Statistics and Computing (2023) 33:84
https://doi.org/10.1007/s11222-023-10252-0

ORIGINAL PAPER

Scalable computations for nonstationary Gaussian processes

Paul G. Beckman[1,2] · Christopher J. Geoga[2,3] · Michael L. Stein[3,4] · Mihai Anitescu[2,4]

Received: 10 May 2022 / Accepted: 4 May 2023

Abstract
Nonstationary Gaussian process models can capture complex spatially varying dependence structures in spatial data. However, the large number of observations in modern datasets makes fitting such models computationally intractable with conventional dense linear algebra. In addition, derivative-free or even first-order optimization methods can be slow to converge when estimating many spatially varying parameters. We present here a computational framework that couples an algebraic block-diagonal plus low-rank covariance matrix approximation with stochastic trace estimation to facilitate the efficient use of second-order solvers for maximum likelihood estimation of Gaussian process models with many parameters. We demonstrate the effectiveness of these methods by simultaneously fitting 192 parameters in the popular nonstationary model of Paciorek and Schervish using 107,600 sea surface temperature anomaly measurements.

Keywords Nonstationary · Spatial analysis · Optimization · Statistical computing

1 Introduction

Gaussian processes are a prevalent class of models in spatial and spatiotemporal statistics. This is due in part to the fact that the model is completely specified by the first two moments of the Gaussian distribution, so that a practitioner needs to select only a mean function and covariance function. Let $Z(\boldsymbol{x})$ be a Gaussian process with mean function $\mu(\boldsymbol{x}) = 0$ and covariance function

$$\mathrm{Cov}(Z(\boldsymbol{x}), Z(\boldsymbol{x}')) = k(\boldsymbol{x}, \boldsymbol{x}'|\boldsymbol{\theta}) \tag{1}$$

observed at locations $\{\boldsymbol{x}_i\}_{i=1}^n$ corresponding to measurements $y_i = Z(\boldsymbol{x}_i)$ with $\boldsymbol{x}_i \in \Omega \subset \mathbb{R}^d$. Here $k(\cdot,\cdot)$ is a parametric covariance function with parameters $\boldsymbol{\theta}$. Then defining the vector $\boldsymbol{y} = [y_1, ..., y_n]^\top \in \mathbb{R}^n$, we have

$$\boldsymbol{y} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta})\big), \tag{2}$$

where the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ is defined by

$$\boldsymbol{\Sigma}(\boldsymbol{\theta})_{ij} = \mathrm{Cov}(y_i, y_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j|\boldsymbol{\theta}). \tag{3}$$

A primary objective after selecting a parametric covariance model is to estimate $\boldsymbol{\theta}$ from the data. One can then predict the value of the process at unobserved locations by treating the estimated parameters as the true parameters. A standard parameter estimation method is to compute the maximum likelihood estimator (MLE), denoted $\widehat{\boldsymbol{\theta}}$, which minimizes the mean zero Gaussian negative log-likelihood function

$$-\ell(\boldsymbol{\theta}) = \frac{1}{2}\,\mathrm{logdet}\,\boldsymbol{\Sigma}(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\boldsymbol{y} + \frac{n}{2}\log(2\pi). \tag{4}$$

For the remainder of this paper we will suppress the dependence of $\boldsymbol{\Sigma}$ on $\boldsymbol{\theta}$ for notational clarity.

Computing (4) for large spatial datasets is computationally challenging, since the determinant and linear solve operations have cubic time complexity and quadratic space complexity using conventional dense linear algebra. Thus, for large n, evaluating the log-likelihood directly becomes

This material was based upon work supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) under Contract DE-AC02-06CH11347.

