ. We focus on the behavior of the algorithms over a fixed spatial region as the grid becomes increasingly dense. We consider observation lattices of size $n_1 = 32$, 64, 128, 256, and 512 with corresponding embedding lattices of size $N_1 = 96$, 192, 384, 768, and 1536, and three different missingness patterns: complete lattice, 10% missing at random, and 10% missing disk.

4.1.1 Bayesian Analysis

For the Bayesian analysis, we assume the prior distribution of the form $\pi(\mu, \sigma^2, \boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta})/\sigma^2$, corresponding to a noninformative Jeffreys' prior for the mean and variance, $\pi(\mu, \sigma^2) \propto 1/\sigma^2$, times a proper prior for the range and smoothness parameters, $\pi(\boldsymbol{\theta}) = \pi(\lambda)\pi(\alpha)$, where

$$\pi(\lambda) = \frac{0.5}{(1 + 0.5\lambda)^2}$$

and $\pi(\alpha) = \mathcal{U}(0, 2)$. Note that a proper prior for λ is needed to ensure a proper posterior (Berger, De Oliveira, and Sansó 2001). Here, $\pi(\lambda)$ has a mode of zero, a median of two, and a long right tail, reflecting our belief that large values of the range are less likely than small ones, while still allowing for large values. This prior is uninformative for $0.5\lambda/(1 + 0.5\lambda)$ on $[0, 1]$. A similar prior was used by Handcock and Stein (1993) and Handcock and Wallis (1994). The prior for the shape parameter α is proper and uniform over its support of $[0, 2]$.

Let $\mathbf{Z}$ denote the complete data over the embedding lattice. Then $\mathbf{Z} \sim \mathcal{N}(\mu\mathbf{1}, \sigma^2\mathbf{C}(\boldsymbol{\theta}))$, where $\mathbf{C}(\boldsymbol{\theta})$ is the correlation matrix obtained by evaluating $P_r\tilde{K}(|\mathbf{s}_i - \mathbf{s}_j|)$ over all pairs of points

Table 1. MCMC results for different lattice sizes and sampling designs.

n_1	Sill, σ^2	Range, λ	Power, α	Mean, μ	Iter	Time
			Complete lattice			
32	4.20 (1.14) 96	0.111 (0.039) 94	0.997 (0.059) 92	10.01 (0.44) 89	3	2 min
64	3.83 (0.75) 90	0.096 (0.020) 90	1.001 (0.032) 94	10.03 (0.38) 79	5	6 min
128	3.85 (0.55) 95	0.095 (0.012) 90	1.000 (0.024) 98	10.05 (0.37) 87	8	25 min
256	3.92 (0.25) 89	0.099 (0.004) 93	0.995 (0.019) 94	10.04 (0.38) 82	14	2.1 hr
512	4.00 (0.08) 91	0.100 (0.002) 85	1.000 (0.010) 83	10.00 (0.38) 83	23	14.2 hr
			Missing at random (10%)			
32	4.26 (1.18) 93	0.112 (0.039) 93	1.000 (0.062) 92	10.02 (0.46) 88	6	4 min
64	3.86 (0.79) 91	0.097 (0.022) 91	0.999 (0.032) 94	10.02 (0.39) 82	11	17 min
128	3.85 (0.57) 93	0.095 (0.012) 88	1.001 (0.024) 98	10.05 (0.37) 83	18	1.3 hr
256	3.94 (0.27) 88	0.099 (0.005) 90	0.996 (0.019) 93	10.04 (0.38) 81	26	6.2 hr
512	3.99 (0.08) 87	0.100 (0.002) 86	0.999 (0.011) 83	10.00 (0.38) 82	38	1.3 day
			Missing disk (10%)			
32	4.22 (1.14) 95	0.112 (0.040) 94	0.998 (0.063) 92	10.02 (0.45) 87	16	3 min
64	3.79 (0.75) 92	0.095 (0.021) 87	0.999 (0.033) 95	10.02 (0.37) 82	44	18 min
128	3.84 (0.56) 94	0.095 (0.012) 93	1.000 (0.024) 99	10.05 (0.37) 83	60	1.5 hr
256	3.91 (0.27) 82	0.099 (0.005) 89	0.994 (0.020) 88	10.05 (0.38) 80	104	10.6 hr
512	3.99 (0.08) 91	0.100 (0.002) 85	0.999 (0.011) 86	10.00 (0.38) 80	200	3.4 day

NOTE: The covariance function is $C(h) = \sigma^2\{\exp(-(h/\lambda)^\alpha) + c\mathbb{1}_{(h=0)}\}$, with parameters $\sigma^2 = 4$, $\lambda = 0.1$, $\alpha = 1$, $c = 0.01$, $\mu = 10$. The columns are the lattice size ($n_1 = n_2$), posterior means, standard deviations, and 95% coverage probabilities for the unknown parameters, average number of PCG iterations, and maximum run time. Results are based on 100 simulated datasets, 2000 MCMC iterations, Vecchia preconditioner with 52 neighbors, and a PCG tolerance of $\epsilon = 10^{-5}$.

$(\mathbf{s}_i, \mathbf{s}_j)$ on the embedding lattice. Multiplying the prior distribution and the complete-data likelihood, we obtain the full conditional posterior for the parameters:

$$\pi(\mu, \sigma^2, \boldsymbol{\theta}|\mathbf{Z}) \;\propto\; (\sigma^2)^{-\frac{N}{2}-1}|\mathbf{C}(\boldsymbol{\theta})|^{-\frac{1}{2}} \times \exp\left\{-\frac{1}{2\sigma^2}\left[\frac{(\mu-\hat{\mu})^2}{(\mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{1})^{-1}} + S(\boldsymbol{\theta})\right]\right\}\pi(\boldsymbol{\theta}),$$

where $\hat{\mu} = \mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{Z}/(\mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{1}) = \mathbf{1}'\mathbf{Z}/N = \bar{Z}$ and $S(\boldsymbol{\theta}) = (\mathbf{Z} - \hat{\mu}\mathbf{1})'\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{Z} - \hat{\mu}\mathbf{1})$ are the generalized least squares estimate and sum of squares, respectively. Note that because $\mathbf{C}(\boldsymbol{\theta})$ is a BCCB matrix, the least squares estimate $\hat{\mu}$ does not depend on $\boldsymbol{\theta}$, and the determinant and sum of squares can be computed efficiently.

