Statistics and Computing (2022) 32:74
https://doi.org/10.1007/s11222-022-10104-3

Kryging: geostatistical analysis of large-scale datasets using Krylov subspace methods

Suman Majumder[1,2] · **Yawen Guan**[3] · **Brian J. Reich**[1] · **Arvind K. Saibaba**[4]

Received: 15 June 2021 / Accepted: 7 May 2022

Abstract
Analyzing massive spatial datasets using a Gaussian process model poses computational challenges. This is a problem prevailing heavily in applications such as environmental modeling, ecology, forestry and environmental health. We present a novel approximate inference methodology that uses profile likelihood and Krylov subspace methods to estimate the spatial covariance parameters and makes spatial predictions with uncertainty quantification for point-referenced spatial data. "Kryging" combines Kriging and Krylov subspace methods and applies for both observations on regular grid and irregularly spaced observations, and for any Gaussian process with a stationary isotropic (and certain geometrically anisotropic) covariance function, including the popular Matérn covariance family. We make use of the block Toeplitz structure with Toeplitz blocks of the covariance matrix and use fast Fourier transform methods to bypass the computational and memory bottlenecks of approximating log-determinant and matrix-vector products. We perform extensive simulation studies to show the effectiveness of our model by varying sample sizes, spatial parameter values and sampling designs. A real data application is also performed on a dataset consisting of land surface temperature readings taken by the MODIS satellite. Compared to existing methods, the proposed method performs satisfactorily with much less computation time and better scalability.

Keywords Approximate inference · Profile likelihood · Block Toeplitz matrix · Fast Fourier transform · Krylov subspace methods · Golub–Kahan bidiagonalization

1 Introduction

Massive spatial datasets, often coming from satellites or other remotely sensed sources, have become increasingly common in applications such as environmental health, forestry, and ecology. Classical geostatistical analysis methods for point-referenced spatial data are burdened with computationally intensive steps such as Cholesky factorization or eigendecomposition, which have cubic complexity in the number of observations. Despite the advances in computing performance, these methods remain prohibitively expensive to apply to datasets of even moderately large size. Therefore, we need to develop methods that perform nearly as well as the classical methods but are more computationally efficient and therefore applicable to problems of massive volume.

There is a rich literature of approximate inference methods for point-referenced spatial data. Early approaches approximated the joint likelihood by decomposing it into a product of conditional distributions (Vecchia 1988; Stein et al. 2004), using pseudo-likelihood (Varin et al. 2011; Eidsvik et al. 2014) or using covariance tapering (Furrer et al. 2006; Kaufman et al. 2008; Stein 2013). Modeling in the spectral domain (Fuentes 2007; Guinness and Fuentes 2017; Guinness 2019) was also used to circumvent the heavy computation. Another class of approaches are based on finite-rank approximations

✉ Suman Majumder
smajumder@hsph.harvard.edu

Yawen Guan
yguan12@unl.edu

Brian J. Reich
bjreich@ncsu.edu

Arvind K. Saibaba
asaibab@ncsu.edu

[1] Department of Statistics, North Carolina State University, Raleigh, USA

[2] Present Address: Department of Nutrition, Harvard T.H. Chan School of Public Health, Boston, USA

[3] Department of Statistics, University of Nebraska-Lincoln, Lincoln, USA

[4] Department of Mathematics, North Carolina State University, Raleigh, USA

Published online: 08 September 2022

such as fixed-rank Kriging (Cressie and Johannesson 2008; Kang and Cressie 2011; Katzfuss and Cressie 2011), predictive process (Banerjee et al. 2008; Finley et al. 2009), process convolution (Higdon 2002) and lattice Kriging (Nychka et al. 2015). Other approaches use a combination of hierarchical matrix approaches and stochastic estimators for the log-likelihood (Anitescu et al. 2012; Ambikasaran et al. 2015; Minden et al. 2017; Eriksson et al. 2018; Stein 2013) or spectral methods and h-likelihood (Dutta and Mondal 2016).

More recent approaches make use of the modern computing platforms and focus on parallelizing the computational load. Paciorek et al. (2015) is one such example. Katzfuss (2017) and Katzfuss and Hammerling (2017) combine low-rank methods with distributed computing. Dividing the data into subsets, drawing inference on these subsets in parallel and recombining them has been proposed by Barbian and Assunção (2017) and Guhaniyogi and Banerjee (2018). Datta et al. (2016a, b, c) use an approximation based on the conditional distribution given the nearest neighbors, inducing sparsity and allowing the method to be parallelized. The stochastic partial differential equation or SPDE (Lindgren et al. 2011) approach induces sparsity in the inverse-covariance matrix for fast approximations. Sun et al. (2012), Bradley et al. (2016), Heaton et al. (2019) and Liu et al. (2020) provide comprehensive reviews of these methods and demonstrate their effectiveness in spatial modeling.

Most of these methods use either finite-rank approximations or introduce sparsity in the covariance or the inverse-covariance structure. Finite rank-based models typically have complexity $\mathcal{O}(nr^2 + r^3)$ with r being the rank of the model such that $r \ll n$. However, in order for the approximation to be effective for large n, a large rank r is needed, which increases the computational costs. This cost can be alleviated by inducing sparsity into the covariance structure using compactly supported covariance function; however, this may not be an appropriate modeling choice when long-range dependence is present in the data.

We present a novel statistical method of log-linear complexity to provide approximate inference for massive geostatistical datasets using profile maximum likelihood estimation and Krylov subspace methods based on the genHyBR method proposed by Chung et al. (2018). The proposed method, a combination of kriging and Krylov subspaces and hence dubbed "Kryging", provides prediction for the observed process at unobserved locations by approximating the underlying spatial process on a regular, equispaced grid. Although we approximate the latent process on a grid, we do not restrict the observations to be on grid and therefore the method can be applied to irregularly spaced large spatial datasets. We generate estimates of the underlying process through Krylov subspace methods. Krylov subspaces (see Saad 2003, for reference) are efficient iterative methods for solving large-scale linear systems and least-square problems. A key advantage of the Krylov subspace approach is that it is matrix-free, in that it does not require forming the matrices explicitly, but only requires the action of the matrix on appropriate vectors. We provide prediction uncertainty estimates in the form of pointwise 95% confidence intervals via a parametric bootstrap approach and estimates for the mean and spatial covariance parameters. Kryging applies to any stationary isotropic covariance structure, e.g., the Matérn covariance family, as well as covariance functions that incorporate geometric anisotropy by allowing dissimilar stretching along the two axes. It exploits the Toeplitz (in one dimension) or block Toeplitz with Toeplitz blocks (BTTB) structure (in higher dimensions) of the resulting covariance matrices and employs a fast Fourier transformation-based method for achieving computational gains for matrix-vector multiplications (See Gray 2006) and approximating log determinants (Kent and Mardia 1996). As a result, Kryging has $\mathcal{O}(n)$ storage costs and only $\mathcal{O}(n \log n)$ computational complexity where n is the size of the underlying grid for estimating the spatial parameters and performing spatial prediction.

The tools used for building the Kryging model have been used in the literature before in different contexts and different problems. However, by efficiently combining them in a specific manner, Kryging has several advantages compared to related methods in the literature. Chung et al. (2018) also use the same core method, but we extend it to include mean and spatial covariance parameter estimation, uncertainty quantification and approximation of log-determinants. Aune et al. (2014) and Dutta and Mondal (2016) also use tools such as Krylov subspaces and the fast Fourier transformation, but their usage differs vastly from ours. First, we construct a different Krylov subspace, one that incorporates the noise covariance, a mapping matrix, and the covariance matrix; in contrast, the approach in the other papers is to build a Krylov subspace method with the covariance matrix alone. Second, we use the Golub–Kahan bidiagonalization rather than Lanczos or conjugate gradient for linear systems. Third, we use the basis vectors from the Krylov subspace to estimate the objective function and the gradients (one exception is the determinant and its derivative for which we use a different approximation). In contrast, other approaches use various tools such as Monte Carlo trace estimators, to estimate the various quantities.

Kryging has a low-rank matrix involved in the approximation process. However, compared to other low-rank methods discussed above, empirical evidence hints that using a small order of the Krylov subspace works well for huge datasets and produces accurate results. Block-circulant embeddings have been proposed as a stand-alone method to approximate determinants (Rue and Held 2005), which nicely gels with the Krylov subspace-based approximation to the problem of maximizing the quadratic part of a Gaussian log-likelihood

to produce a fast and scalable approximate inference method for massive geostatistical datasets.

We establish the particular form of latent Gaussian model that we use for our method in Sect. 2. Section 3 gives the details of the method. We provide detailed description and algorithms of components of the method in various subsections of Sect. 3. A thorough simulation study is performed in Sect. 4, and an application to MODIS satellite data is performed in Sect. 5. The data analysis is based on Heaton et al. (2019). The rationale behind this was to be able to compare the performance of our method to other available methods directly. We finish with a discussion and concluding remarks in Sect. 6.

2 Latent Gaussian model

Let $y(\mathbf{s})$ be the observed process, and $x(\mathbf{s})$ is the underlying process of interest at location $\mathbf{s} \in \mathbb{R}^d$, $d \geq 1$; throughout this paper, we illustrate the methods using the $d = 2$ but our approach is applicable to problems with two or three spatial dimensions with a possible additional time dimension. A realization from the observation process, $\mathbf{y} = [y(\mathbf{s}_1), \ldots, y(\mathbf{s}_p)]^\mathsf{T}$, at p locations $\mathbf{s}_1, \ldots, \mathbf{s}_p$ is related to a realization from the latent process, $\mathbf{x} = [x(\mathbf{s}_1^*), \ldots, x(\mathbf{s}_n^*)]^\mathsf{T}$, at n possibly different locations $\mathbf{s}_1^*, \ldots, \mathbf{s}_n^*$ by the relationship

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{R})$ with $\mathbf{R}$ being a cheaply invertible matrix of individual variances for each location, $\mathbf{X}$ being the matrix of corresponding covariates observed at the same locations as the observations themselves and $\mathbf{A}$ being a matrix that specifies the linear combinations that connect the mean removed $\mathbf{y}$ and $\mathbf{x}$. For this paper, we make the standard assumption that the nugget variance is constant across space and set $\mathbf{R} = \tau^2\mathbf{I}_p$.

The mapping matrix $\mathbf{A}$ permits the flexibility of $\mathbf{y}$ and $\mathbf{x}$ not being co-located, as well as change of support. For example, $\mathbf{A} = \mathbf{I}$, the identity matrix, represents the case where $\mathbf{y}$ is a noisy observation of $\mathbf{x}$ itself after accounting for the mean process. In case the response locations are a subset of the n locations $\mathbf{s}_1^*, \ldots, \mathbf{s}_n^*$, then $\mathbf{A}$ is the $n \times n$ identity matrix with $n - p$ rows removed. The matrix $\mathbf{A}$ can be non-diagonal as well, for the case when value of $\mathbf{y}$ at each location is considered as an average of the unobserved $\mathbf{x}$ at nearby locations, as it can be when $y(\mathbf{s})$ is observed at locations at irregularly spaced locations and $x(\mathbf{s})$ is considered on a grid around those locations.

When the observations are not on a regular grid, we still set the latent process locations $\mathbf{s}_1^*, \ldots, \mathbf{s}_n^*$ to be on a rectangular grid and account for the irregularity of the observation locations in the mapping matrix, $\mathbf{A}$. We specify the entries of $\mathbf{A}$ so that each observation is a convex combination of the latent process in the neighborhood of the observation. Specifically, the latent process is weighted by the Wendland kernel function (Wendland 1995) $w(d_{ij}) = (1 - d_{ij})_+^4 (1 + 4d_{ij})$, where $d_{ij} = \max\{|s_{i1} - s_{j1}^*|/\Delta_1, |s_{i2} - s_{j2}^*|/\Delta_2\}$ and $(x)_+ = \max\{x, 0\}$, Δ_1 and Δ_2 are the grid spacings in the two directions and $\mathbf{s}_i = (s_{i1}, s_{i2})$ and $\mathbf{s}_j^* = (s_{j1}^*, s_{j2}^*)$ are the i-th observation location and j-th grid-point location, respectively. This particular formulation allows to approximate the value at a point outside of the grid as a weighted combination of its nearest four neighbors, while for a point on the grid itself, the approximation is exact. To ensure the weights are convex, they are normalized to sum to one for each observation. That is, we assume the mean response is

$$E\{y(\mathbf{s}_i)\} = X(\mathbf{s}_i)^\mathsf{T}\boldsymbol{\beta} + \frac{\sum_{j=1}^n w(d_{ij})x(\mathbf{s}_j^*)}{\sum_{k=1}^n w(d_{ik})}.$$

This is equivalent to setting the (i, j) element of $\mathbf{A}$ to $w(d_{ij})/\{\sum_{k=1}^n w(d_{ik})\}$. The truncation function $(x)_+$ ensures that $\mathbf{A}$ is a sparse matrix with at most four nonzero entries per row, i.e., the matrix $\mathbf{A}$ has $\mathcal{O}(p)$ nonzero entries.