✉ Paul G. Beckman
paul.beckman@cims.nyu.edu

[1] Courant Institute of Mathematical Sciences, New York University, New York, USA

[2] Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, USA

[3] Department of Statistics, Rutgers University, New Brunswick, USA

[4] Department of Statistics, University of Chicago, Chicago, USA

Published online: 30 May 2023

prohibitively expensive. In response to the first of these computational challenges, a number of approximations have been proposed, including Vecchia and "nearest-neighbor Gaussian process" methods (Vecchia 1988; Stein et al. 2004; Katzfuss and Guinness 2021; Guinness 2021), matrix tapering (Furrer et al. 2006), and Markov random field approximations (Lindgren et al. 2011), which each approximate $\boldsymbol{\Sigma}$ or $\boldsymbol{\Sigma}^{-1}$ using a sparse or block-sparse matrix. Low-rank updates to a diagonal matrix have also been considered (Cressie and Johannesson 2008; Banerjee et al. 2008; Eidsvik et al. 2012; Katzfuss and Cressie 2012; Solin and Särkkä 2020), although they suffer from limitations for nonsmooth fields or when including a nugget term is not an appropriate modeling assumption (Stein 2014). Approaches using hierarchical matrices have also been proposed (Börm and Garcke 2007; Ambikasaran et al. 2015; Chen and Stein 2021; Minden et al. 2017; Litvinenko et al. 2019), which use rank-structured matrix arithmetic to efficiently evaluate (4).

In spite of the development of these scalable methods, minimizing the negative log-likelihood remains challenging depending on the covariance function and its parameterization. Even for covariance functions with few parameters, for example the stationary isotropic Matérn covariance

$$\mathscr{M}_\nu(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}}\left(\frac{2\sqrt{\nu}\,\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\rho}\right) \mathscr{K}_\nu\left(\frac{2\sqrt{\nu}\,\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\rho}\right), \tag{5}$$

where ν is a positive constant smoothness parameter and $\mathscr{K}_\nu(\cdot)$ is the modified Bessel function of the second kind of order ν, simultaneously estimating the scale σ^2 and range ρ parameters poses a significant challenge (Zhang 2004). The likelihood surface is far from convex, and even first-order methods may become trapped in nearly flat nonellipsoidal regions of the likelihood surface and fail to make meaningful progress. While a few relatively recent articles attempt to treat the optimization problem more seriously (Guinness 2021; Minden et al. 2017; Geoga et al. 2019), the norm in practice is to use derivative-free or first-order methods with finite difference derivatives.

For more complicated covariance functions that provide more flexibility and thus require more parameters, the optimization problem becomes even harder, and second-order optimization can mean the difference between an optimizer stagnating and successfully reaching the MLE (Guinness 2021; Geoga et al. 2022). The computational problem posed in this setting is that efficient linear solves and log-determinants with $\boldsymbol{\Sigma}$ are no longer sufficient. Derivatives $\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}$ need to be computed and applied scalably; and if the gradient and Fisher matrix are computed directly, matrix-matrix products of the form $\boldsymbol{\Sigma}^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}$ need to be computed efficiently. Even for data sizes in which $\boldsymbol{\Sigma}$ does not require some form of approximation, with sufficiently many parameters the computational burden of the matrix-matrix operations necessary for the gradient alone can be problematic. In this work we address the problem of second-order optimization for a covariance function with many parameters, using as our motivating example a nonstationary spatial model, which we introduce now.

For large spatial datasets it is often unrealistic to assume that process parameters are constant over the entire domain, in other words, that the process is stationary. Therefore nonstationary covariance functions in which the parameters vary in space become necessary in order to accurately capture the dependence structure of the data. One such covariance function that we will use here is derived by Paciorek and Schervish (2006) as a modification of the stationary Matérn covariance (5) and is frequently used in the nonstationary Gaussian process literature (Li and Sun 2019; Risser and Calder 2015; Banerjee et al. 2008; Sang and Huang 2012; Huang et al. 2021). We use the following anisotropic version,