We follow the MCMC approach described in Section 3.1, but improve the efficiency of the algorithm by updating all parameters as a block. To do this, we factorize the full conditional posterior for the parameters as $\pi(\mu, \sigma^2, \boldsymbol{\theta} \mid \mathbf{Z}) = \pi(\mu|\sigma^2, \boldsymbol{\theta}, \mathbf{Z})$ $\pi(\sigma^2 \mid \boldsymbol{\theta}, \mathbf{Z})\;\pi(\boldsymbol{\theta} \mid \mathbf{Z})$, where

$$\pi(\mu|\sigma^2, \boldsymbol{\theta}, \mathbf{Z}) = \mathcal{N}(\bar{Z}, \sigma^2(\mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{1})^{-1})$$

$$\pi(\sigma^2|\boldsymbol{\theta}, \mathbf{Z}) = \mathcal{IG}\,((N-1)/2, S(\boldsymbol{\theta})/2)$$

$$\pi(\boldsymbol{\theta} \mid \mathbf{Z}) \propto |\mathbf{C}(\boldsymbol{\theta})|^{-\frac{1}{2}}|\mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{1}|^{-\frac{1}{2}}\,\{S(\boldsymbol{\theta})\}^{-\frac{N-1}{2}}\;\pi(\boldsymbol{\theta}). \quad (13)$$

The conditional posterior for (μ, σ^2) has the standard conjugate normal-inverse gamma form. The marginal posterior for $\boldsymbol{\theta}$ in (13) is not of a recognizable form, but can be efficiently evaluated pointwise, since the determinants and sum of squares involve the BCCB matrix $\mathbf{C}(\boldsymbol{\theta})$. This leads to a Metropolis-Hastings algorithm to generate the parameters as a block from their full conditional distribution $(\mu, \sigma^2, \boldsymbol{\theta}) \sim \pi(\mu, \sigma^2, \boldsymbol{\theta}|\mathbf{Z})$. Given the current values $(\mathbf{Z}^i, \boldsymbol{\theta}^i)$, the parameter update is as follows:

1. Draw a candidate value $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^i)$.
2. Accept $\boldsymbol{\theta}^*$ with probability

$$\min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\mathbf{Z}^i)}{\pi(\boldsymbol{\theta}^i|\mathbf{Z}^i)}\frac{q(\boldsymbol{\theta}^i|\boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^i)}\right\}. \quad (14)$$

3. If $\boldsymbol{\theta}^*$ is accepted, draw $\sigma^2 \sim \mathcal{IG}((N-1)/2, S(\boldsymbol{\theta}^*)/2)$, and $\mu \sim \mathcal{N}(\bar{\mathbf{Z}}^i, \sigma^2(\mathbf{1}'\mathbf{C}(\boldsymbol{\theta}^*)^{-1}\mathbf{1})^{-1})$; otherwise, leave (μ, σ^2) unchanged.

We choose the proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}^i)$ to be a bivariate log-normal, with covariance matrix chosen to achieve an acceptance probability of around 35%. We note that a similar Metropolis algorithm with block updating was proposed by Huerta, Sansó, and Stroud (2004) in the context of spatio-temporal models. They found that blocking provides huge gains in computational efficiency relative to updating each parameter one at a time, particularly for large datasets.

The results from the MCMC algorithm are reported in Table 1 and Figures 3 and 4. All results are based on the Vecchia preconditioner with prediction sets of size 4 and conditioning sets of size 52, with a PCG tolerance of $\epsilon = 10^{-5}$. Calculations are implemented in C on an Intel Xeon 2.8 GHz processor with 22 GB of RAM on a Mac OS X operating system. FFTs are implemented with the FFTW package (Frigo and Johnson 1998).

Figures 3 and 4 show the results from the MCMC analysis of one simulated dataset on a 128 × 128 lattice with 10% missing values in a disk-shaped region. In total, there are 14,743 observations. The MCMC described above was run for 10,000 iterations after a burn-in period of 1000. Figure 3 shows posterior trace plots and histograms for the parameters. The trace plots are stable, indicating no clear violations of stationarity. The histograms are all unimodal and fairly symmetric. All of the histograms contain the true parameter values, and all of the 95% posterior intervals (not shown) contain the true parameter values. This indicates that the algorithm is providing accurate samples from the posterior distribution.

Figure 4 summarizes the posterior distribution of the spatial field. Panel (a) shows the simulated data $Z(\mathbf{s})$, including the circular region, which is treated as missing data for the analysis. Panels (b) and (c) show the posterior mean and standard deviation for $Z(\mathbf{s})$, and panels (d)–(f) show the difference between three posterior draws and the posterior mean. Note that the posterior mean agrees with the observed data at the observation

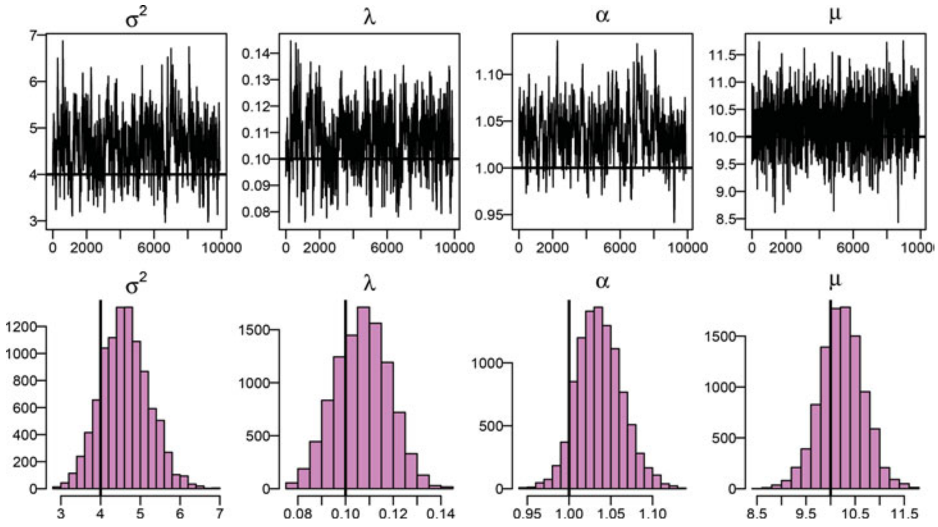

Figure 3. MCMC trace plots and histograms for the parameters. Results are based on simulated data on a 128 × 128 lattice with 10% missing data in a disk shape. The true parameter values are $\sigma^2 = 4, \lambda = 0.1, \alpha = 1$, and $\mu = 10$.