Choosing the mapping matrix to be sparse ensures there is not significantly higher computational cost due to these changes when applying to an irregularly spaced data. This approach to handling irregularly spaced observations introduces an additional tuning parameter, n, which controls the density of the latent space observations. When the observation locations are on a regular grid, we simply set it to be equal to p so that the latent process locations match the observations. However, when the observations are not on a grid, then there is no natural choice for n. Accuracy should increase with n at the expense of computational burden. This issue is explored further in the simulation study of Sect. 4.

We use a latent Gaussian process to model the true state $\mathbf{x}(\mathbf{s})$, with zero mean and isotropic Matérn covariance kernel (Matérn 1960) with standard deviation σ, spatial range parameter ρ and smoothness parameter ν. Therefore, at finite collection of locations, $\mathbf{x}$ is a multivariate Gaussian distribution with mean $\mathbf{0}$ and $n \times n$ correlation matrix $\boldsymbol{\Sigma}$, i.e.,

$$\mathbf{x} \sim N\left(\mathbf{0}, \sigma^2\boldsymbol{\Sigma}\right), \tag{2}$$

with $\mathbf{0}$ being the vector of all zeros and

$$\boldsymbol{\Sigma}_{ij} = \frac{2^{1-\nu}}{\boldsymbol{\Gamma}(\nu)}\left(\sqrt{2\nu}\frac{d_{ij}}{\rho}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{d_{ij}}{\rho}\right)$$

being the spatial correlation between locations i and j induced by the stationary isotropic Matérn covariance kernel for $i, j = 1, \ldots, n$. Here $d_{ij} = \|\mathbf{s}_i - \mathbf{s}_j\|_2$ and $\|\cdot\|_2$ denotes the Euclidean norm in $\mathbb{R}^2$ and $K_\nu(\cdot)$ is the modified Bessel function of the second kind with parameter ν. The

choice of Matérn covariance kernel is common, but any other stationary covariance function (or geometrically anisotropic covariance function that induces different stretching along the two axes) may be used along with the approach for both regularly gridded and irregularly spaced datasets with same computational complexity that we outline in the next section.

3 Inferential approach

In this section, we describe an inferential approach for the latent Gaussian model that combines Kriging and Krylov subspace methods, which we have been calling "Kryging". The likelihood function for the latent state $\mathbf{x}$, the mean parameter, the spatial variance parameters $\boldsymbol{\theta} = \left(\boldsymbol{\beta}^\top, \sigma^2, \tau^2, \rho\right)^\top$ and a given smoothness parameter ν can be written as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}; \mathbf{y}, \nu) = f_{\mathbf{y},\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) f_{\mathbf{x},\boldsymbol{\theta},\nu}(\mathbf{x}|\boldsymbol{\theta}; \nu), \tag{3}$$

where $f_{\mathbf{y},\boldsymbol{\theta}}(\cdot|\mathbf{x})$ is the density of the data given $\mathbf{x}$ and $f_{\mathbf{x},\boldsymbol{\theta};\nu}(\cdot)$ is the density of $\mathbf{x}$; both densities depend on the parameters $\boldsymbol{\theta}$ and ν. Since we assumed a Gaussian model for $\mathbf{y}|\mathbf{x}$ and $\mathbf{x}$, we have

$$\log f_{\mathbf{y},\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \simeq -\frac{p}{2}\log\tau^2 - \frac{1}{2\tau^2}\boldsymbol{\psi}^\top\boldsymbol{\psi},$$

where $\simeq$ means equal up to a constant that is unimportant for the purposes of optimization and $\boldsymbol{\psi} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{A}\mathbf{x}$ and

$$\begin{aligned}\log f_{\mathbf{x},\boldsymbol{\theta}}(\mathbf{x}) \simeq &-\frac{n}{2}\log\sigma^2 - \frac{1}{2}\log\det\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}; \nu)\right) \\ &- \frac{1}{2\sigma^2}\mathbf{x}^\top\boldsymbol{\Sigma}(\boldsymbol{\theta}; \nu)^{-1}\mathbf{x}.\end{aligned} \tag{4}$$

Thus, the log-likelihood function, $l(\mathbf{x}, \boldsymbol{\theta}; \nu) = \log \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}; \mathbf{y}, \nu)$, has the form

$$\begin{aligned}l(\mathbf{x}, \boldsymbol{\theta}; \nu) \simeq &-\frac{p}{2}\log\tau^2 - \frac{1}{2\tau^2}\boldsymbol{\psi}^\top\boldsymbol{\psi} \\ &- \frac{n}{2}\log\sigma^2 - \frac{1}{2}\log\det\boldsymbol{\Sigma}(\boldsymbol{\theta}; \nu) \\ &- \frac{1}{2\sigma^2}\mathbf{x}^\top\boldsymbol{\Sigma}(\boldsymbol{\theta}; \nu)^{-1}\mathbf{x}.\end{aligned} \tag{5}$$

Evaluation of the log-likelihood function involves inverting and computing the log-determinant of the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta}; \nu)$, both of which require $\mathcal{O}(n^3)$ many operations which is not feasible for large n. Since the optimization needs to run on both $\mathbf{x}$ and $\boldsymbol{\theta}$, it would be a ultra high-dimensional optimization, which would generally be infeasible to implement. Therefore, running an optimization procedure over both $\boldsymbol{\theta}$ and $\mathbf{x}$ on this objective function straightaway is futile, and we must look into approximation methods to avoid these computational bottlenecks.

We propose a computationally efficient inference approach using approximate inference for fast estimation for both parameters $\boldsymbol{\theta}$ and the underlying true state variables $\mathbf{x}$ along with its uncertainty. We profile $\mathbf{x}$ as a function of the parameters $\boldsymbol{\theta}$ and maximize the corresponding profile likelihood over $\boldsymbol{\theta}$ (Cox and Snell 1989). This reduces the dimensionality of the optimization problem greatly, but it requires an estimate of $\mathbf{x}$ for a given value of $\boldsymbol{\theta}$. Since we consider ν to be a given and known value, we drop it from further notations in the rest of the paper.

The genHyBR method (Chung et al. 2018) circumvents the matrix inversion problem as it brings down the total complexity of computing the quadratic term to that of a matrix vector multiplication. Typically this would take $\mathcal{O}(n^2)$ operations. However, computational techniques such as fast Fourier transforms (FFTs) or $\mathcal{H}$-matrices (a review of techniques can be found in Ambikasaran et al. (2015)) can reduce the computational cost of storage and the mathematical operators to $\mathcal{O}(n \log^r n)$, where r is a nonnegative exponent which depends on the operation and the method used. In particular, we use the symmetric BTTB structure of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The symmetric BTTB structure allows us to store $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ in $\mathcal{O}(n)$, since only one row/column of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ needs to be stored and compute the matrix vector products involving $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ in $\mathcal{O}(n \log n)$ time. If the underlying process realizations are not on a regular grid, then the $\mathcal{H}$-matrix approach can be used instead with the same computational cost. However, with the mapping matrix strategy laid out in Sect. 2, we do not require this approach. The symmetric BTTB structure also allows us to compute the log-determinant of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ in $\mathcal{O}(n \log n)$ time. This gives us a good estimate for $\mathbf{x}$ for a given value of $\boldsymbol{\theta}$.

3.1 Profile likelihood

Maximizing the log-likelihood function in Eq. (5) as a function of both $\mathbf{x}$ and $\boldsymbol{\theta}$ is not feasible, and therefore, we use a profile likelihood-based optimization strategy by profiling $\mathbf{x}$ as a function of $\boldsymbol{\theta}$. Profiling out $\mathbf{x}$ from Eq. (5) as a function of $\boldsymbol{\theta}$, in exact arithmetic, results in

$$\begin{aligned}\widehat{\mathbf{x}}(\boldsymbol{\theta}) = &\left(\frac{1}{\sigma^2}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} + \frac{1}{\tau^2}\mathbf{A}^\top\mathbf{A}\right)^{-1} \\ &\left(\frac{1}{\tau^2}\mathbf{A}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right).\end{aligned} \tag{6}$$

Plugging in $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ in Eq. (5) and calling $\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}) = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{A}\widehat{\mathbf{x}}(\boldsymbol{\theta})$ produces the exact profile log-likelihood function

$$\begin{aligned}\mathrm{pl}(\boldsymbol{\theta}) \simeq &-\frac{p}{2}\log\tau^2 - \frac{1}{2\tau^2}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})^\top\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}) \\ &-\frac{n}{2}\log\sigma^2 - \frac{1}{2}\log\det\boldsymbol{\Sigma}(\boldsymbol{\theta}) -\end{aligned}$$

$$\frac{1}{2\sigma^2}\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta}). \tag{7}$$

Since simply evaluating this function involves computing inverses and determinants of the dense covariance matrix, it must be approximated.

Evaluating the exact profile likelihood presents three computational challenges: (1) computing $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ involves inverting large dense $n \times n$ matrices, (2) computing the quadratic term $\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})$ and (3) computing the log-determinant of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The first two are overcome using the genHyBR method (Chung et al. 2018), while the log-determinant term is approximated using the symmetric BTTB structure of the resulting covariance matrix from the choice of appropriate covariance function previously mentioned in Sect. 2. Once these approximations are in place, the optimization of an approximated profile likelihood function can be performed using typical optimization routines to get the estimates of $\boldsymbol{\theta}$ and $\mathbf{x}$.

3.2 genHyBR method

A key component in maximizing the profile likelihood is to quickly compute $\widehat{\mathbf{x}}(\boldsymbol{\theta}) = \underset{\mathbf{x}}{\operatorname{argmin}}\ l(\mathbf{x}, \boldsymbol{\theta})$ for a given $\boldsymbol{\theta}$. The computation of $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ in this context is tantamount to computing

$$\widehat{\mathbf{x}}(\boldsymbol{\theta}) = \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{\tau^2}\|\boldsymbol{\psi}\|_2^2 + \frac{1}{\sigma^2}\|\mathbf{x}\|^2_{\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}}, \tag{8}$$

where $\|\mathbf{r}\|^2_{\mathbf{M}} = \mathbf{r}^{\mathsf{T}}\mathbf{M}\mathbf{r}$ and $\|\cdot\|_2$ represents the Euclidean norm. The genHyBR algorithm (Chung et al. 2018) solves this weighted least squares problem iteratively using generalized Golub–Kahan bidiagonalization which is a special type of Krylov subspace method (Benbow 1999; Chung and Saibaba 2017). To simplify notation, we drop the dependence on $\boldsymbol{\theta}$ and write $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$.