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 \frac{|\boldsymbol{\Lambda}(\boldsymbol{x}_i)|^{\frac{1}{4}}\left|\boldsymbol{\Lambda}(\boldsymbol{x}_j)\right|^{\frac{1}{4}}}{\left|\frac{\boldsymbol{\Lambda}(\boldsymbol{x}_i)+\boldsymbol{\Lambda}(\boldsymbol{x}_j)}{2}\right|^{\frac{1}{2}}} \mathscr{M}_\nu\left(\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top\left(\frac{\boldsymbol{\Lambda}(\boldsymbol{x}_i) + \boldsymbol{\Lambda}(\boldsymbol{x}_j)}{2}\right)^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)}\right), \tag{6}$$

where $\boldsymbol{\Lambda}(\cdot)$ is a spatially varying function that assigns a positive definite local anisotropy matrix at each location. In fact, Stein (2011) gives an extension of this nonstationary covariance that allows for spatially varying scale $\sigma(\cdot)$ and smoothness $\nu(\cdot)$ functions. However, jointly estimating range and scale parameters can be challenging even in the simplest stationary settings (Zhang 2004), and robustly computing the derivatives of $\mathscr{K}_\nu(\cdot)$ in the smoothness parameter ν is numerically challenging, although progress on this topic has been made recently (Geoga et al. 2022). Thus we restrict our investigations in this work to models that have only nonstationary local ranges via the anisotropy parameters. While many options exist to parameterize $\boldsymbol{\Lambda}(\cdot)$, in this work we demonstrate a basis function strategy detailed in Sect. 4.1.

As one might expect, however, a basis function expansion of $\boldsymbol{\Lambda}(\cdot)$ for a highly nonstationary process on a large domain requires many parameters, which makes fitting the entire global model a difficult high-dimensional optimization problem for which derivative-free and even first-order solvers are often ineffective. A number of past works using the covariance model of Paciorek and Schervish circumvent this difficulty by fitting parameters only locally and subsequently smoothing or regressing them into a global model (Li and Sun 2019; Risser and Calder 2015; Huang et al. 2021). Without considering all the basis coefficients simultaneously, how-

ever, relationships between parameters in adjacent regions are ignored, and it can be difficult to verify the quality of the global model fit. The primary goal of our work is to present a computational framework in which this global model fitting is tractable and to consider its benefits.

We emphasize here that the two computational challenges of large data size and large parameter dimension are intimately linked. Large data collected over extensive spatiotemporal domains can require complex models in order to accurately characterize their dependence structure, and, conversely, complex models often require large data in order to accurately estimate their many parameters. Therefore, considering these two problems together is imperative to developing effective practical methods.

In this vein, we introduce methods to control the computational complexity of second-order optimization. Further, we provide and discuss an application to high-resolution sea surface temperature measurements, fitting a dataset of 107,600 measurements to a model with 192 parameters. The results of this application demonstrate that there can be substantial gains by estimating the entire global model jointly.

Our contribution is two-fold. First, we provide the necessary ingredients for second-order optimization by giving detailed accounts of the computation of gradients, Hessians, and expected Fisher information matrices in linear time and storage complexity using the *block full-scale approximation* (Snelson and Ghahramani 2007; Sang et al. 2011) with or without a nugget, which to our knowledge do not exist in the literature. Second, we show that the symmetrized stochastic trace estimation method of Geoga et al. (2019) allows us to estimate the gradient and expected Fisher matrix using a single pass over the derivative matrices, which demonstrates significant performance gains in practice and makes possible the simultaneous optimization of all parameters. With these strategies, despite the computational burden of a highly expensive covariance function, we demonstrate that fitting large datasets with many-parameter models can be done effectively.

2 Block full-scale approximation

While a number of past works have approximated the covariance matrix as the sum of a diagonal matrix and a low-rank matrix (Cressie and Johannesson 2008; Banerjee et al. 2008; Eidsvik et al. 2012; Katzfuss and Cressie 2012; Solin and Särkkä 2020), these models often fail to capture short-range behavior of the data. To improve the model fit, one can add a banded or block diagonal matrix to the low-rank approximation. We consider here a particular algebraic approximation of this type referred to as the *partially independent conditional* approximation by Snelson and Ghahramani (2007) or the *block full-scale approximation* by Sang et al. (2011). This approximation allows one to modulate the block size and off-diagonal rank independently to capture both smooth long-range dependence and rough local dependence while still allowing computations that scale linearly in the number of observations.