locations, and it approaches the unconditional mean ($\hat{\mu} \approx 10$) at the center of the domain. The posterior uncertainty for $Z(\mathbf{s})$ is zero at the observation locations, and the posterior standard deviation converges to the unconditional SD ($\hat{\sigma} \approx 2$) near the center of the domain. Panels (d)–(f) illustrate the sample-to-sample variation in the posterior field; the plots show the correlation length scales and highlight that the uncertainty is highest in the center of the domain.

We next study the performance of the MCMC algorithm for different lattice sizes and different missingness patterns. As mentioned previously, we consider increasingly dense lattices of size $n_1 \times n_1$ with $n_1 = 32, 64, 128, 256$, and 512, with corresponding embedding lattices of size $3n_1 \times 3n_1$. We consider three different missingness patterns: complete lattice, 10% missing at random, and 10% missing disk. For each lattice size and missingness pattern, we simulated 100 datasets. The results are based on 2000 MCMC iterations after a burn-in period of 500. The true parameters, choice of r, and proposal distribution are the same as above.

Table 1 reports the parameter estimates, number of PCG iterations, and computational run times for the MCMC results. For each lattice size and missingness pattern, we report the posterior means, standard deviations, and 95% coverage probabilities for the unknown parameters; the average number of PCG iterations per conditional simulation; and total run time in minutes, averaged across 100 datasets. There are a number of points to note. First, the posterior means are fairly close to the true values, and get closer as n increases. Second, notice that the posterior standard deviation decreases as n increases for all parameters except μ. The latter is expected under fixed-domain asymptotics (Stein 1999), since the degree of "learning" about the mean is limited by the size of the domain rather than the lattice size.

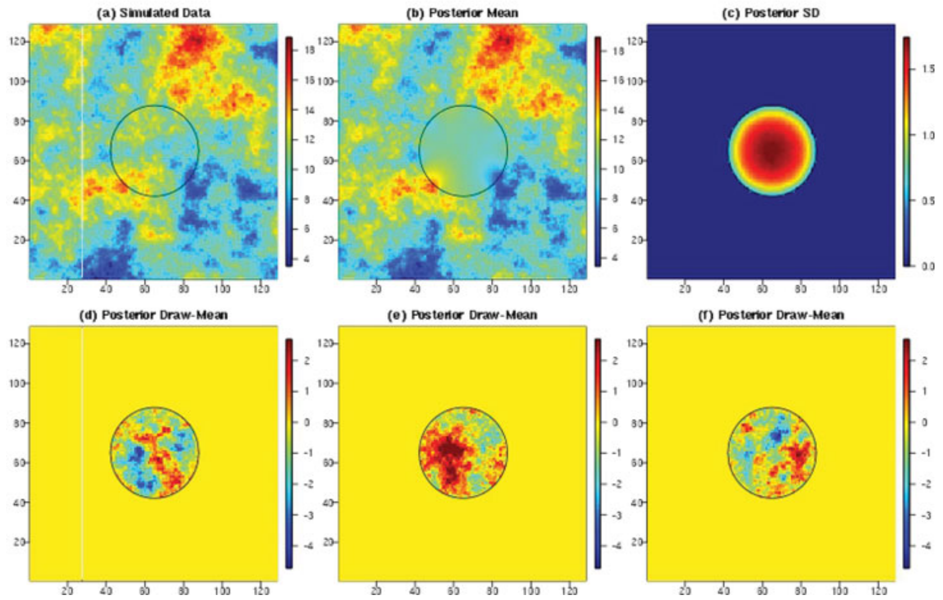

Figure 4. MCMC posterior summaries of the random field for simulated data on a 128 × 128 lattice with 10% missing data in a disk shape. (a) Simulated dataset $Z(\mathbf{s})$; the data inside of the circle is treated as missing data. (b) and (c) Posterior mean and standard deviation for $Z(\mathbf{s})$. (d)–(f) Difference between three different posterior draws of $Z(\mathbf{s})$ and the posterior mean.

Third, note that, with the exception of μ, the 95% posterior coverage probabilities for all parameters are close to the nominal values.

Fourth, the number of PCG iterations (I) depends on lattice size and missingness pattern. For complete lattices, the average number of PCG iterations increases from 3 to 23 as we move from $n_1 = 32$ to $n_1 = 512$. Similar proportional increases occur for the other two missingness patterns. In addition, complete lattice designs require fewer PCG iterations than incomplete lattices. For example, for $n_1 = 128$, the average number of iterations are 8, 18, and 60 for the complete data, missing at random, and missing disk designs, respectively. This implies that the Vecchia preconditioner provides a better approximation to the precision matrix for complete lattices than for incomplete lattices. To quantify these relationships, we fit regressions of the number of PCG iterations against lattice size (both on the logarithmic scale) for different missingness patterns, parameter values, and PCG error tolerances. The regression estimates indicate that the number of PCG iterations grows like $O(n^a)$ where $0.3 < a < 0.5$, where a depends on missingness pattern, true parameters, and error tolerance.

Finally, note that for a given lattice size and missingness pattern, the MCMC run times are roughly proportional to the number of PCG iterations. For the complete lattice design, 2500 iterations of the MCMC algorithm requires about 2 min for $n_1 = 32$, 6 min for $n_1 = 64$, 25 min for $n_1 = 128$, and about 2.1 hr for $n_1 = 256$. The run times for the latter two seem quite reasonable given that the method provides an excellent approximation to the posterior for datasets of this size. We again used regression to estimate the growth in run time as a function of lattice size. We found that run time grows like $O(n^a)$ for $1.1 < a < 1.6$, where a depends on the missingness pattern, parameter values, and PCG tolerance.