We provide an outline of the algorithm here. Denote $\mathcal{K}_k(\mathbf{M}, \mathbf{r}) = \operatorname{span}\{\mathbf{r}, \mathbf{M}\mathbf{r}, \ldots, \mathbf{M}^{k-1}\mathbf{r}\}$ as the Krylov subspace of degree k. Observing that Eq. (8) involves the inverse of $\boldsymbol{\Sigma}$, employing a change of variables $\mathbf{w} = \boldsymbol{\Sigma}^{-1}\mathbf{x}$ and $\mathbf{b} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$, we then compute $\widehat{\mathbf{x}}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}\widehat{\mathbf{w}}(\boldsymbol{\theta})$ and

$$\widehat{\mathbf{w}}(\boldsymbol{\theta}) = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{\tau^2}\|\mathbf{A}\boldsymbol{\Sigma}\mathbf{w} - \mathbf{b}\|_2^2 + \frac{1}{\sigma^2}\|\mathbf{w}\|^2_{\boldsymbol{\Sigma}}. \tag{9}$$

Then, for our problem of estimating $\mathbf{x}$, the genHyBR method (Chung et al. 2018) looks for the solution of $\mathbf{w}$ in

$$\mathcal{S}_k = \mathcal{K}_k\left(\frac{1}{\tau^2}\mathbf{A}^{\mathsf{T}}\mathbf{A}\boldsymbol{\Sigma}, \frac{1}{\tau^2}\mathbf{A}^{\mathsf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right).$$

The genHyBR algorithm creates an $n \times k$ basis $\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ for this subspace, i.e., $\mathcal{S}_k = \operatorname{span}\{\mathbf{V}_1, \ldots, \mathbf{V}_k\}$ using an efficient Golub–Kahan bidiagonalization iteration scheme, which is sketched in Algorithm 1.

Algorithm 1 Generalized Golub–Kahan (genGK) bidiagonalization

Ensure: Matrices $\mathbf{A}$, $\boldsymbol{\Sigma}$, vector $\mathbf{b} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ and τ^2.
1: Compute $\mathbf{u}_1 = \mathbf{b}/\beta_1$, where $\beta_1 = \|\mathbf{b}\|_2/\tau$.
2: Compute $\mathbf{v}_1 = \frac{1}{\tau^2}\mathbf{A}^{\mathsf{T}}\mathbf{u}_1/\alpha_1$ where $\alpha_1 = \|\frac{1}{\tau^2}\mathbf{A}^{\mathsf{T}}\mathbf{u}_1\|_{\boldsymbol{\Sigma}}$.
3: **for** $i = 1, \ldots, k$ **do**
4: Compute $\mathbf{u}_{i+1} = (\mathbf{A}\boldsymbol{\Sigma}\mathbf{v}_i - \alpha_i\mathbf{u}_i)/\beta_{i+1}$ where $\beta_{i+1} = \frac{1}{\tau^2}\|\mathbf{A}\boldsymbol{\Sigma}\mathbf{v}_i - \alpha_i\mathbf{u}_i\|_2$.
5: Compute $\mathbf{v}_{i+1} = \left(\mathbf{A}^{\mathsf{T}}\mathbf{u}_{i+1}/\tau^2 - \beta_{i+1}\mathbf{v}_i\right)/\alpha_{i+1}$ where $\alpha_{i+1} = \|\mathbf{A}^{\mathsf{T}}\mathbf{u}_{i+1}/\tau^2 - \beta_{i+1}\mathbf{v}_i\|_{\boldsymbol{\Sigma}}$.
6: **end for**
7: **return** β_1, $\mathbf{U}_{k+1}$, $\mathbf{V}_{k+1}$ and $\mathbf{B}_k$.

From Algorithm 1, we also obtain a $(k+1) \times k$ bidiagonal matrix

$$\mathbf{B}_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix}.$$

The outputs of the algorithms satisfy the following relationships

$$\begin{aligned} &\mathbf{A}\boldsymbol{\Sigma}\mathbf{V}_k = \mathbf{U}_{k+1}\mathbf{B}_k, \\ &\mathbf{U}_{k+1}^{\mathsf{T}}\mathbf{U}_{k+1} = \tau^2\mathbf{I}_{k+1}, \\ &\mathbf{V}_k^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{V}_k = \mathbf{I}_k. \end{aligned} \tag{10}$$

Since we are looking for a solution of $\mathbf{w} \in \mathcal{S}_k$, we can write $\mathbf{w}_k = \mathbf{V}_k\mathbf{z}_k$ and determine $\mathbf{z}_k$ by solving

$$\begin{aligned} &\min_{\mathbf{w}_k\in\mathcal{S}_k} \frac{1}{\tau^2}\|\mathbf{A}\boldsymbol{\Sigma}\mathbf{w}_k - \mathbf{b}\|_2^2 + \frac{1}{\sigma^2}\|\mathbf{w}_k\|^2_{\boldsymbol{\Sigma}} \\ &\Leftrightarrow \min_{\mathbf{z}_k\in\mathbb{R}^k} \|\mathbf{B}_k\mathbf{z}_k - \beta_1\mathbf{e}_1\|_2^2 + \frac{1}{\sigma^2}\|\mathbf{z}_k\|_2^2. \end{aligned} \tag{11}$$

Therefore, given $\mathbf{B}_k$ and $\mathbf{V}_k$ and by undoing the change of variables, we approximate the solution to Eq. (8) as

$$\mathbf{x}_k^*(\boldsymbol{\theta}) = \boldsymbol{\Sigma}\mathbf{V}_k\left(\mathbf{B}_k^{\mathsf{T}}\mathbf{B}_k + \frac{1}{\sigma^2}\mathbf{I}\right)^{-1}\mathbf{B}_k^{\mathsf{T}}\beta_1\mathbf{e}_1, \tag{12}$$

where $\mathbf{e}_1$ is the first column of the $(k+1) \times (k+1)$ identity matrix; that is, the vector with the first entry 1 and every other entry equal to 0. In general, a stopping criterion must be used to terminate the iterations and to automatically determine the number of iterations k. Details on one such choice of stopping criterion are given in Chung et al. (2018). However, we do not use the said criterion for our method and instead treat the parameter k as an algorithm parameter to be input by

the user. The orthogonal basis vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ may not remain numerically orthogonal and therefore may require a reorthogonalization scheme. Such a scheme is described in the Chung et al. (2018) paper and is available for the user to use in Kryging as well. However, we do not use it for the results presented in this paper.

The genHyBR method reduces the computational complexity of solving for $\mathbf{x}$ from $\mathcal{O}(n^3)$ to that of matrix vector multiplication, $\mathcal{O}(n^2 + nk^2)$. When the latent process locations $\mathbf{s}_1^*, \ldots, \mathbf{s}_n^*$ are arranged on a rectangular grid, $\boldsymbol{\Sigma}$ is symmetric BTTB, and thus, the matrix-vector multiplication can be achieved swiftly, in $\mathcal{O}(n \log n + nk^2)$ flops, using circulant embedding. Additionally, due to the form of $\mathbf{x}_k^*(\boldsymbol{\theta})$ in Eq. (12) and the exact arithmetic relationships presented in Eq. (10), the quadratic term $\widehat{\mathbf{x}}(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} \widehat{\mathbf{x}}(\boldsymbol{\theta})$ can now be approximated as $\|\mathbf{z}_k^*\|_2^2$, where $\mathbf{z}_k^* = \left(\mathbf{B}_k^\top \mathbf{B}_k + \frac{1}{\sigma^2}\mathbf{I}\right)^{-1} \mathbf{B}_k^\top \beta_1 \mathbf{e}_1$. This requires only $\mathcal{O}(k^3)$ operations.

3.3 Log-determinant approximation

To compute the log determinant of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, we once again use the symmetric BTTB structure of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. Gray (2006) reviews methods for creating a circulant matrix based on a Toeplitz matrix and using the circulant matrix structure to approximate the log-determinant of a Toeplitz matrix using inverse FFTs. Refer to Section 4.1, 4.4 and 5.3 of Gray (2006) for details. This behavior can be extended to a symmetric BTTB structure as well and a similar asymptotic result also holds for them (Gyires 1956; Widom 1974). The block circulant matrix $C = \left((C_{jk})\right)_{(2n_1-1)\times(2n_2-1)}$ can be created exactly as it is done for circulant embedding-based matrix-vector product and therefore does not add any extra computation. The approximation to the log-determinant is of the form

$$\widetilde{\log\det}\,\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \sum_{p=1}^{n_1}\sum_{q=1}^{n_2} \log\left(\sum_{j=1}^{2n_1-1}\sum_{k=1}^{2n_2-1} \omega_{n_1}^{(j-1)(p-1)}\omega_{n_2}^{(k-1)(q-1)} C_{jk}\right),$$

where $\omega_{n_1} = \exp(-2\pi i/(2n_1-1))$ and $\omega_{n_2} = \exp(-2\pi i/(2n_2-1))$.

The approximation stems from the fact that the result is only exact in an asymptotic sense. However, numerical evidence suggests that the approximation to the log determinant and its derivatives improves as the number of grid points n increases; a more precise statement of convergence can be found in Theorem 1.1 and Lemma 4.1(b) of Kent and Mardia (1996). We mention that besides the BCCB approximation, there are other ways of estimating the log-determinant, such as stochastic trace estimation (Anitescu et al. 2012; Ubaru et al. 2017) and using Hierarchical matrix structure (Ambikasaran et al. 2015; Minden et al. 2017). In particular, the advantage of the stochastic trace estimator is that the information used in estimating the log-determinant can be reutilized during the computation of the gradient information. These approaches can be used in place of the proposed estimator.

3.4 Optimization details

The approximations described in the previous sections render the approximate profile log-likelihood function $\widetilde{\text{pl}}(\boldsymbol{\theta})$ to have the form

$$\begin{aligned}\widetilde{\text{pl}}(\boldsymbol{\theta}) \simeq & -\frac{p}{2}\log\tau^2 - \frac{1}{2\tau^2}\boldsymbol{\psi}_k^*(\boldsymbol{\theta})^\top \boldsymbol{\psi}_k^*(\boldsymbol{\theta}) \\ & - \frac{n}{2}\log\sigma^2 - \frac{1}{2}\widetilde{\log\det}\,(\boldsymbol{\Sigma}(\boldsymbol{\theta})) \\ & - \frac{1}{2\sigma^2}\|\mathbf{z}_k\|_2^2,\end{aligned} \tag{13}$$

where $\boldsymbol{\psi}_k^* = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{A}\mathbf{x}_k^*(\boldsymbol{\theta})$, $\mathbf{x}_k^*(\boldsymbol{\theta})$ and $\mathbf{z}_k^*$ are described in Sect. 3.2 and $\widetilde{\log\det}\,(\boldsymbol{\Sigma}(\boldsymbol{\theta}))$ is described in Sect. 3.3. Evaluating this function is faster and we can put it in an optimization routine to optimize over $\boldsymbol{\theta}$ to get the estimates of $\boldsymbol{\theta}$ and $\widehat{\mathbf{x}}(\boldsymbol{\theta})$.

We use the MATLAB optimization routine `fminunc` with log-transformed range and variance parameters to avoid the nonnegativity constrains. The optimization algorithm we use is a trust-region algorithm, which requires derivative information such as gradients and Hessians. The true gradient functions involve terms with $\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}$ and therefore needs to be approximated. These problems are averted by using the genHyBR solution of $\mathbf{x}_k^*(\boldsymbol{\theta})$ in place of $\mathbf{x}$ as the matrix inversion problem reduces to a matrix vector multiplication problem. The derivative of the log determinant is also approximated by using the BTTB structure. The details are given in Appendix A. To approximate the Hessian, we use a rank-one estimate of Hessian computed as the outer product of the approximate gradient. The rationale behind this approximation is the fact that, in expectation, the outer product of the score function equals the information matrix. Once again, the details are given in Appendix A.

3.5 Uncertainty quantification

Besides a point estimate for $\mathbf{x}$, we also want to quantify the uncertainty associated with the estimated $\mathbf{x}$ and the predicted $\mathbf{y}$. We employ a parametric bootstrap for uncertainty quantification. Using the estimated $\hat{\boldsymbol{\theta}}$, we generate B samples of $\mathbf{x}_1, \ldots, \mathbf{x}_B$ from a zero-mean Gaussian process. For each $\mathbf{x}_b$, we generate $\mathbf{y}_b$ from the model in Eq. (1) with τ^2 and $\boldsymbol{\beta}$ replaced by their estimates. We then estimate $\hat{\mathbf{y}}_b$ by Kryging, but assuming $\boldsymbol{\theta}$ is known.