Let $N = \{1, 2, ..., n\}$ indicate the index set of all observations, and take I and J to be subsets of N. Let $\boldsymbol{\Sigma}_{IJ}$ denote the submatrix of $\boldsymbol{\Sigma}$ corresponding to rows I and columns J. We start by ordering the observation locations using a k-d tree—a binary space partitioning tree that iteratively bisects $\mathbb{R}^d$ along axis-aligned hyperplanes (Bentley 1975). We then use the k-d tree ordering to choose a set of p landmark points $\boldsymbol{X}_P = \{\boldsymbol{x}_i\}_{i \in P}$ indexed by $P \subset N$ that are roughly equispaced over the spatial domain, and we construct the low-rank Nyström approximation to $\boldsymbol{\Sigma}$ given by $\boldsymbol{\Sigma}_{NP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{NP}^{\top} \in \mathbb{R}^{n \times n}$. We note that the landmark points $\boldsymbol{X}_P$ used in the Nyström approximation need not be observation points, and could be any arbitrary "knot" locations in the domain, but for the remainder of this paper we assume we are using observation locations as they are simple to choose and result in concise block matrix expressions.

Next, using the same k-d tree, we partition the data into disjoint blocks of observations at nearby spatial locations. Let block ℓ consist of observations indexed by $B_\ell \subset N$ for $\ell = 1, ..., m$, where m is the total number of blocks, and denote the collection of these block index sets as $B = \{B_1, ..., B_m\}$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, define the block diagonalization operator with block structure B as

$$\left[\operatorname{blkdiag}_B(\boldsymbol{A})\right]_{ij} = \begin{cases} \boldsymbol{A}_{ij} & \text{for} i, j \in B_\ell \text{ for some } \ell \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We use this operator to construct a block diagonal correction term that takes the Nyström approximation to the exact values. This yields our approximate covariance matrix

$$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{NP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{NP}^{\top} + \operatorname{blkdiag}_B\left(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{NP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{NP}^{\top}\right), \tag{8}$$

where the block diagonal structure allows us to capture short-range covariances exactly within disjoint local neighborhoods. Alternatively, one can view this approximation as a two-level approximation to the covariance function given by

$$\tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} k(\boldsymbol{x}_i, \boldsymbol{x}_j) & \text{for } i, j \in B_\ell \text{ for some } \ell \\ k(\boldsymbol{X}_P, \boldsymbol{x}_i)^{\top} & \\ k(\boldsymbol{X}_P, \boldsymbol{X}_P)^{-1}\, k(\boldsymbol{X}_P, \boldsymbol{x}_j) & \text{otherwise.} \end{cases} \tag{9}$$

This approximation is equivalent to assuming that observations in different neighborhoods are conditionally independent given the observations at the landmark points $\boldsymbol{X}_P$

(Snelson and Ghahramani 2007). Considering this conditional structure, one can also see this approximation as a special case of the Vecchia approximation (Katzfuss and Guinness 2021).

To leverage recent advances in scalable hierarchical matrix operations and factorizations, Geoga et al. (2019) used the Nyström approximation to compress off-diagonal blocks within a hierarchical off-diagonal low-rank (HODLR) approximation to the covariance matrix that could be assembled and factorized in quasilinear time and storage complexity. In that work, a set of p landmark points $\boldsymbol{X}_P = \{\boldsymbol{x}_i\}_{i\in P}$ are selected from the data and indexed by $P \subset N$, and each off-diagonal block is approximated using the low-rank Nyström scheme $\tilde{\boldsymbol{\Sigma}}_{IJ} = \boldsymbol{\Sigma}_{IP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{JP}^{\top}$. Crucially, the set P of landmark points is fixed and used in every off-diagonal block. Looking entrywise, this approach induces the two-level matrix approximation

$$\tilde{\boldsymbol{\Sigma}}_{ij} = \begin{cases} \boldsymbol{\Sigma}_{ij} & \text{for } i, j \in B_\ell \text{ for some } \ell \\ \boldsymbol{\Sigma}_{Pi}^{\top}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{Pj} & \text{otherwise,} \end{cases} \tag{10}$$

where the block diagonal entries are exactly the entries of the full covariance matrix $\boldsymbol{\Sigma}$ and the nonleaf entries are given by a low-rank approximation. This is precisely the two-level approximation (8). The simplification from a Nyström-based hierarchical covariance to a two-level covariance is also discussed by Chen et al. (2017).