4.1.2 Maximum Likelihood Analysis

We use the EM algorithm from Section 3.2 with a slight modification to improve computational efficiency. In particular, the expected complete-data log-likelihood for the model in Section 4.1 allows us to concentrate out μ and σ^2 and maximize over $\boldsymbol{\theta}$ alone. Define $\boldsymbol{\Theta} = (\mu, \sigma^2, \boldsymbol{\theta})$. The expected complete-data log-likelihood is

$$\begin{aligned} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t) = & -\frac{N}{2}\log\sigma^2 - \frac{1}{2}\log|\mathbf{C}(\boldsymbol{\theta})| \\ & -\frac{1}{2\sigma^2}E\left\{S(\mu, \boldsymbol{\theta}, \mathbf{Z})|\mathbf{Z}_o, \boldsymbol{\Theta}^t\right\}, \end{aligned} \qquad (15)$$

where $S(\mu, \boldsymbol{\theta}, \mathbf{Z}) = (\mathbf{Z} - \mu\mathbf{1})'\mathbf{C}(\boldsymbol{\theta})^{-1}(\mathbf{Z} - \mu\mathbf{1})$. Using Equations. (8) and (9) with $\boldsymbol{\mu} = \mu\mathbf{1}$ and $\mathbf{C} = \sigma^2\mathbf{C}(\boldsymbol{\theta})$, and differentiating with respect to μ and σ^2, we can show that $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$ is maximized at

$$\hat{\mu}(\boldsymbol{\theta}) = \frac{\mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\widetilde{\boldsymbol{\mu}}}{\mathbf{1}'\mathbf{C}(\boldsymbol{\theta})^{-1}\mathbf{1}} = \frac{c^*\mathbf{1}'\widetilde{\boldsymbol{\mu}}}{c^*\mathbf{1}'\mathbf{1}} = \frac{\mathbf{1}'\widetilde{\boldsymbol{\mu}}}{N}, \qquad (16)$$

$$\hat{\sigma}^2(\boldsymbol{\theta}) = E\left\{S(\hat{\mu}, \boldsymbol{\theta}, \mathbf{Z})|\mathbf{Z}_o, \boldsymbol{\Theta}^t\right\}/N, \qquad (17)$$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} Q_p(\boldsymbol{\theta}|\boldsymbol{\Theta}^t), \qquad (18)$$

where $\widetilde{\boldsymbol{\mu}} = E(\mathbf{Z}|\mathbf{Z}_o, \boldsymbol{\Theta}^t)$ is the conditional mean for the complete data, $S(\hat{\mu}, \boldsymbol{\theta}, \mathbf{Z})$ is the generalized sum of squares, and $Q_p(\boldsymbol{\theta}|\boldsymbol{\Theta}^t)$ is the profile function for $\boldsymbol{\theta}$ obtained by substituting $\hat{\mu}(\boldsymbol{\theta})$ and $\hat{\sigma}^2(\boldsymbol{\theta})$ into (15). Note that because $\mathbf{C}(\boldsymbol{\theta})$ is BCCB, the estimate $\hat{\mu}$ does not depend on $\boldsymbol{\theta}$, and can be computed independently of the other parameters. The profile function for $\boldsymbol{\theta}$ is

$$Q_p(\boldsymbol{\theta}|\boldsymbol{\Theta}^t) = -\frac{N}{2}\log\hat{\sigma}^2(\boldsymbol{\theta}) - \frac{1}{2}\log|\mathbf{C}(\boldsymbol{\theta})|. \qquad (19)$$

Maximization of this function provides the estimate $\hat{\boldsymbol{\theta}}$, which is then substituted into (17) to obtain an estimate of σ^2 as $\hat{\sigma}^2(\hat{\boldsymbol{\theta}})$. However, as noted before, the conditional expectation of $S(\hat{\mu}, \boldsymbol{\theta}, \mathbf{Z})$ in (17) is computationally intractible for large datasets, so we approximate it using the Monte Carlo estimate:

$$\widetilde{S}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} S(\hat{\mu}, \boldsymbol{\theta}, \mathbf{Z}^{(i)}), \qquad (20)$$

where $\mathbf{Z}^{(i)} \sim p(\mathbf{Z}\,|\,\mathbf{Z}_o, \boldsymbol{\Theta}^t)$, $i = 1, \ldots, M$ are conditional simulations of the complete data generated using the current parameter value $\boldsymbol{\Theta}^t$. We then substitute $\widetilde{S}(\boldsymbol{\theta})$ for $E\{S(\hat{\mu}, \boldsymbol{\theta}, \mathbf{Z})|\mathbf{Z}_o, \boldsymbol{\Theta}^t\}$ in (17) and (19) to obtain an approximate profile function, $\widehat{Q}_p(\boldsymbol{\theta}|\boldsymbol{\Theta}^t)$. This function is then maximized numerically to obtain the estimate $\hat{\boldsymbol{\theta}}$ and finally, $\hat{\sigma}^2 = \hat{\sigma}^2(\hat{\boldsymbol{\theta}})$.

To illustrate the MLE procedure, we generate 50 datasets on a 32×32 square lattice using three different missingness patterns (complete lattice, 10% missing at random, and 10% missing disk). We assume an exponential covariance with no nugget effect (i.e., we fix $\alpha = 1$ and $c = 0$), with true parameter values $\sigma^2 = 2, \lambda = 0.141$, and $\mu = 0$. The exponential model is used in this example because it has a closed-form spectral density, which is needed for the two spectral ML methods described below.

For the EM algorithm, we use the approach described above with a Monte Carlo sample size of $M = 400$, using the Vecchia preconditioner and a PCG tolerance of $\epsilon = 10^{-5}$ as in Section 4.1.1. For comparison, we also implement two approximate ML methods: the composite likelihood approach of Vecchia (1988) and Stein, Chi, and Welty (2004), using prediction sets of size 4 and conditioning sets of size 52; and the spectral approximations of Whittle (1954) and Fuentes (2007) for complete and incomplete lattices, respectively. Note that we also considered other approximate likelihood methods, including covariance tapering and block diagonal approximations, but these methods were substantially less accurate than the other methods, so these results are not reported here.