On the set of prediction locations $\mathbf{s}_1^*, \ldots, \mathbf{s}_m^*$, we compute the bootstrap MSE for each location $\mathbf{s}_i^*$ as

$$\begin{aligned} \text{var}\,(x(\mathbf{s}_i^*)|\hat{\boldsymbol{\theta}}) &\approx \frac{1}{B}\sum_{b=1}^{B}\left(x_b(\mathbf{s}_i^*|\hat{\boldsymbol{\theta}}) - \hat{x}_b(\mathbf{s}_i^*|\hat{\boldsymbol{\theta}})\right)^2, \\ \text{var}\,(y(\mathbf{s}_i^*)|\hat{\boldsymbol{\theta}}) &\approx \frac{1}{B}\sum_{b=1}^{B}\left(y_b(\mathbf{s}_i^*|\hat{\boldsymbol{\theta}}) - \hat{y}_b(\mathbf{s}_i^*|\hat{\boldsymbol{\theta}})\right)^2. \end{aligned} \tag{14}$$

This serves as an estimate of the classical Kriging variance for spatial prediction (Den Hertog et al. 2006). Since we use a parametric bootstrap approach, we use $B = 20$ bootstrap samples as just this many bootstrap samples provide satisfactory performance. The entire scenario entails using genHyBR method (Chung et al. 2018) B times and therefore costs $\mathcal{O}(n \log n + nk^2)$ flops. This procedure only approximates the uncertainty of the predictions assuming $\boldsymbol{\theta}$ is known. However, the bootstrap could be extended to give standard errors for the elements of $\hat{\boldsymbol{\theta}}$ as well as prediction variances that account for uncertainty in $\boldsymbol{\theta}$ by simply estimating $\boldsymbol{\theta}$ for each bootstrap sample.

3.6 Summary of the method

We now summarize the overall computational cost of this procedure. There are three main steps:

1. Optimizing the profiled likelihood $\widetilde{\text{pl}}(\boldsymbol{\theta})$ to obtain $\boldsymbol{\theta}^*$
2. Compute $\mathbf{x}_k^*(\boldsymbol{\theta}^*)$ and $\widehat{\mathbf{y}} = \mathbf{A}\mathbf{x}_k^*(\boldsymbol{\theta}^*)$.
3. Compute prediction variance using bootstrap sampling.

The optimization routine involves computing an approximate profile likelihood function and uses approximations based on the genGK algorithm to gradients and Hessian. Using genGK algorithm takes only $\mathcal{O}(nk \log n)$ steps for computing $\mathbf{x}_k^*(t)$ at the t-th iteration of the optimization.

Kryging has an overall computational complexity $\mathcal{O}(n \log n)$, assuming k is small, which is comparable to the best available approximation methods. Naturally, it is most useful in scenarios where the parameter settings favor a small value of k. Empirical observations from our simulation study suggest these to be scenarios where the spatial range parameter ρ has a moderate-to-large value or the partial sill parameter σ^2 is small. Also, Kryging is most efficient when the observations are on a grid or they are somewhat uniformly distributed over the space.

Caveat: Kryging depends on circulant embedding operations via the log-determinant approximation and bootstrap-based uncertainty quantification. A successful execution requires that a positive-definite embedding be found for the corresponding Gaussian process. Without this, the method may fail to produce a bootstrap sample from the Gaussian process in question and as a result fail to estimate uncertainty. This will also result in poor approximation of the log-determinant as many near-zero positive eigenvalues would be computed as near-zero negative eigenvalues and throw off the overall computation. This problem is evidently present when the spatial range parameter is high for the Gaussian process (see Graham et al. 2018). This problem with circulant embedding is well known. The problem of generating samples from a Gaussian process can be ameliorated by using different periodic embedding schemes (see Stein 2002; Gneiting et al. 2006; Guinness and Fuentes 2017). Forcefully resetting the small negative eigenvalues to zero or machine-precision value is a quick recourse for approximating the log-determinant. The different embedding schemes proposed in the literature may also be considered for this. However, none of these can solve the computational issue completely.

4 Simulation studies

In this section, we perform simulation studies to evaluate the performance of our proposed method. These studies aim to demonstrate the effectiveness of the model with varying sample size as well as under different parametric settings for both gridded and irregularly spaced data. We perform three different simulation studies toward this goal. In each of the experiments, for each case, we repeat the process on 25 replications. Throughout the studies, the observed values $\mathbf{y}$ are created by adding noise to $\mathbf{x}$, where $\mathbf{x}$ is an observation from a Gaussian process with constant mean β and exponential covariance function (i.e., Matérn covariance with $\nu = 0.5$) with sill σ^2 and spatial range ρ. We take the variance of the noise process to be τ^2.

The first study varies the number of observations n by generating data on a 100×100, 200×200, 300×300 and 400×400 grid in the unit square. The covariate matrix $\mathbf{X}$ is a single column vector of ones, and the choice of $\boldsymbol{\theta} = (\beta, \sigma^2, \tau^2, \rho)^\mathsf{T}$ is taken to be $(44.49, 3, 0.5, 0.1)$. The Kryging method is fit using the same grid of $p = n$ used to generate the data, and we compare performance for $k \in \{20, 50, 100, 200\}$. About 5% of the observed data $\mathbf{y}$ were held out and were treated as test data upon which the performance was evaluated.

The second study demonstrates the performance of the method under different parametric settings on a grid of 200×200 points. The spatial extents were kept same as in the first study. The four different parametric settings that were used for this study are as follows:

1. Small spatial range, $\boldsymbol{\theta} = (44.49, 3, 0.5, 0.05)^\mathsf{T}$.
2. Large spatial range, $\boldsymbol{\theta} = (44.49, 3, 0.5, 0.2)^\mathsf{T}$.
3. Small partial sill, $\boldsymbol{\theta} = (44.49, 1.5, 0.5, 0.1)^\mathsf{T}$.

4. Large partial sill, $\boldsymbol{\theta} = (44.49, 6, 0.5, 0.1)^{\mathsf{T}}$.

In all of these cases, about 5% of the data from randomly chosen locations on the grid were held out from the observed $\mathbf{y}$ and kept as test sample data on which to evaluate the method.

The third study deals with the issue of irregularly spaced data. We used the first parametric setting, $\boldsymbol{\theta} = (44.49, 3, 0.5, 0.1)$, and the spatial extent of the data as in the first study.

The number of observed points was 40,000 of which 5% were held out as test samples. The data were generated by drawing $\mathbf{x}$ on a 1000×1000 grid and discarding 96% of the data at random, leaving an irregularly spaced dataset of 40,000 observations. For testing the scalability with the grid size n, we used 200×200, 300×300 and 400×400 grids for $\mathbf{s}_i^*$.

The root-mean-squared error (RMSE) in predicting $\mathbf{y}$ pointwise coverage (CVG) of 95% prediction intervals for these predictions was averaged over replications, and median of computation time (MedTime) for all the replications was noted. These were used as performance metrics for each of the cases. For a competing method, we use the SPDE method available in the R package `INLA`. The SPDE method emerged from the comparison of several methods in Heaton et al. (2019) as one of the leading methods in terms of both computational speed and predictive accuracy.

Table 1 presents the RMSE and pointwise coverage values, averaged over replications, for the first simulation study and the median time for computation over the replicates for different choices of the tuning parameter k and different grid sizes. In all cases, $k = 50$ seems to be sufficient. The occasional inconsistencies in the computation times in Table 1 are due to the differences in the number of iterations taken by the optimization procedure to converge. In terms of RMSE and coverage, both the methods perform similarly, but Kryging is considerably faster and is more scalable. On the other hand, the coverage for the proposed method is slightly below the nominal level. This may be due to ignoring uncertainty in $\boldsymbol{\theta}$ when computing the prediction variances using Eq. 14. A possible fix for this is mentioned at the end of Sect. 3.5. However, the coverage is not so low as to require such a fix sacrificing its fast runtime advantage.

The results for the 200×200 grids with different true spatial covariance parameters are given in Table 2. For Settings 2 and 3, $k = 25$ works well. This is not surprising for Setting 2 because the process with large range is smooth as easier to represent with a small number of terms. Solid performance for small k in Setting 3 with lower partial sill is also expected because genHyBR (Chung et al. 2018) makes use of the partial sill to nugget ratio being moderate. As in the first simulation, going beyond $k = 50$ seems unnecessary and the prediction RMSE performance is comparable to that of the SPDE method, but with substantially faster computation. Since `INLA` is implemented in R and Kryging is implemented in MATLAB, the difference in platform makes the computing time comparisons difficult to interpret. However, the gain in computation time for Kryging is likely not the result of change in platform solely because `INLA` is highly optimized code (Martino and Rue 2009).

The results for irregularly spaced data are shown in Table 3. The performance is similar to the SPDE method for the proposed method with slight undercoverage. In essence, the performance is quite similar to the regularly gridded data scenario in the first simulation study.

We also check the performance of the proposed method in estimating the true mean and spatial covariance parameters against those obtained from SPDE. Across all settings and irrespective of whether the data were on a regular grid

Table 1 **a** represents $\text{RMSE}_{\text{Coverage}}$ for predicting $\mathbf{y}$ over different grid sizes and different choices of the tuning parameter k and the SPDE method, averaged over replications. The last column presents the maximum standard error for the given grid size across methods. **b** shows the median computation times in minutes over different grid sizes and different choices of the tuning parameter k and SPDE method. The figures in the bracket indicate standard errors

(a)

Grid Size	SPDE	Kryging				SE
		k = 20	k = 50	k = 100	k = 200	
100×100	$0.91_{0.95}$	$0.93_{0.92}$	$0.91_{0.91}$	$0.91_{0.91}$	$0.91_{0.91}$	$0.03_{0.03}$
200×200	$0.83_{0.95}$	$0.86_{0.92}$	$0.84_{0.91}$	$0.83_{0.91}$	$0.83_{0.91}$	$0.01_{0.02}$
300×300	$0.80_{0.95}$	$0.86_{0.92}$	$0.84_{0.91}$	$0.83_{0.91}$	$0.83_{0.91}$	$0.01_{0.02}$
400×400	$0.78_{0.95}$	$0.84_{0.91}$	$0.80_{0.89}$	$0.79_{0.88}$	$0.78_{0.88}$	$0.01_{0.02}$

(b)

Grid Size	SPDE	Kryging			
		k = 20	k = 50	k = 100	k = 200
100×100	5.42 (0.66)	0.14 (0.00)	1.84 (0.49)	5.14 (0.03)	11.00 (0.15)
200×200	44.64 (10.57)	5.51 (0.02)	1.66 (0.28)	2.12 (0.09)	48.24 (0.54)
300×300	170.01 (21.46)	3.39 (0.01)	4.11 (0.03)	5.49 (0.20)	7.81 (0.24)
400×400	662.95 (108.14)	10.78 (0.03)	12.02 (0.12)	14.28 (0.17)	18.09 (0.22)

Table 2 **a** represents $\text{RMSE}_{\text{Coverage}}$ for predicting **y** under different parametric settings for the SPDE and the proposed method with different choices of the tuning parameter k, averaged over replications. The last column presents the maximum standard error for the given setting across methods. **b** shows median computation times in minutes over different choices of the tuning parameter k and SPDE method for different parametric settings. The figures in the bracket indicate standard errors

(a)

Setting	SPDE	Kryging				SE
		k = 20	k = 50	k = 100	k = 200	
Setting 1	$0.91_{0.95}$	$0.98_{0.89}$	$0.92_{0.88}$	$0.91_{0.88}$	$0.91_{0.88}$	$0.01_{0.01}$
Setting 2	$0.78_{0.95}$	$0.80_{0.91}$	$0.80_{0.89}$	$0.79_{0.89}$	$0.79_{0.89}$	$0.01_{0.03}$
Setting 3	$0.80_{0.95}$	$0.80_{0.86}$	$0.79_{0.83}$	$0.79_{0.83}$	$0.79_{0.83}$	$0.08_{0.03}$
Setting 4	$0.90_{0.95}$	$0.98_{0.96}$	$0.92_{0.96}$	$0.91_{0.96}$	$0.90_{0.96}$	$0.02_{0.01}$

(b)

Setting	SPDE	Kryging			
		k = 20	k = 50	k = 100	k = 200
Setting 1	17.69 (2.63)	0.90 (0.00)	1.40 (0.08)	2.15 (0.10)	48.25 (0.47)
Setting 2	19.03 (1.06)	5.56 (0.05)	1.60 (0.25)	2.12 (0.07)	48.74 (0.88)
Setting 3	18.26 (3.10)	5.47 (0.41)	1.37 (0.08)	2.17 (0.08)	48.53 (0.27)
Setting 4	18.49 (2.19)	5.57 (0.04)	2.37 (1.02)	2.16 (0.08)	48.62 (0.52)