3 Linear complexity computations

Here we discuss how to exploit the two-level covariance approximation described above to obtain direct linear complexity computations of the likelihood and its derivatives. We derive complexity estimates in terms of the rank of the Nyström approximation and the block sizes in the block diagonal term. We can tune these rank and block size parameters to trade off between approximation accuracy and computational complexity.

For the remainder of this paper, we assume that the mean function of the model is zero everywhere and focus exclusively on covariance matrices. As in the real data application in Sect. 4, it is fairly common to study anomaly fields of climatological variables and treat these as having mean zero for a mean function specified up to some vector of unknown linear parameters. The matrix calculus for derivatives with respect to these parameters is much simpler and poses no computational concern.

3.1 Computing the log-likelihood

In order to perform maximum likelihood estimation, our first concern is to efficiently compute the negative log-likelihood using our approximate covariance matrix $\tilde{\boldsymbol{\Sigma}}$, given up to an additive constant by

$$-\ell(\boldsymbol{\theta}) = \frac{1}{2}\operatorname{logdet}\tilde{\boldsymbol{\Sigma}} + \frac{1}{2}\boldsymbol{y}^{\top}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}. \tag{11}$$

This requires the computation of the determinant of $\tilde{\boldsymbol{\Sigma}}$ as well as the linear solve $\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}$. For this purpose, one might hope to use the matrix determinant lemma

$$\det\left(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{V}^{\top}\right) = \det\left(\boldsymbol{I} + \boldsymbol{V}^{\top}\boldsymbol{A}^{-1}\boldsymbol{U}\right)\det\left(\boldsymbol{A}\right) \tag{12}$$

and the Sherman-Woodbury-Morrison formula

$$(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{V}^{\top})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{V}^{\top}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}\boldsymbol{A}^{-1} \tag{13}$$

which take advantage of the low-rank update structure for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times p}$ to reduce the complexity of these computations. Note, however, that for the Nyström approximation (8) we have $\tilde{\boldsymbol{\Sigma}}_{PP} = \boldsymbol{\Sigma}_{PP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{PP}^{\top} = \boldsymbol{\Sigma}_{PP}$. In other words, the Nyström approximation is exact on rows and columns corresponding to landmark points, so $\operatorname{blkdiag}_B\left(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{NP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{NP}^{\top}\right)$ is zero on these rows and columns. In particular it is not invertible, and thus we cannot use formulas (12) and (13). Previous works add a nugget $\sigma^2\boldsymbol{I}$ to $\tilde{\boldsymbol{\Sigma}}$, circumventing this issue (Snelson and Ghahramani 2007; Sang et al. 2011). However, this is not strictly necessary. We can remedy the rank deficiency without a nugget by defining a matrix $\boldsymbol{\Pi}$ which permutes the landmark indices P to the last p indices $\{N-p+1, ..., N\}$, leaving all non-landmark points in the same order. Let $Q = N \setminus P$ denote the index set of non-landmark indices, and let $B' = [B'_1, ..., B'_m]$ be a new collection of block index sets which partition Q and thus contain no Nyström points. One method of constructing B' is to simply remove any Nyström indices from B, namely $B'_\ell = B_\ell \cap Q$. Then we have

$$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Pi}^{\top}\begin{bmatrix} \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^{\top} + \operatorname{blkdiag}_{B'}\left(\boldsymbol{\Sigma}_{QQ} - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^{\top}\right) & \boldsymbol{\Sigma}_{QP} \\ \boldsymbol{\Sigma}_{QP}^{\top} & \boldsymbol{\Sigma}_{PP} \end{bmatrix}\boldsymbol{\Pi}. \tag{14}$$

We note that the Nyström rank $p \ll n$, and thus the $(n-p)\times(n-p)$ upper left block of the permuted matrix contains the vast majority of the covariance information between observations, and the other blocks are small dense matrices.