Table 2 shows the root mean squared difference (RMSD) for each of the estimation methods: EM algorithm, composite likelihood, and spectral approximation. For each unknown parameter $\theta = \sigma^2$, λ, μ, and each estimation method $k = 1, 2, 3$, we define

$$\text{RMSD}_k = \sqrt{\frac{1}{50}\sum_{r=1}^{50}(\hat{\theta}_{k,r} - \hat{\theta}_r)^2},$$

where $\hat{\theta}_{k,r}$ denotes the approximate MLE for method k, and $\hat{\theta}_r$ denotes the exact MLE for each replicate $r = 1, 2, \ldots, 50$. The exact MLE can be computed due to the relatively small sample size in this example ($n \leq 1024$). Note that the RMSD compares the approximate MLE to the exact MLE, rather than the true parameter. This allows us to focus on the approxi-

Table 2. Maximum likelihood estimation for the exponential covariance model, 32 × 32 lattice, and three different sampling designs.

	Complete			Random 10%			Disk 10%			
	σ^2	λ	μ	σ^2	λ	μ	σ^2	λ	μ	Run time
Exact ML	450	34.8	550	446	47.2	545	442	47.3	554	1.6
EM algorithm	26	2.5	2	31	2.9	2	26	2.5	3	486
Composite ML	45	4.2	22	58	5.9	54	268	25.5	393	1.7
Spectral ML	387	30.9	237	596	92.5	231	370	40.2	212	3.0

NOTE: Results are based on 50 simulated datasets. For exact MLE, we report RMSE $= (\frac{1}{50}\sum_{r=1}^{50}(\hat{\theta}_r - \theta)^2)^{1/2}$, where θ is the true parameter and $\hat{\theta}_r$ is the exact MLE. For the approximate methods, we report RMSD$_k = (\frac{1}{50}\sum_{r=1}^{50}(\hat{\theta}_{k,r} - \hat{\theta}_r)^2)^{1/2}$, where $\hat{\theta}_{k,r}$ is the approximate MLE for method k. True parameter values are $\sigma^2 = 2$, $\lambda = 0.141$, $\mu = 0$. Run times are given in seconds. All other values in the table were multiplied by 1000.

mation error of the estimate (e.g., an RMSD of 0 means no approximation error). For the exact MLEs, we also compute the root mean squared error, RMSE $= (\frac{1}{50}\sum_{r=1}^{50}(\hat{\theta}_r - \theta)^2)^{1/2}$, where θ is the true parameter value. For all methods, numerical maximization is carried out using the BOBYQA algorithm (Powell 2009) implemented in the NLOPT package (Johnson 2014).

Table 2 shows that the EM algorithm with $M = 400$ is more accurate than both the spectral approximation and composite likelihood with 52 neighbors. Specifically, the RMSDs are 30%–90% lower for σ^2 and λ, and 90%–99% lower for μ, for our approach than the other approaches. Overall, the spectral approach is the least accurate, largely due to a negative bias for the sill and range parameters (not shown). This bias has been well documented (Guyon 1982), and many improvements have been proposed, including data tapering (Dalhaus and Künsch 1987). However, when we repeated the analysis using data tapering (not shown), the results did not improve much. Composite likelihood is more accurate than the spectral approximation, but less accurate than EM algorithm with $M = 400$. Composite likelihood performs worst for the missing disk design, likely due to a breakdown of the nearest-neighbor approximation with large blocks of missing data.

We note that the EM algorithm with $M = 400$ is substantially slower than the other two methods. When the methods were rerun with inputs chosen to give similar run times (not shown),

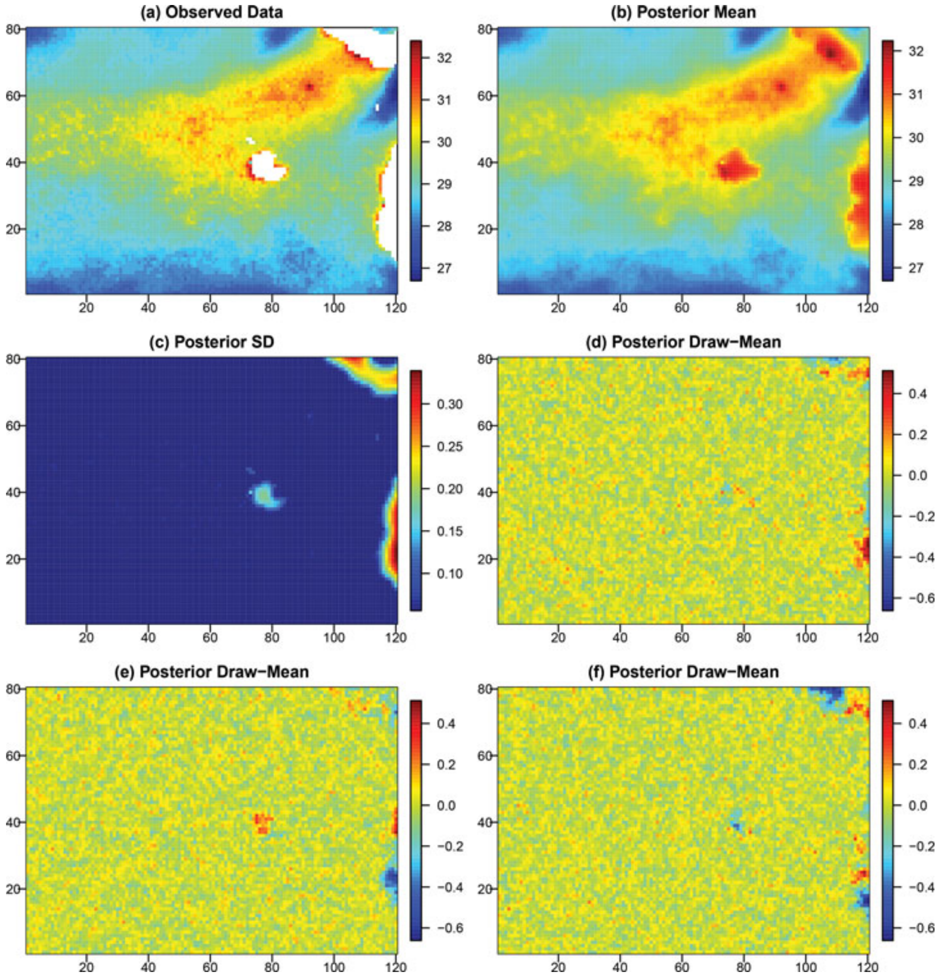

Figure 5. Satellite sea surface temperatures from the TMI satellite data. (a) Observed data $Y(\mathbf{s})$. (b) Posterior mean for $Z(\mathbf{s})$. (c) Posterior standard deviation for $Z(\mathbf{s})$. (d)–(f) Posterior samples of $Z(\mathbf{s})$ minus the posterior mean for $Z(\mathbf{s})$.