Table 3 **a** represents $\text{RMSE}_{\text{Coverage}}$ for predicting **y** for the SPDE and Kryging with different choices of the tuning parameter k and different underlying grid sizes, averaged over replications for irregularly spaced datasets. The last column presents the maximum standard error for the given setting across methods. **b** shows median computation times in minutes over different grid sizes and different choices of the tuning parameter k and SPDE method. The figures in the bracket indicate standard errors

(a)

Grid size	SPDE	Kryging			SE
		k = 20	k = 50	k = 100	
200 × 200	$0.82_{0.95}$	$0.85_{0.90}$	$0.83_{0.88}$	$0.83_{0.87}$	$0.02_{0.02}$
300 × 300	$0.82_{0.95}$	$0.85_{0.90}$	$0.83_{0.89}$	$0.83_{0.88}$	$0.02_{0.02}$
400 × 400	$0.82_{0.95}$	$0.85_{0.91}$	$0.83_{0.89}$	$0.82_{0.89}$	$0.02_{0.02}$

(b)

Grid size	SPDE	Kryging		
		k = 20	k = 50	k = 100
200 × 200	33.55 (3.58)	8.08 (0.02)	3.33 (0.31)	3.93 (0.15)
300 × 300	33.55 (3.58)	4.86 (0.02)	5.58 (0.03)	6.85 (0.05)
400 × 400	33.55 (3.58)	8.58 (0.10)	9.96 (0.11)	12.40 (0.09)

or not, the results are consistent. While SPDE does a better job at estimating the nugget parameter, Kryging does a better job at estimating the partial sill. For estimating range and the mean parameters, both the method perform similarly. Detailed comparisons are presented in tables in Appendix B.

5 Application to MODIS/Terra land surface temperature data

In this section, we analyze a real dataset using the proposed method. We use the dataset used by Heaton et al. (2019) for a comparison of methods for analyzing massive spatial data. The dataset consists of Level-3 data on land surface temperatures as measured by the Terra instrument onboard the MODIS satellite on August 4, 2016. The original data were available in MODIS reprojection tool web (MRTweb), which has since been decommissioned. The entire dataset is available in the GitHub repository for the Heaton et al. (2019) project at this GitHub repository. The main reason for using this dataset is so that we can compare to other existing methods easily as this dataset was previously analyzed by twelve other existing methods in Heaton et al. (2019).

The observations were laid out on a regular grid of size 500×300 within longitude values -95.91153 to -91.28381 and latitude values 34.29519 to 37.06811. About 1.1% of the data, 1, 691 grid cells out of 150, 000 cells, were corrupted due to cloud cover. A further 42,740 observations were held out from the training set, keeping about 70% of the data in the training set and about 30% in the test set. The training and testing datasets along the locations are available in the previously mentioned GitHub repository. Figure 1 shows the true data (top) and training data (second from top) created after removing some observations.

We ran the Kryging algorithm with $k = 50$, 100, 200 and 300. For each value of k, we use different initial val-

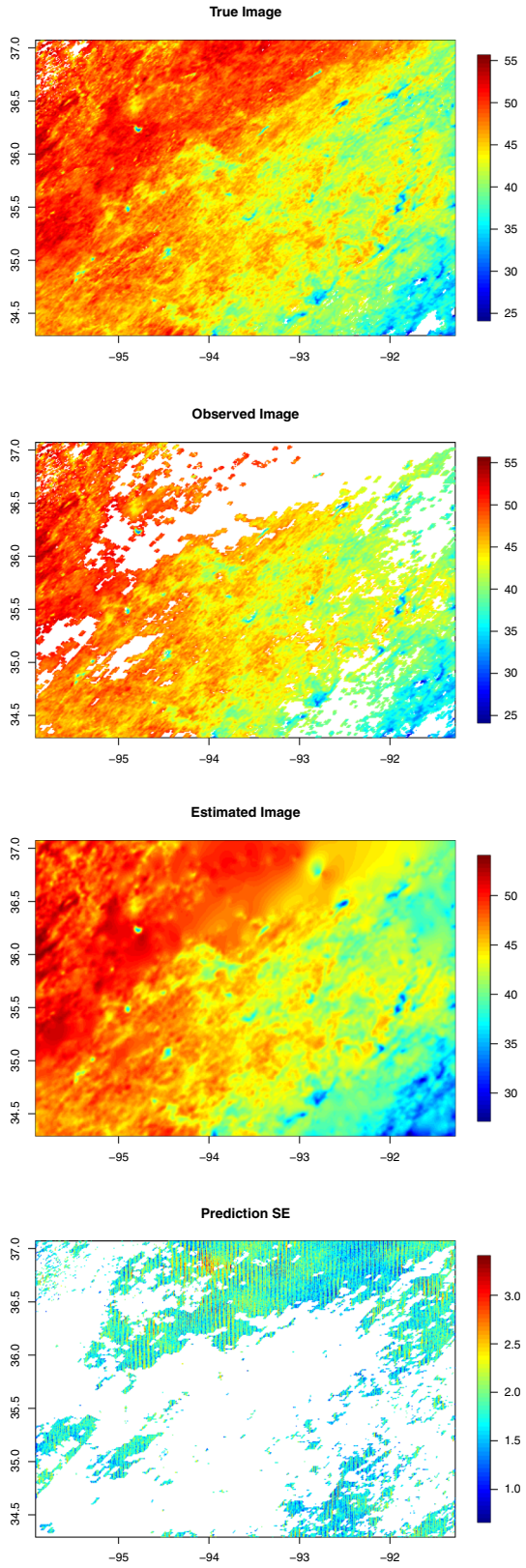

Fig. 1 True satellite image (top), the image used for training after holding out data for test sample (second from top), the image obtained from the estimated values (second from bottom) and the prediction standard errors (bottom) for $k = 200$

ues and pick the best one using fivefold cross-validation within the training dataset. The mean absolute error (MAE), root-mean-squared error (RMSE), continuously ranked probability score (CRPS), interval score (INT) and pointwise coverage (CVG) for the predictions of the test set datapoints were computed for each case, and the computation times were noted and are tabulated in Table 4. Figure 1 shows the estimates (second from bottom) and corresponding standard errors (bottom) for the data. The estimated image picks up all the spatial features in the true data, indicating a good fit. Since the same dataset was also analyzed by twelve other existing methods, the above-mentioned metrics for which are available in the Heaton et al. (2019) paper. The relevant results, as presented in the original paper, are presented in Table 5. This allows us a chance to compare the performance of our method to other existing methods, although the computing platforms were not the same for the two cases.

In terms of RMSE and coverage, SPDE (Lindgren et al. 2011), nearest neighbor Gaussian process or NNGP (Datta et al. 2016a, b, c) and LatticeKrig (Nychka et al. 2015) perform better than the proposed method. The time taken by the method is significantly less than the SPDE method and comparable to LatticeKrig. Although it should be mentioned that the they were run in different platforms with similar hardware setup, the comparison should not be considered a direct one. The time presented for the NNGP method in Heaton et al. (2019) considers only the time taken for the conjugate model where a well-defined grid of possible parameter values were supplied to the model to use cross-validation in parallel. This range of parameter values need to be determined first and is the more difficult and time-consuming part of any existing approximate inference method and neither the strategy nor the time taken to arrive at those numbers was reported in Heaton et al. (2019).

6 Conclusion

In this article, we propose an approximate inference method for analyzing massive spatial datasets using Krylov subspace approximation and profile maximum likelihood methods. The method assumes that the underlying process realizations are on a regular equispaced grid, but the observations need not be colocated on the grid. While we exclusively model the spatial process covariance using the Matérn covariance family, the method works for any choice of stationary covariance function. We also propose an approach to approximate log-determinants for symmetric BTTB matrices which has guaranteed asymptotic convergence to the true log-determinant value. The method has computational complexity of $\mathcal{O}(n \log n)$, resulting in fast run times and excellent scalability with the sample size while producing decent estimates and requires little tuning. The method is expected to

Table 4 Performance of the proposed method on the MODIS dataset for various choices of k

k	MAE	RMSE	CRPS	INT	CVG	Run time (min.)	Cores used
50	1.43	1.95	1.07	10.97	0.93	11.18	4
100	1.43	1.85	1.04	9.74	0.93	15.10	4
200	1.36	1.78	0.99	9.60	0.93	19.59	4
300	1.36	1.79	0.99	9.68	0.93	27.01	4

Table 5 Results from the case study competition for the satellite data as in Table 3 of Heaton et al. (2019)

Method	MAE	RMSE	CRPS	INT	CVG	Run time(min)	Cores used
FRK	1.96	2.44	1.44	14.08	0.70	2.32	1
Gapfill	1.33	1.86	1.17	34.78	0.36	1.39	40
LatticeKrig	1.22	1.68	0.87	7.55	0.96	27.92	1
LAGP	1.65	2.08	1.17	10.81	0.83	2.27	40
Metakriging	2.08	2.50	1.44	10.77	0.89	2888.52	30
MRA	1.33	1.85	0.94	8.00	0.92	15.61	1
NNGP	1.21	1.64	0.85	7.57	0.95	2.06	10
Partition	1.41	1.80	1.02	10.49	0.86	79.98	55
Pred. Proc.	2.15	2.64	1.55	15.51	0.83	160.24	10
SPDE	1.10	1.53	0.83	8.85	0.97	120.33	2
Tapering	1.87	2.45	1.32	10.31	0.93	133.26	1
Periodic Embedding	1.29	1.79	0.91	7.44	0.93	9.81	1

run especially well when the spatial range is small to moderate and partial sill-to-nugget ratio is moderate. This is seen in the applications involving both synthetic and real datasets.

Although uncertainties for the mean and spatial parameter estimates are not provided directly, they can be obtained using the following approaches. A reasonable approach would be to compute the exact Hessian and its inverse for the optimization process of Eq. (13). However, that is time-consuming as it has $\mathcal{O}(n^3)$ complexity involved with the computation. A suitable approximation to the inverse of the Hessian will be needed to efficiently estimate the uncertainties associated with these parameters. A computationally expensive alternative is to estimate the parameters using the parametric bootstrap, as outlined in Sect. 3.5.

The method is proposed as a d-dimensional method. However, for irregular datasets on dimensions higher than 3, the grid formation is slow and difficult. But for the purposes of geostatistical analyses, we need only concern ourselves with problems in $\mathbb{R}^2$ or $\mathbb{R}^2 \times \mathbb{R}$ where grids are simple and easy to deal with. Should the case arise where one has to deal with higher-dimensional geospatial analysis, one needs to look for a suitable alternative to the grid structure, which can be a future avenue for research. Moreover, Kryging is most attractive when the observations are approximately on a grid or uniformly distributed and adaptations for extremely irregular cases such as data observed along transects or in separated clusters is another area of future work.

The proposed model can be utilized in many other scenarios than simply what has been illustrated in this article. The computational amenities of the method can be utilized for spatiotemporal modeling. Changing the observational model to include two or more sources of data can be contemplated as well. Quantifying uncertainties for the mean and the spatial parameters can be one possible extension. Extending the method to non-Gaussian observational models, for example, binary or count data, would be another possibility.

Acknowledgements The authors were partially supported by the National Science Foundation through the awards DMS-1845406 and DMS-1638521. The authors were also partially supported by the National Institute of Health through the awards R01ES031651-01 and R01ES027892 and by The King Abdullah University of Science and Technology grant 3800.2. We would like to thank them for their support.

Funding The authors were partially supported by the National Science Foundation through the awards DMS-1845406 and DMS-1638521. The authors were also partially supported by the National Institute of Health through the awards R01ES031651-01 and R01ES027892 and by The King Abdullah University of Science and Technology grant 3800.2.

Availability of Data and Material The dataset analyzed in Sect. 5 is available in the GitHub repository for the Heaton et al. (2019) project at this GitHub repository.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Code availability A GitHub repository has been set up that contains codes and a demonstration file for the methods described in the article.