This permuted representation yields efficient and convenient computations, since the block diagonal correction matrix in the upper left block is now full rank. For ease of notation, we will write this matrix as

$$\boldsymbol{D} = \operatorname{blkdiag}_{B'}\left(\boldsymbol{\Sigma}_{QQ} - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^{\top}\right). \tag{15}$$

Returning to the determinant of $\tilde{\boldsymbol{\Sigma}}$, we see that the block diagonal correction matrix is the Schur complement of $\boldsymbol{\Sigma}_{PP}$ in the permuted matrix, and thus we can compute the determinant of the approximate covariance matrix using the matrix determinant lemma as

$$\det\left(\tilde{\boldsymbol{\Sigma}}\right) = \det(\boldsymbol{D})\det(\boldsymbol{\Sigma}_{PP}). \tag{16}$$

Here and for the remainder of this section we will assume for ease of analysis that the block structure B' consists of equally sized blocks of size $b = \left|B'_\ell\right|$ for all $\ell = 1, ..., m$. We will also assume we have factorized $\boldsymbol{\Sigma}_{PP}$ and $\boldsymbol{D}$ as a precomputation step requiring $\mathcal{O}(p^3 + nb^2)$ work. Therefore, computing the determinant of $\tilde{\boldsymbol{\Sigma}}$ requires computing the determinant of one factorized $p \times p$ matrix and n/b determinants of factorized $b \times b$ blocks, yielding $\mathcal{O}(n)$ complexity overall.

The linear solve $\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}$ can also be computed in a convenient way that leverages Schur complements. Defining the permuted vector $\boldsymbol{z} = \boldsymbol{\Pi}\boldsymbol{y}$, we solve the permuted linear system

$$\begin{aligned}
\begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_2\end{bmatrix} &= \begin{bmatrix}\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top + \boldsymbol{D} & \boldsymbol{\Sigma}_{QP}\\ \boldsymbol{\Sigma}_{QP}^\top & \boldsymbol{\Sigma}_{PP}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{z}_1\\ \boldsymbol{z}_2\end{bmatrix}\\
\boldsymbol{w}_1 &= \boldsymbol{D}^{-1}\left(\boldsymbol{z}_1 - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{z}_2\right)\\
\boldsymbol{w}_2 &= \boldsymbol{\Sigma}_{PP}^{-1}\left(\boldsymbol{z}_2 - \boldsymbol{\Sigma}_{QP}^\top\boldsymbol{w}_1\right)\\
\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y} &= \boldsymbol{\Pi}^\top\boldsymbol{w},
\end{aligned} \tag{17}$$

which requires $\mathcal{O}(np + nb)$ work to perform the matrix–vector products and solve the linear systems. Continuing on to study derivatives of (11) in much the same way, we show that even matrix-matrix products with this rank structure can be worked with conveniently and efficiently, from which scalability of all the required linear algebra follows.

3.2 Computing the gradient

To employ gradient-based optimization algorithms for maximum likelihood estimation, we must compute the gradient of the negative log-likelihood (4). Each component of the gradient is

$$\begin{aligned}
\left[-\nabla\ell(\boldsymbol{\theta})\right]_j = &\frac{1}{2}\operatorname{tr}\left[\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}\right)\right]\\
&-\frac{1}{2}\boldsymbol{y}^\top\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}\right)\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}.
\end{aligned} \tag{18}$$

Alternative rank-structured approximations to $\boldsymbol{\Sigma}$ are often constructed using early-terminating pivoted factorizations and are thus not differentiable with respect to the kernel parameters. As a result, one must introduce an additional approximation to the derivative matrices $\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial\theta_j}$. For example, the hierarchical matrix method by Minden et al. (2017) computes an independent hierarchical approximation to each $\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial\theta_j}$ in order to compute the trace term in the gradient of the log-likelihood. In contrast, Geoga et al. (2019) use an algebraic hierarchical covariance matrix approximation for which the derivatives matrices $\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial\theta_j}$ can be computed exactly in quasilinear complexity, but they must use stochastic estimators to compute the trace term efficiently.