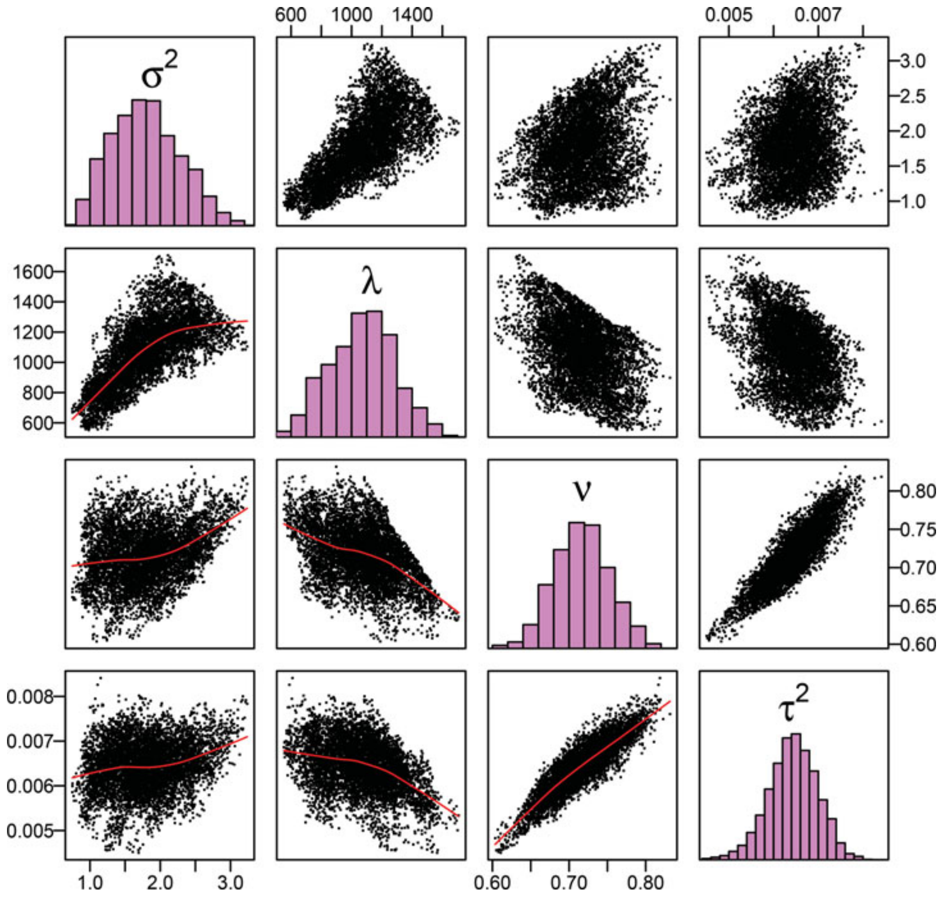

Figure 6. TMI satellite data. Posterior histograms and pairs plots for the Matérn covariance parameters. Note that λ are in units of km, which are obtained by multiplying their original distance units by (144.2 pixels/distance unit) · (25 km/pixel) = 3605 km/distance unit.

composite likelihood was generally slightly more accurate than the EM algorithm.

4.2 Application to Satellite Sea Surface Temperatures

Figure 5(a) shows a satellite image of sea surface temperatures over the Pacific Ocean, obtained from the Tropical Rainfall Measuring Mission's (TRMM) Thermal Microwave Imager (TMI). The data represent an average of 31 daily images of sea surface temperatures in March 1998. The observation region is 120 × 80 pixels, and there are 4% missing data (corresponding to land locations in Central and South America and the Galapagos Islands). The total number of observations is $n = 9203$. This dataset was previously analyzed by Fuentes (2007), who used spectral ML methods to fit a stationary Gaussian process to the data.

To compare our results to Fuentes (2007), we model the observed data $Y(\mathbf{s})$ as a stationary process plus measurement error. That is, we assume that $Y(\mathbf{s}) = Z(\mathbf{s}) + \varepsilon(\mathbf{s})$, where $\varepsilon(\mathbf{s})$ is a white-noise process with mean zero and variance τ^2, and $Z(\mathbf{s})$ is a stationary, isotropic Gaussian process with mean μ and covariance $C(h) = \sigma^2 K(h)$, where $K(h)$ is the Matérn correlation function (see Stein 1999),

$$K(h) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{h}{\lambda}\right)^{\nu} \mathcal{K}_\nu\left(\frac{h}{\lambda}\right). \tag{21}$$

Here σ^2 and λ are the sill and range parameters, $\nu > 0$ is the smoothness parameter, $\mathcal{K}_\nu$ is a modified Bessel function of the third kind of order ν (see Abramowitz and Stegun 1964, sec. 9). and τ^2 is the so-called nugget effect. Note that the model in Fuentes (2007) also included two anisotropy parameters; however, both parameters were found to be insignificant, implying that the process is isotropic. We implement the Bayesian MCMC approach described in Sections 3.1 and 4.1.1 to estimate the parameters μ, σ^2, and $\boldsymbol{\theta} = (\lambda, \nu, c)$, where $c = \tau^2/\sigma^2$ is the noise-to-signal ratio. We assume the following independent prior distributions: $\pi(\mu, \sigma^2) \propto \sigma^{-2}$, $\pi(\lambda) \propto 0.5(1 + 0.5\lambda)^{-2}$, $\nu \sim \mathcal{U}(0, 50)$, and $c \sim \mathcal{U}(0, 10)$.