A Gradient and Hessian computation for the optimization procedure

In this section, we present the necessary details of computing and approximating the gradient and Hessian for the optimization routine.

We first derive exact expressions for the gradient and then show how to approximate them using the strategy in Sects. 3.2 and 3.1. Computing the analytical gradient would require computing derivatives of $\mathbf{\Gamma} = \left(\frac{1}{\sigma^2}\mathbf{\Sigma}(\boldsymbol{\theta})^{-1} + \frac{1}{\tau^2}\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}$ and $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ with respect to each of μ, σ^2, τ^2 and ρ. For convenience, we reparameterize $1/\sigma^2 = \lambda^2$ and $1/\tau^2 = \lambda_e^2$. Using the precision instead of variance brings about greater ease in computing the analytical derivatives. Under the new parameterization,

$$\mathbf{\Gamma} = \left(\lambda_e^2\mathbf{A}^\mathsf{T}\mathbf{A} + \lambda^2\mathbf{\Sigma}^{-1}\right)^{-1}, \tag{15}$$

$$\widehat{\mathbf{x}}(\boldsymbol{\theta}) = \mathbf{\Gamma}\lambda_e^2\mathbf{A}^\mathsf{T}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \tag{16}$$

and

$$\begin{aligned}\text{pl}(\boldsymbol{\theta}) \simeq\ & \frac{p}{2}\log\lambda_e^2 - \frac{\lambda_e^2}{2}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})^\mathsf{T}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}) \\ & + \frac{n}{2}\log\lambda^2 - \frac{1}{2}\log\det\ \mathbf{\Sigma}(\boldsymbol{\theta}) - \\ & \frac{\lambda^2}{2}\hat{\mathbf{x}}(\boldsymbol{\theta})^\mathsf{T}\mathbf{\Sigma}(\boldsymbol{\theta})^{-1}\hat{\mathbf{x}}(\boldsymbol{\theta}),\end{aligned} \tag{17}$$

where $\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}) = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{A}\hat{\mathbf{x}}(\boldsymbol{\theta})$.

The derivatives for $\mathbf{\Gamma}$ are computed to be

$$\begin{aligned}\frac{\partial\mathbf{\Gamma}}{\partial\boldsymbol{\beta}} &= \mathbf{0}, \\ \frac{\partial\mathbf{\Gamma}}{\partial\lambda^2} &= -\mathbf{\Gamma}\mathbf{\Sigma}(\rho)^{-1}\mathbf{\Gamma}, \\ \frac{\partial\mathbf{\Gamma}}{\partial\rho} &= \lambda^2\mathbf{\Gamma}\mathbf{\Sigma}(\rho)^{-1}\left(\text{d}\mathbf{\Sigma}(\rho)\right)\mathbf{\Sigma}(\rho)^{-1}\mathbf{\Gamma}, \\ \frac{\partial\mathbf{\Gamma}}{\partial\lambda_e^2} &= -\mathbf{\Gamma}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{\Gamma},\end{aligned} \tag{18}$$

where $\text{d}\mathbf{\Sigma}(\rho)$ denotes the derivative of $\mathbf{\Sigma}(\rho)$ with respect to ρ. This is easy to compute analytically and has the nice BTTB property that $\mathbf{\Sigma}(\rho)$ has.

Using the expressions in (18), we compute the derivatives of $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ to be

$$\begin{aligned}\frac{\partial\widehat{\mathbf{x}}(\boldsymbol{\theta})}{\partial\boldsymbol{\beta}} &= -\lambda_e^2\mathbf{\Gamma}\mathbf{A}^\mathsf{T}\mathbf{X}, \\ \frac{\partial\widehat{\mathbf{x}}(\boldsymbol{\theta})}{\partial\lambda^2} &= -\mathbf{\Gamma}\mathbf{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta}), \\ \frac{\partial\widehat{\mathbf{x}}(\boldsymbol{\theta})}{\partial\rho} &= \lambda^2\mathbf{\Gamma}\mathbf{\Sigma}(\rho)^{-1}\left(\text{d}\mathbf{\Sigma}(\rho)\right)\mathbf{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta}), \\ \frac{\partial\widehat{\mathbf{x}}(\boldsymbol{\theta})}{\partial\lambda_e^2} &= \mathbf{\Gamma}\mathbf{A}^\mathsf{T}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}).\end{aligned} \tag{19}$$

Substituting the expressions for analytical derivatives of $\mathbf{\Gamma}$ and $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ in the expression for the analytical gradient, we have it computed to be

$$\begin{aligned}\frac{\partial\text{pl}}{\partial\boldsymbol{\beta}} &= \lambda_e^2\mathbf{X}^\mathsf{T}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}), \\ \frac{\partial\text{pl}}{\partial\lambda^2} &= \frac{n}{2\lambda^2} - \frac{1}{2}\widehat{\mathbf{x}}(\boldsymbol{\theta})^\mathsf{T}\mathbf{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta}) \\ \frac{\partial\text{pl}}{\partial\rho} &= -\frac{1}{2}\text{dL} + \frac{1}{2}\lambda^2\widehat{\mathbf{x}}(\boldsymbol{\theta})^\mathsf{T}\mathbf{\Sigma}(\rho)^{-1}\left(\text{d}\mathbf{\Sigma}(\rho)\right)\mathbf{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta}), \\ \frac{\partial\text{pl}}{\partial\lambda_e^2} &= \frac{p}{2\lambda_e^2} - \frac{1}{2}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})^\mathsf{T}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}),\end{aligned} \tag{20}$$

where dL is the derivative of $\log\det\ \mathbf{\Sigma}(\rho)$ with respect to ρ.

We approximate the gradient expressions in (20) by approximating $\widehat{\mathbf{x}}(\boldsymbol{\theta})$ by $\mathbf{x}_k^*(\boldsymbol{\theta})$ as in (12) and using the exact arithmetic identities expressed in (10). The approximated gradients can be computed as

$$\begin{aligned}\frac{\partial\text{pl}}{\partial\boldsymbol{\beta}} &\approx \lambda_e^2\mathbf{X}^\mathsf{T}\boldsymbol{\psi}_k^*(\boldsymbol{\theta}), \\ \frac{\partial\text{pl}}{\partial\lambda^2} &\approx \frac{n}{2\lambda^2} - \frac{1}{2}\|\mathbf{z}_k\|_2^2, \\ \frac{\partial\text{pl}}{\partial\rho} &\approx -\frac{1}{2}\widehat{\text{dL}} + \frac{\lambda^2}{2}\mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\text{d}\mathbf{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{z}_k, \\ \frac{\partial\text{pl}}{\partial\lambda_e^2} &\approx \frac{p}{2\lambda_e^2} - \frac{1}{2}\boldsymbol{\psi}_k^*(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{\psi}_k^*(\boldsymbol{\theta}),\end{aligned} \tag{21}$$

where $\boldsymbol{\psi}_k^*(\boldsymbol{\theta}) = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{A}\mathbf{x}_k^*(\boldsymbol{\theta})$ and $\mathbf{V}_k, \mathbf{z}_k$ have been defined in Sect. 3.2.

$\widehat{\text{dL}}$ is an approximation to dL, the derivative of the log-determinant of $\mathbf{\Sigma}(\rho)$ with respect to ρ. The analytical expression for dL turns out to be

$$\text{dL} = \text{trace}\left(\mathbf{\Sigma}(\rho)^{-1}\text{d}\mathbf{\Sigma}(\rho).\right)$$

This is infeasible to compute directly and is therefore approximated using the BTTB structure of $\mathbf{\Sigma}(\rho)$ and $\text{d}\mathbf{\Sigma}(\rho)$.

Any symmetric matrix with BTTB structure can be extended to have a BCCB structure as was done in computing the log-determinant itself and one can extract the eigenvalues of the matrix with BTTB structure using the matrix with BCCB structure. Any BCCB matrix is diagonalizable as $\mathbf{FDF}^\mathsf{T}$, where $\mathbf{F}$ is a scaled matrix consisting of

d-dimensional (d=2, in our case) Fourier coefficients, irrespective of the BCCB matrix being diagonalized. Therefore, we can say

$$\begin{aligned}\boldsymbol{\Sigma}(\rho) &= \mathbf{F}\mathbf{D}_1\mathbf{F}^{\mathsf{T}},\\ \boldsymbol{\Sigma}(\rho)^{-1} &= \mathbf{F}\mathbf{D}_1^{-1}\mathbf{F}^{\mathsf{T}},\\ \mathrm{d}\boldsymbol{\Sigma}(\rho) &= \mathbf{F}\mathbf{D}_2\mathbf{F}^{\mathsf{T}}.\end{aligned} \tag{22}$$

These imply that

$$\begin{aligned}\operatorname{trace}\left(\boldsymbol{\Sigma}(\rho)^{-1}\mathrm{d}\boldsymbol{\Sigma}(\rho)\right) &= \operatorname{trace}\left(\mathbf{F}\mathbf{D}_1^{-1}\mathbf{F}^{\mathsf{T}}\mathbf{F}\mathbf{D}_2\mathbf{F}^{\mathsf{T}}\right)\\ &= \operatorname{trace}\left(\mathbf{D}_1^{-1}\mathbf{D}_2\right).\end{aligned} \tag{23}$$

Since both $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal, approximating dL boils down to computing $\mathbf{D}_1$ and $\mathbf{D}_2$, which can be computed by d-dimensional FFT of the corresponding first circulant block structures of the extended BCCB structure and subsetting it properly. The equivalence in computing the derivative of log-determinant of the BTTB and matrix and its corresponding BCCB matrix has been demonstrated by Kent and Mardia (1996), showing the approximation to have the same error rate as in approximating the log-determinant itself. Approximating the derivative of the log-determinant term also costs the same as approximating the log-determinant itself, $\mathcal{O}(n \log n)$.

While minimizing the negative log-likelihood function, the Hessian turns out to be simply the Information matrix $\mathbf{I}(\boldsymbol{\theta})$. While

$$\mathbb{E}\left(-\nabla_2 \mathrm{pl}(\boldsymbol{\theta})\right) = \mathbf{I}(\boldsymbol{\theta}),$$

we also have

$$\mathbb{E}\left(\nabla \mathrm{pl}(\boldsymbol{\theta})\nabla \mathrm{pl}(\boldsymbol{\theta})^{\mathsf{T}}\right) = \mathbb{E}\left[\left(-\nabla \mathrm{pl}(\boldsymbol{\theta})\right)\left(-\nabla \mathrm{pl}(\boldsymbol{\theta})\right)^{\mathsf{T}}\right].$$

Here the expectations are computed with respect to $\mathbf{y}$ and ∇, ∇_2 represent the gradient and Hessian created by computing first- and second-order partial derivatives with respect to $\boldsymbol{\theta}$. Therefore, the outer product of the gradient with itself serves as a rank-one estimate for the Hessian for a likelihood optimization problem. Although we are using profile likelihood instead of the actual likelihood function, the approximation still stands in an asymptotic sense since both the actual likelihood estimator and the profile likelihood estimators have the same asymptotic properties. This prompts us to take the outer product of the approximated gradient with itself as a rank-one approximation to the Hessian.