The key observation for efficient computation of the gradient for the block full-scale approximation is that $\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}$ has a similar rank structure to $\tilde{\boldsymbol{\Sigma}}$. Since the Nyström factors $\boldsymbol{\Sigma}_{QP}$ and $\boldsymbol{\Sigma}_{PP}$ are simply submatrices of the covariance matrix $\boldsymbol{\Sigma}$, we can compute the derivatives of these factors using the derivatives of the covariance function. Following basic matrix differentiation, we have

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\left(\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top\right) = &\left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top -\\
&\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top + \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\right)^\top.
\end{aligned} \tag{19}$$

Since the second term on the right side shares a factor with each of the others, the derivative of the rank-p Nyström approximation has rank at most $2p$.

The derivative of the approximate covariance matrix $\tilde{\boldsymbol{\Sigma}}$ is then given by

$$\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j} = \boldsymbol{\Pi}^\top\begin{bmatrix}\frac{\partial}{\partial\theta_j}\left(\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top\right) + \frac{\partial\boldsymbol{D}}{\partial\theta_j} & \frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\\ \frac{\partial\boldsymbol{\Sigma}_{QP}^\top}{\partial\theta_j} & \frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\end{bmatrix}\boldsymbol{\Pi}, \tag{20}$$

which has the same block rank structure as $\tilde{\boldsymbol{\Sigma}}$ except that the low-rank portion of the upper left block has at most double the rank.

Given this shared rank structure, we can compute the matrix-matrix linear solve $\tilde{\boldsymbol{\Sigma}}^{-1}(\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j})$ with Schur complements as follows:

$$\begin{aligned}
\begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{W}_3\\ \boldsymbol{W}_2 & \boldsymbol{W}_4\end{bmatrix} &= \begin{bmatrix}\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top + \boldsymbol{D} & \boldsymbol{\Sigma}_{QP}\\ \boldsymbol{\Sigma}_{QP}^\top & \boldsymbol{\Sigma}_{PP}\end{bmatrix}^{-1}\\
&\quad\times\begin{bmatrix}\frac{\partial}{\partial\theta_j}\left(\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top\right) + \frac{\partial\boldsymbol{D}}{\partial\theta_j} & \frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\\ \frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}^\top & \frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\end{bmatrix}\\
\boldsymbol{W}_1 &= \boldsymbol{D}^{-1}\left(\frac{\partial}{\partial\theta_j}\left(\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top\right) + \frac{\partial\boldsymbol{D}}{\partial\theta_j}\right.
\end{aligned}$$

$$- \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\frac{\partial \boldsymbol{\Sigma}_{QP}}{\partial \theta_j}^{\top}\Bigg)$$

$$\boldsymbol{W}_2 = \boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}_{QP}}{\partial \theta_j}^{\top} - \boldsymbol{\Sigma}_{QP}^{\top}\boldsymbol{W}_1\right)$$

$$\boldsymbol{W}_3 = \boldsymbol{D}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}_{QP}}{\partial \theta_j} - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\frac{\partial \boldsymbol{\Sigma}_{PP}}{\partial \theta_j}\right)$$

$$\boldsymbol{W}_4 = \boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}_{PP}}{\partial \theta_j} - \boldsymbol{\Sigma}_{QP}^{\top}\boldsymbol{W}_3\right)$$

$$\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\right) = \boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^{\top}. \tag{21}$$

This requires only linear solves and matrix-matrix products involving $n \times n$ block diagonal matrices, $n \times p$ low-rank factors, and $p \times p$ matrices, resulting in $\mathcal{O}(np^2 + nb^2)$ complexity. In addition, the resulting matrix $\tilde{\boldsymbol{\Sigma}}^{-1}(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j})$ has the same permuted block diagonal plus low-rank structure as $\tilde{\boldsymbol{\Sigma}}$, where $\boldsymbol{W}_1 \in \mathbb{R}^{(n-p)\times(n-p)}$ is a matrix of rank at most $2p$ plus a block diagonal term and $\boldsymbol{W}_2 \in \mathbb{R}^{(n-p)\times p}$, $\boldsymbol{W}_3 \in \mathbb{R}^{p\times(n-p)}$, and $\boldsymbol{W}_4 \in \mathbb{R}^{p\times p}$ are small dense matrices. The preservation of this rank structure under linear solves will prove useful for computing second-order information in the next section.