Because of the nonsquare observation lattice, our embedding approach needs to be modified slightly. We embed the original 120 × 80 lattice in a square lattice of size 320 × 320. We define 1 distance unit to be the maximum distance in the original domain, $\sqrt{120^2 + 80^2} \approx 144.2$ pixels. We then apply the embedding approach as in Section 4.1 with a radius of $r = 1.5/\sqrt{2} \approx 1.06$. This yields a minimum embedding lattice size of $2r \cdot 144.2 \approx 305.7$, which is rounded up to the next highly composite integer of 320. This choice of radius and embedding lattice led to a small number of negative eigenvalues in the sampler. We discarded these parameter draws, under the prior assumption that the parameters are constrained to values that yield nonnegative definite embeddings. The MCMC algorithm described in Section 4.1.1 was run for 65,000 iterations after a burn-in period of 35,000.

Table 3. TMI satellite data: comparison of parameter estimates (standard errors) for the spectral MLE results reported by Fuentes (2007), from our Bayesian MCMC approach, and the exact MLEs.

Method	σ^2	λ	σ^2/λ	ν	τ^2	Loglike	Time
Spectral MLE	0.57 (0.02)	312 (70)	0.0018 (0.0004)	1.00 (1.20)	0.0010 (0.0020)	− 2077.11	10 min
MCMC Bayes	1.54 (0.40)	953 (215)	0.0016 (0.0003)	0.72 (0.04)	0.0066 (0.0005)	5528.13	19 hr
Exact MLE	1.45 (0.52)	911 (312)	0.0016 (0.0004)	0.72 (0.04)	0.0066 (0.0006)	5528.19	55 min

NOTE: Standard errors for σ^2/λ for the spectral and exact MLEs are obtained using the delta method. The last two columns show the exact log-likelihood for each estimate, and the computational run time.

Figure 5 shows the observed data $Y(\mathbf{s})$, and the posterior mean, standard deviation, and three draws of the latent Gaussian process, $Z(\mathbf{s})$. Since the estimated nugget effect is relatively small, the posterior mean for $Z(\mathbf{s})$ closely matches the observed data $Y(\mathbf{s})$ at the observed locations, but the process appears much smoother. The posterior standard deviation and draws illustrate the posterior uncertainty for $Z(\mathbf{s})$, which is small except over the land regions. Figure 6 shows the posterior histograms and scatterplots for the parameters. Note that the posterior distributions are all unimodal and fairly symmetric. There are dependencies between the parameters, most notably a strong positive correlation between the smoothness parameter ν and the nugget effect τ^2.

Table 3 compares the posterior means and standard deviations for the parameters from our MCMC approach with the spectral MLEs and standard errors reported in Fuentes (2007). For comparison we also calculate the exact MLEs and their asymptotic standard errors, which are (just barely) computable for the sample size of $n = 9203$. Note that our results differ substantially from the spectral method: our posterior mean estimates are $\hat{\sigma}^2 = 1.54$, $\hat{\lambda} = 953$ km, $\hat{\nu} = 0.72$, and $\hat{\tau}^2 = 0.006$, whereas the estimates from Fuentes (2007) are $\hat{\sigma}^2 = 0.57$, $\hat{\lambda} = 312$ km, $\hat{\nu} = 1.00$, and $\hat{\tau}^2 = 0.001$. However, the estimates of σ^2/λ, which relates to the fine scale variation of the process, are similar for the two methods. The standard errors are also quite different for the two approaches, in particular for σ^2 (0.40 vs. 0.02) and ν (0.04 vs. 1.20). In contrast, the estimates and standard errors from our approach agree closely with the exact ML values for all parameters. This is quite reassuring. As a final comparison, we compute the exact log-likelihood values for the three sets of estimates. The log-likelihood for the spectral estimates is −2077, while the MCMC estimates and exact MLEs have nearly identical log-likelihoods of 5528. Thus, our Bayesian posterior mean estimate provides an improvement of 7605 log-likelihood points over the spectral MLE.

5. Conclusions

In this article, we have proposed a new approach for Bayesian inference and MLE for stationary Gaussian processes observed on a large, possibly incomplete, lattice. We show that the method is feasible for large datasets (lattices of up to size 512 × 512), allows for missing data or irregular boundaries, and provides accurate inference for the parameters and missing values. We propose an MCMC algorithm for Bayesian inference and a Monte Carlo EM algorithm for MLE.

The proposed algorithms are conceptually simple and widely applicable. The main requirements of the algorithms are: (1) specification of a periodic covariance function and embedding lattice to ensure a positive-definite covariance matrix $\mathbf{C}$; (2) choice of a fast and accurate preconditioner; and (3) efficient computer code to implement the method. To address the second point, we have developed a number of new preconditioners, and find that two work quite well: the observed block of $\mathbf{C}^{-1}$, and an incomplete Cholesky preconditioner based on the composite likelihood methods of Vecchia (1988) and Stein, Chi, and Welty (2004). To address the last point, we have developed efficient R code, which will be made available online.

There are many potential extensions of this work, including estimation of anisotropic, nonstationary, and/or non-Gaussian processes, multivariate processes, and spatio-temporal models. We have successfully implemented the approach for anisotropic processes; however, the results are not reported here due to space limitations. Finally, we are presently exploring the use of parallel computing to improve efficiency of the algorithms.

Supplementary Materials

Appendix A: Provides a summary of BCCB matrices. **Appendix B:** Summarizes the preconditioned conjugate gradient (PCG) algorithm. **Appendix C:** Describes the Vecchia preconditioner used in the PCG algorithm. **R Code:** Code to implement the MCMC and MCEM algorithms for an exponential covariance. **Dataset:** ASCII file with the TMI sea surface temperature data

Acknowledgments

Stein's work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0011097. The authors thank the editor, associate editor, and three referees for their careful reading and excellent comments that greatly improved the article.