However, we compute the unique entries of the exact Hessian to be

$$\begin{aligned}\frac{\partial^2 \mathrm{pl}}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathsf{T}}} &= -\lambda_e^2\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda_e^4\mathbf{X}^{\mathsf{T}}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathsf{T}}\mathbf{X}\\ \frac{\partial^2 \mathrm{pl}}{\partial\boldsymbol{\beta}\partial\lambda^2} &= \lambda_e^2\mathbf{X}^{\mathsf{T}}\mathbf{A}\boldsymbol{\Gamma}\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\boldsymbol{\beta}\partial\rho} &= -\lambda_e^2\lambda^2\mathbf{X}^{\mathsf{T}}\mathbf{A}\boldsymbol{\Gamma}\boldsymbol{\Sigma}(\rho)^{-1}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\boldsymbol{\beta}\partial\lambda_e^2} &= \mathbf{X}^{\mathsf{T}}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}) - \lambda_e^2\mathbf{X}^{\mathsf{T}}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathsf{T}}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\lambda^4} &= -\frac{n}{2\lambda^4} + \widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\rho)^{-1}\boldsymbol{\Gamma}\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\theta)\\ \frac{\partial^2 \mathrm{pl}}{\partial\lambda^2\partial\rho} &= \frac{1}{2}\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\rho)^{-1}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ &\quad - \lambda^2\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\rho)^{-1}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\\ &\quad \times \boldsymbol{\Sigma}(\rho)^{-1}\boldsymbol{\Gamma}\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\lambda^2\partial\lambda_e^2} &= -\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\rho)^{-1}\boldsymbol{\Gamma}\mathbf{A}^{\mathsf{T}}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\rho^2} &= -\frac{1}{2}\mathrm{d}^2\mathrm{L} + \frac{\lambda^2}{2}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ &\quad \times \boldsymbol{\Sigma}(\rho)^{-1}\left(\mathrm{d}^2\boldsymbol{\Sigma}(\rho)\right)\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ &\quad - \lambda^2\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\rho)^{-1}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\Big[\boldsymbol{\Sigma}(\rho)^{-1}\\ &\quad -\boldsymbol{\Sigma}(\rho)^{-1}\boldsymbol{\Gamma}\boldsymbol{\Sigma}(\rho)^{-1}\Big]\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\boldsymbol{\Sigma}(\rho)^{-1}\widehat{\mathbf{x}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\rho\partial\lambda_e^2} &= \lambda^2\widehat{\mathbf{x}}(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Sigma}(\rho)^{-1}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\boldsymbol{\Sigma}(\rho)^{-1}\boldsymbol{\Gamma}\mathbf{A}^{\mathsf{T}}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})\\ \frac{\partial^2 \mathrm{pl}}{\partial\lambda_e^4} &= -\frac{p}{\lambda_e^4} + \widehat{\boldsymbol{\psi}}(\boldsymbol{\theta})^{\mathsf{T}}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathsf{T}}\widehat{\boldsymbol{\psi}}(\boldsymbol{\theta}),\end{aligned} \tag{24}$$

where $\mathrm{d}^2\mathrm{L}$ represents the second derivative of $\log\det\boldsymbol{\Sigma}(\rho)$ with respect to ρ and $\mathrm{d}^2\boldsymbol{\Sigma}(\rho)$ is the second derivative of $\boldsymbol{\Sigma}(\rho)$ with respect to ρ. $\mathrm{d}^2\boldsymbol{\Sigma}(\rho)$ also has a BTTB structure as $\boldsymbol{\Sigma}(\rho)$ and $\mathrm{d}\boldsymbol{\Sigma}(\rho)$.

These entries are then approximated using the approximation to $\boldsymbol{\Gamma}$ as presented in Chung et al. (2018), namely

$$\boldsymbol{\Gamma} \approx \lambda^{-2}\left(\boldsymbol{\Sigma}(\rho) - \mathbf{Z}_k\boldsymbol{\Delta}_k\mathbf{Z}_k^{\mathsf{T}}\right), \tag{25}$$

where $\mathbf{Z}_k = \boldsymbol{\Sigma}(\rho)\mathbf{V}_k\mathbf{W}_k$ with $\mathbf{B}_k^{\mathsf{T}}\mathbf{B}_k = \mathbf{W}_k\boldsymbol{\Theta}_k\mathbf{W}_k$ and $\boldsymbol{\Delta}_k = \left(\mathbf{I} + \lambda^{-2}\boldsymbol{\Theta}_k\right)^{-1}$.

We define $\mathbf{z}_0 = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{A}\mathbf{x}_k^*(\boldsymbol{\theta}) = \boldsymbol{\psi}_k^*(\boldsymbol{\theta})$. The approximated entries of the Hessian are

$$\begin{aligned}\frac{\partial^2 \mathrm{pl}}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathsf{T}}} &\approx -\lambda_e^2\mathbf{X}^{\mathsf{T}}\mathbf{X} + \frac{\lambda_e^4}{\lambda^2}\mathbf{X}^{\mathsf{T}}\mathbf{A}\boldsymbol{\Sigma}(\rho)\mathbf{A}^{\mathsf{T}}\mathbf{X}\\ &\quad -\frac{\lambda_e^4}{\lambda^2}\mathbf{X}^{\mathsf{T}}\mathbf{U}_k\mathbf{B}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^{\mathsf{T}}\mathbf{B}_k^{\mathsf{T}}\mathbf{U}_k^{\mathsf{T}}\mathbf{X}\end{aligned}$$

$$
\begin{aligned}
\frac{\partial^2 \mathrm{pl}}{\partial \boldsymbol{\beta} \partial \lambda^2} &\approx \frac{\lambda_e^2}{\lambda^2}\mathbf{X}^\mathsf{T}\mathbf{A}\widehat{x^*}(\boldsymbol{\theta}) - \frac{\lambda_e^2}{\lambda^2}\mathbf{X}^\mathsf{T}\mathbf{U}_k\mathbf{B}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{z}_k \\
\frac{\partial^2 \mathrm{pl}}{\partial \boldsymbol{\beta} \partial \rho} &\approx -\lambda_e^2\mathbf{X}^\mathsf{T}\mathbf{A}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{z}_k \\
&\quad +\lambda_e^2\mathbf{X}^\mathsf{T}\mathbf{U}_k\mathbf{B}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{z}_k \\
\frac{\partial^2 \mathrm{pl}}{\partial \boldsymbol{\beta} \partial \lambda_e^2} &\approx -\mathbf{X}^\mathsf{T}\mathbf{z}_0 - \frac{\lambda_e^2}{\lambda^2}\mathbf{X}^\mathsf{T}\mathbf{A}\boldsymbol{\Sigma}(\rho)\mathbf{A}^\mathsf{T}\mathbf{z}_0 \\
&\quad +\frac{\lambda_e^2}{\lambda^2}\mathbf{X}^\mathsf{T}\mathbf{U}_k\mathbf{B}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{B}_k^\mathsf{T}\mathbf{U}_k^\mathsf{T}\mathbf{z}_0 \\
\frac{\partial^2 \mathrm{pl}}{\partial \lambda^4} &\approx -\frac{n}{2\lambda^4} + \frac{1}{\lambda^2}\|\mathbf{z}_k\|_2^2 - \frac{1}{\lambda^2}\mathbf{z}_k^\mathsf{T}\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{z}_k \\
\frac{\partial^2 \mathrm{pl}}{\partial \lambda^2 \partial \rho} &\approx -\frac{1}{2}\mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{z}_k \\
&\quad +\mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{z}_k \\
\frac{\partial^2 \mathrm{pl}}{\partial \lambda^2 \partial \lambda_e^2} &\approx -\frac{1}{\lambda^2}\mathbf{z}_k^\mathsf{T}\mathbf{B}_k^\mathsf{T}\mathbf{U}_k^\mathsf{T}\mathbf{z}_0 + \frac{1}{\lambda^2}\mathbf{z}_k^\mathsf{T}\mathbf{W}_k^\mathsf{T}\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{B}_k^\mathsf{T}\mathbf{U}_k^\mathsf{T}\mathbf{z}_0 \\
\frac{\partial^2 \mathrm{pl}}{\partial \rho^2} &\approx -\frac{1}{2}\widehat{\mathrm{d}^2\mathrm{L}} + \frac{\lambda^2}{2}\mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}^2\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{z}_k \\
&\quad -\lambda^2\mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{W}_k\boldsymbol{\Delta}_k \\
&\quad \times\mathbf{W}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{z}_k \\
\frac{\partial^2 \mathrm{pl}}{\partial \rho \partial \lambda_e^2} &\approx \mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{A}^\mathsf{T}\mathbf{z}_0 \\
&\quad -\mathbf{z}_k^\mathsf{T}\mathbf{V}_k^\mathsf{T}\left(\mathrm{d}\boldsymbol{\Sigma}(\rho)\right)\mathbf{V}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{B}_k^\mathsf{T}\mathbf{U}_k^\mathsf{T}\mathbf{z}_0 \\
\frac{\partial^2 \mathrm{pl}}{\partial \lambda_e^4} &\approx -\frac{p}{2\lambda_e^4} + \frac{1}{\lambda^2}\mathbf{z}_0^\mathsf{T}\mathbf{A}\boldsymbol{\Sigma}(\rho)\mathbf{A}^\mathsf{T}\mathbf{z}_0 \\
&\quad -\frac{1}{\lambda^2}\mathbf{z}_0^\mathsf{T}\mathbf{U}_k\mathbf{B}_k\mathbf{W}_k\boldsymbol{\Delta}_k\mathbf{W}_k^\mathsf{T}\mathbf{B}_k^\mathsf{T}\mathbf{U}_k^\mathsf{T}\mathbf{z}_0, \qquad (26)
\end{aligned}
$$

where $\widehat{\mathrm{d}^2\mathrm{L}}$ is a numerical approximation to $\mathrm{d}^2\mathrm{L}$. We do not use this approximation for our computing, but hope to use it in future.

B Additional tables from the simulation study

In this section, we provide additional results for the simulation study. Table 6 evaluates parameter estimations for the first simulation study for both SPDE and Kryging methods. The same is done in Tables 7 and 8 for the second and third simulation studies. The results across the board are similar as mentioned in Section 4. SPDE performs better in estimating the nugget parameter τ^2, while Kryging performs better in estimating the partial sill parameter σ^2. Both methods do equally well in estimating the mean parameter β and the spatial range parameter ρ.

Table 6 RMSE in estimating the parameters for SPDE and Kryging for different grid sizes and choices of k as in the first simulation study. The true values for the parameters were (44.49, 3, 0.5, 1). The figures in brackets indicate standard error

Parameter	Grid Size	SPDE	Kryging			
			k = 20	k = 50	k = 100	k = 200
β	100 × 100	0.30 (0.30)	0.31 (0.32)	0.30 (0.32)	0.31 (0.32)	0.31 (0.32)
	200 × 200	0.23 (0.22)	0.28 (0.23)	0.28 (0.23)	0.28 (0.23)	0.28 (0.23)
	300 × 300	0.32 (0.24)	0.32 (0.25)	0.32 (0.25)	0.32 (0.25)	0.32 (0.25)
	400 × 400	0.26 (0.26)	0.29 (0.26)	0.29 (0.26)	0.29 (0.26)	0.29 (0.26)
σ^2	100 × 100	1.43 (0.11)	0.36 (0.34)	0.36 (0.34)	0.36 (0.34)	0.36 (0.34)
	200 × 200	1.59 (0.07)	0.31 (0.24)	0.31 (0.24)	0.31 (0.24)	0.31 (0.24)
	300 × 300	1.70 (0.07)	0.33 (0.26)	0.33 (0.26)	0.33 (0.26)	0.33 (0.26)
	400 × 400	1.81 (0.06)	0.30 (0.17)	0.30 (0.17)	0.30 (0.17)	0.30 (0.17)
τ^2	100 × 100	0.10 (0.01)	0.17 (0.05)	0.17 (0.05)	0.17 (0.05)	0.17 (0.05)
	200 × 200	0.06 (0.01)	0.16 (0.04)	0.16 (0.04)	0.16 (0.04)	0.16 (0.04)
	300 × 300	0.05 (0.00)	0.16 (0.05)	0.16 (0.05)	0.16 (0.05)	0.16 (0.05)
	400 × 400	0.04 (0.00)	0.19 (0.03)	0.19 (0.03)	0.19 (0.03)	0.19 (0.03)
ρ	100 × 100	0.06 (0.02)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.00)
	200 × 200	0.01 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)
	300 × 300	0.01 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)
	400 × 400	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)

Table 7 RMSE in estimating the parameters for SPDE and Kryging under different parametric settings and different choices of k as in the second simulation study. The true values for the parameters were (44.49, 3, 0.5, 0.05), (44.49, 3, 0.5, 0.2), (44.49, 1.5, 0.5, 0.1) and (44.49, 6, 0.5, 0.1) for settings 1 through 4, respectively. The figures in brackets indicate standard error