References

Abramowitz, M., and Stegun, I. A. (1964), *Handbook of Mathematical Functions*, New York: Dover. [118]

Agarwal, D. K., and Gelfand, A. E. (2005), "Slice Sampling With Application to Spatial Data," *Statistics and Computing*, 15, 61–69. [112]

Banerjee, S., Gelfand, A., Finley, A., and Sang, H. (2008), "Gaussian Predictive Process Models for Large Spatial Data Sets," *Journal of the Royal Statistical Society*, Series B, 70, 825–848. [108]

Berger, J. O., De Oliveira, V., and Sansó, B. (2001), "Objective Bayesian Analysis of Spatially Correlated Data," *Journal of the American Statistical Association*, 96, 1361–1374. [113]

Chilès, J.-P., and Delfiner, P. (2012), *Geostatistics: Modeling Spatial Uncertainty* (2nd ed.), New York: Wiley. [111]

Cressie, N. (1993), *Statistics for Spatial Data* (Rev. ed.), New York: Wiley-Interscience. [113]

Cressie, N., and Johannesson, G. (2008), "Fixed Rank Kriging for Very Large Spatial Data Sets," *Journal of the Royal Statistical Society*, Series B, 70, 209–226. [108]

Dalhaus, R., and Künsch, H. (1987), "Edge Effects and Efficient Parameter Estimation for Stationary Random Fields," *Biometrika*, 74, 877–882. [117]

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood From Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society,* Series B, 39, 1–38. [112]

Dietrich, C., and Newsam, G. (1997), "Fast and Exact Simulation of Stationary Gaussian Processes Through Circulant Embedding of the Covariance Matrix," *SIAM Journal of Scientific Computation*, 18, 1088–1107. [109,110]

Ecker, M. D., and Gelfand, A. E. (1997), "Bayesian Variogram Modeling for an Isotropic Spatial Process," *Journal of Agricultural, Biological and Environmental Statistics*, 2, 347–369. [112]

Eidsvik, J., Shaby, B. A., Reich, B. J., Wheeler, M., and Niemi, J. (2014), "Estimation and Prediction in Spatial Models With Block Composite Likelihood," *Journal of Computational and Graphical Statistics*, 23, 295–315. [108]

Frigo, M., and Johnson, S. G. (1998), "FFTW: An Adaptive Software Architecture for the FFT," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1381–1384. [114]

Fuentes, M. (2007), "Approximate Likelihood for Large Irregularly Spaced Data," *Journal of the American Statistical Association*, 102, 321–331. [108,116,118,119]

Gneiting, T., Ševčíková, H., Percival, D. B., Schlather, M., and Jiang, Y. (2006), "Fast and Exact Simulation of Large Gaussian Lattice Systems in $\mathbb{R}^2$: Exploring the Limits," *Journal of Computational and Graphical Statistics*, 15, 483–501. [109,110]

Golub, G., and Van Loan, C. (1996), *Matrix Computations*, Baltimore, MD: Johns Hopkins Press. [111]

Guyon, X. (1982), "Parameter Estimation for a Stationary Process on a *d*-Dimensional Lattice," *Biometrika*, 69, 95–105. [117]

Handcock, M. S., and Stein, M. L. (1993), "A Bayesian Analysis of Kriging," *Technometrics*, 35, 403–410. [112,113]

Handcock, M. S., and Wallis, J. R. (1994), "An Approach to Statistical Spatial-Temporal Modeling of Meteorological Fields," *Journal of the American Statistical Association*, 89, 368–378. [113]

Huerta, G., Sansó, B., and Stroud, J. R. (2004), "A Spatio-Temporal Model for Mexico City Ozone Levels," *Journal of the Royal Statistical Society,* Series C, 53, 231–248. [114]

Johnson, S. G. (2014), "The NLopt Nonlinear-Optimization Package," available at *http://ab-initio.mit.edu/nlopt*. [117]

Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008), "Covariance Tapering for Likelihood-Based Estimation in Large Spatial Data Sets," *Journal of the American Statistical Association*, 103, 1545–1555. [108]

Kozintsev, B. (1999), "Computations With Gaussian Random Fields," Ph.D. thesis, University of Maryland. [110]

Kozintsev, B., and Kedem, B. (2000), "Generation of 'Similar' Images From a Given Discrete Image," *Journal of Computational and Graphical Statistics*, 9, 286–302. [109]

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" (with discussion), *Journal of the Royal Statistical Society,* Series B, 73, 423–498. [108]

Liu, J. S., Wong, W. H., and Kong, A. (1994), "Covariance Structure of the Gibbs Sampler With Applications to the Comparisons of Estimators and Augmentation Schemes," *Biometrika*, 81, 27–40. [109]

Matheron, G. (1976), "A Simple Substitute for Conditional Expectation: The Disjunctive Kriging," in *Advanced Geostatistics in the Mining Industry*, eds. M. Guarascio, C. J. Huybrechts, and M. David, Dordrecht: Reidel, pp. 221–236. [111]

Powell, M. J.D. (2009), "The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives," Technical Report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK. [117]

Rue, H., and Held, L. (2005), *Gaussian Markov Random Fields: Theory and Applications*, Boca Raton, FL: Chapman & Hall/CRC. [108]

Sang, H., and Huang, J. Z. (2012), "A Full-Scale Approximation of Covariance Functions for Large Spatial Data Sets," *Journal of the Royal Statistical Society,* Series B, 74, 111–132. [108]

Stein, M. L. (1999), *Interpolation of Spatial Data*, New York: Springer. [115,118]

——— (2002), "Fast and Exact Simulation of Fractional Brownian Surfaces," *Journal of Computational and Graphical Statistics*, 11, 587–599. [109,110]

Stein, M. L., Chen, J., and Anitescu, M. (2013), "Stochastic Approximation of Score Functions for Gaussian Processes," *Annals of Applied Statistics*, 7, 1162–1191. [108]

Stein, M. L., Chi, Z., and Welty, L. J. (2004), "Approximating Likelihoods for Large Spatial Datasets," *Journal of the Royal Statistical Society,* Series B, 66, 275–296. [108,111,116,119]

Vecchia, A. V. (1988), "Estimation and Model Identification for Continuous Spatial Processes," *Journal of the Royal Statistical Society,* Series B, 50, 297–312. [108,111,116,119]

Whittle, P. (1954), "On Stationary Processes in the Plane," *Biometrika*, 41, 43–49. [108,116]

Wood, A. T.A., and Chan, G. (1994), "Simulation of Stationary Gaussian Processes in $[0, 1]^d$," *Journal of Computational and Graphical Statistics*, 3, 409–432. [109,110]

Zimmerman, D. (1989), "Computationally Efficient Restricted Maximum Likelihood Estimation of Generalized Covariance Functions," *Mathematical Geology*, 21, 655–672. [108,109]