Parameter	Setting	SPDE	Kryging			
			k=20	k=50	k=100	k=200
β	Setting 1	0.16 (0.15)	0.16 (0.14)	0.16 (0.14)	0.16 (0.14)	0.16 (0.14)
	Setting 2	0.42 (0.37)	0.46 (0.39)	0.46 (0.39)	0.46 (0.39)	0.46 (0.39)
	Setting 3	0.18 (0.10)	0.22 (0.10)	0.22 (0.10)	0.22 (0.10)	0.22 (0.10)
	Setting 4	0.42 (0.27)	0.43 (0.28)	0.43 (0.28)	0.43 (0.28)	0.43 (0.28)
σ^2	Setting 1	1.43 (0.05)	0.16 (0.17)	0.16 (0.17)	0.16 (0.17)	0.16 (0.17)
	Setting 2	1.79 (0.11)	0.59 (0.32)	0.59 (0.32)	0.59 (0.32)	0.59 (0.32)
	Setting 3	0.47 (0.06)	0.23 (0.21)	0.23 (0.21)	0.23 (0.21)	0.23 (0.21)
	Setting 4	4.07 (0.06)	0.70 (0.44)	0.70 (0.44)	0.70 (0.44)	0.70 (0.44)
τ^2	Setting 1	0.10 (0.01)	0.16 (0.02)	0.16 (0.02)	0.16 (0.02)	0.16 (0.02)
	Setting 2	0.04 (0.01)	0.21 (0.06)	0.21 (0.06)	0.21 (0.06)	0.21 (0.06)
	Setting 3	0.07 (0.15)	0.30 (0.03)	0.30 (0.03)	0.30 (0.03)	0.30 (0.03)
	Setting 4	0.12 (0.01)	0.10 (0.06)	0.10 (0.06)	0.10 (0.06)	0.10 (0.06)
ρ	Setting 1	0.03 (0.00)	0.02 (0.00)	0.02 (0.00)	0.02 (0.00)	0.02 (0.00)
	Setting 2	0.05 (0.02)	0.13 (0.00)	0.13 (0.00)	0.13 (0.00)	0.13 (0.00)
	Setting 3	0.03 (0.02)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)
	Setting 4	0.01 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)

Table 8 RMSE in estimating the parameters for SPDE and Kryging for different choices of underlying grid size and k for the simulation study with irregularly spaced data. The true parameter values were (44.49, 3, 0.5, 0.1). The figures in brackets indicate standard error

Parameter	Grid size	SPDE	Kryging		
			k=20	k=50	k=100
β	200×200	0.24 (0.18)	0.29 (0.16)	0.29 (0.16)	0.29 (0.16)
	300×300	0.24 (0.18)	0.29 (0.16)	0.29 (0.16)	0.29 (0.16)
	400×400	0.24 (0.18)	0.29 (0.16)	0.29 (0.16)	0.29 (0.16)
σ^2	200×200	1.61 (0.07)	0.26 (0.21)	0.26 (0.21)	0.26 (0.21)
	300×300	1.61 (0.07)	0.26 (0.21)	0.26 (0.21)	0.26 (0.21)
	400×400	1.61 (0.07)	0.26 (0.21)	0.26 (0.21)	0.26 (0.21)
τ^2	200×200	0.06 (0.01)	0.17 (0.04)	0.17 (0.04)	0.17 (0.04)
	300×300	0.06 (0.01)	0.17 (0.04)	0.17 (0.04)	0.17 (0.04)
	400×400	0.06 (0.01)	0.17 (0.04)	0.17 (0.04)	0.17 (0.04)
ρ	200×200	0.01 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)
	300×300	0.01 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)
	400×400	0.01 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)

References

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D.W., O'Neil, M.: Fast direct methods for Gaussian processes. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 252–265 (2015)

Anitescu, M., Chen, J., Wang, L.: A matrix-free approach for solving the parametric Gaussian process maximum likelihood problem. SIAM J. Sci. Comput. **34**(1), A240–A262 (2012)

Aune, E., Simpson, D.P., Eidsvik, J.: Parameter estimation in high dimensional gaussian distributions. Stat. Comput. **24**(2), 247–263 (2014)

Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H.: Gaussian predictive process models for large spatial data sets. J. R. Statist. Soc. Ser. B Statist. Methodol. **70**(4), 825–848 (2008)

Barbian, M.H., Assunção, R.M.: Spatial subsemble estimator for large geostatistical data. Spat. Statist. **22**, 68–88 (2017)

Benbow, S.J.: Solving generalized least-squares problems with LSQR. SIAM J. Matrix Anal. Appl. **21**(1), 166–177 (1999)

Bradley, J.R., Cressie, N., Shi, T., et al.: A comparison of spatial predictors when datasets could be very large. Statist. Surv. **10**, 100–131 (2016)

Chung, J., Saibaba, A.K.: Generalized hybrid iterative methods for large-scale Bayesian inverse problems. SIAM J. Sci. Comput. **39**(5), S24–S46 (2017)

Chung, J., Saibaba, A.K., Brown, M., Westman, E.: Efficient generalized Golub-Kahan based methods for dynamic inverse problems. Inverse Prob. **34**(2), 024005 (2018)

Cox, D.R., Snell, E.J.: Analysis of Binary Data, vol. 32. CRC Press, Cambridge (1989)

Cressie, N., Johannesson, G.: Fixed rank Kriging for very large spatial data sets. J. R. Statist. Soc. Ser. B Statist. Methodol. **70**(1), 209–226 (2008)

Datta, A., Banerjee, S., Finley, A.O., Gelfand, A.E.: Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. J. Am. Stat. Assoc. **111**(514), 800–812 (2016)

Datta, A., Banerjee, S., Finley, A.O., Gelfand, A.E.: On nearest-neighbor Gaussian process models for massive spatial data. Wiley Interdiscip. Rev. Comput. Statist. **8**(5), 162–171 (2016)

Datta, A., Banerjee, S., Finley, A.O., Hamm, N.A., Schaap, M.: Non-separable dynamic nearest neighbor Gaussian process models for large spatio-temporal data with an application to particulate matter analysis. Ann. Appl. Statist. **10**(3), 1286 (2016)

Den Hertog, D., Kleijnen, J.P., Siem, A.Y.: The correct Kriging variance estimated by bootstrapping. J. Oper. Res. Soc. **57**(4), 400–409 (2006)

Dutta, S., Mondal, D.: REML estimation with intrinsic Matérn dependence in the spatial linear mixed model. Electr. J. Statist. **10**(2), 2856–2893 (2016)

Eidsvik, J., Shaby, B.A., Reich, B.J., Wheeler, M., Niemi, J.: Estimation and prediction in spatial models with block composite likelihoods. J. Comput. Graph. Stat. **23**(2), 295–315 (2014)

Eriksson, D., Dong, K., Lee, E., Bindel, D., Wilson, A.G.: Scaling Gaussian process regression with derivatives. In: Advances in Neural Information Processing Systems, pp. 6867–6877 (2018)

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

Fuentes, M.: Approximate likelihood for large irregularly spaced spatial data. J. Am. Stat. Assoc. **102**(477), 321–331 (2007)

Furrer, R., Genton, M.G., Nychka, D.: Covariance tapering for interpolation of large spatial datasets. J. Comput. Graph. Stat. **15**(3), 502–523 (2006)

Gneiting, T., Ševčíková, H., Percival, D.B., Schlather, M., Jiang, Y.: Fast and exact simulation of large Gaussian lattice systems in $\mathbb{R}^2$: Exploring the limits. J. Comput. Graph. Stat. **15**(3), 483–501 (2006)

Graham, I.G., Kuo, F.Y., Nuyens, D., Scheichl, R., Sloan, I.H.: Analysis of circulant embedding methods for sampling stationary random fields. SIAM J. Numer. Anal. **56**(3), 1871–1895 (2018)

Gray, R.M.: Toeplitz and circulant matrices: A review. Found. Trends® Commun. Inf. Theory **2**(3), 155–239 (2006)

Guhaniyogi, R., Banerjee, S.: Meta-Kriging: Scalable Bayesian modeling and inference for massive spatial datasets. Technometrics **60**(4), 430–444 (2018)

Guinness, J.: Spectral density estimation for random fields via periodic embeddings. Biometrika **106**(2), 267–286 (2019)

Guinness, J., Fuentes, M.: Circulant embedding of approximate covariances for inference from Gaussian data on large lattices. J. Comput. Graph. Stat. **26**(1), 88–97 (2017)

Gyires, B.: Eigenwerte verallgemeinerter Toeplitzschen matrizen. Publ. Math. Debrecen. **4**, 171–179 (1956)

Heaton, M.J., Datta, A., Finley, A.O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R.B., Hammerling, D., Katzfuss, M., et al.: A case study competition among methods for analyzing large spatial data. J. Agric. Biol. Environ. Stat. **24**(3), 398–425 (2019)

Higdon, D.: Space and space-time modeling using process convolutions. In: Quantitative methods for current environmental issues, Springer, pp. 37–56 (2002)

Kang, E.L., Cressie, N.: Bayesian inference for the spatial random effects model. J. Am. Stat. Assoc. **106**(495), 972–983 (2011)

Katzfuss, M.: A multi-resolution approximation for massive spatial datasets. J. Am. Stat. Assoc. **112**(517), 201–214 (2017)

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

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

Kaufman, C.G., Schervish, M.J., Nychka, D.W.: Covariance tapering for likelihood-based estimation in large spatial data sets. J. Am. Stat. Assoc. **103**(484), 1545–1555 (2008)

Kent, J.T., Mardia, K.V.: Spectral and circulant approximations to the likelihood for stationary Gaussian random fields. J. Statist. Plan. Inference **50**(3), 379–394 (1996)

Lindgren, F., Rue, H., Lindström, J.: An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. J. R. Statist. Soc. Ser. B Statist. Methodol. **73**(4), 423–498 (2011)

Liu, H., Ong, Y.S., Shen, X., Cai, J.: When Gaussian process meets big data: A review of scalable GPs. IEEE Trans. Neural Netw. Learn. Syst. (2020)

Martino, S., Rue, H.: Implementing approximate bayesian inference using integrated nested laplace approximation: A manual for the inla program. Department of Mathematical Sciences, NTNU, Norway (2009)

Matérn, B.: Spatial variation, volume 36 of. Lecture Notes in Statistics (1960)

Minden, V., Damle, A., Ho, K.L., Ying, L.: Fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations. Multiscale Model. Simul. **15**(4), 1584–1611 (2017)

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., Sain, S.: A multiresolution Gaussian process model for the analysis of large spatial datasets. J. Comput. Graph. Stat. **24**(2), 579–599 (2015)

Paciorek, C.J., Lipshitz, B., Zhuo, W., Kaufman, C.G., Thomas, R.C., et al.: Parallelizing Gaussian Process Calculations in R. J. Statist. Softw. **63**(i10), (2015)

Rue, H., Held, L.: Gaussian Markov random fields: theory and applications. CRC Press, Cambridge (2005)

Saad, Y.: Iterative methods for sparse linear systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2003), https://doi.org/10.1137/1.9780898718003, https://doi-org.prox.lib.ncsu.edu/10.1137/1.9780898718003

Stein, M.L.: Fast and exact simulation of fractional Brownian surfaces. J. Comput. Graph. Stat. **11**(3), 587–599 (2002)

Stein, M.L.: Statistical properties of covariance tapers. J. Comput. Graph. Stat. **22**(4), 866–885 (2013)

Stein, M.L., Chi, Z., Welty, L.J.: Approximating likelihoods for large spatial data sets. J. R. Statist. Soc. Ser. B Statist. Methodol. **66**(2), 275–296 (2004)

Sun, Y., Li, B., Genton, M.G.: Geostatistics for large datasets. In: Advances and challenges in space-time modelling of natural events, Springer, pp. 55–77 (2012)

Ubaru, S., Chen, J., Saad, Y.: Fast estimation of tr(f(A)) via stochastic Lanczos quadrature. SIAM J. Matrix Anal. Appl. **38**(4), 1075–1099 (2017)

Varin, C., Reid, N., Firth, D.: An overview of composite likelihood methods. Statistica Sinica pp 5–42 (2011)

Vecchia, A.V.: Estimation and model identification for continuous spatial processes. J. R. Stat. Soc. Ser. B Methodol. **50**(2), 297–312 (1988)

Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv. Comput. Math. **4**(1), 389–396 (1995)

Widom, H.: Asymptotic behavior of block Toeplitz matrices and determinants. Adv. Math. **13**(3), 284–322 (1974)