The remaining inner product term in the gradient of the negative log-likelihood (18) can be computed in $\mathcal{O}(np + nb)$ time by using Eq. (17) along with a straightforward block matrix–vector product. This allows us to compute each entry of the gradient in linear complexity in n.

3.3 Computing the Fisher matrix

We can employ the linear solve method above to compute the entries of the expected Fisher information matrix given by

$$\mathcal{I}_{jk} = \frac{1}{2}\operatorname{tr}\left[\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\right)\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_k}\right)\right]. \tag{22}$$

Computing the terms $\tilde{\boldsymbol{\Sigma}}^{-1}(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j})$ and $\tilde{\boldsymbol{\Sigma}}^{-1}(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_k})$ using equation (21), applying a straightforward block matrix-matrix product, and computing the trace of the resulting rank-structured matrix, we obtain $\mathcal{O}(np^2 + nb^2)$ complexity per entry. Efficient methods for computing $\mathcal{I}$ facilitate the use of Fisher scoring algorithms to obtain the MLE and can be used to produce confidence intervals for estimated parameters. Analogous methods can be used to compute the Hessian of the negative log-likelihood for use in Newton-based optimization routines. See the appendix.

3.4 Symmetrized trace estimation

Although the above computations of the gradient and the Fisher and Hessian matrices have the desired linear scaling in the data size n, the matrix-matrix solves and products in the trace terms require a large number of intermediate allocations, which are computationally expensive in practice. To provide fast unbiased estimates of these trace terms, we rely on a sample average approximation based on the Hutchinson estimator (Hutchinson 1989)

$$\operatorname{tr}(\boldsymbol{A}) \approx \frac{1}{s}\sum_{\ell=1}^{s}\boldsymbol{u}_\ell^{\top}\boldsymbol{A}\boldsymbol{u}_\ell \tag{23}$$

with s samples, where $\boldsymbol{u}_\ell$ are independent symmetric Bernoulli vectors, although alternatives exist (see Stein et al. 2013). In particular, factorizing the approximate covariance as $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{W}\boldsymbol{W}^{\top}$, one can use the symmetrized estimator presented by Stein et al. (2013), which is given by

$$\begin{aligned}\operatorname{tr}\left[\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\right)\right] &= \operatorname{tr}\left[\boldsymbol{W}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\right)\boldsymbol{W}^{-\top}\right] \\ &\approx \frac{1}{s}\sum_{\ell=1}^{s}\boldsymbol{u}_\ell^{\top}\boldsymbol{W}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\right)\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell.\end{aligned} \tag{24}$$

Stein et al. (2013) prove a bound on the variance of this estimator that is at least as strong as the variance bound on the nonsymmetrized version, and Geoga et al. (2019) provide numerical results that indicate greatly improved accuracy with the symmetrized estimator compared with the standard Hutchinson procedure.

An analogous symmetrized estimator for the trace terms in the Fisher matrix and Hessian that require only a small number of linear solves with $\boldsymbol{W}$ and matrix–vector products with $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}$ is constructed in Geoga et al. (2019). The Fisher matrix estimator can be written as

$$\begin{aligned}\mathcal{I}_{jk} \approx \frac{1}{4s}\sum_{\ell=1}^{s}\boldsymbol{u}_\ell^{\top}\boldsymbol{W}^{-1}&\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j} + \frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_k}\right)\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j} + \frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_k}\right) \\ &\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell - \frac{1}{2}\mathcal{I}_{jj} - \frac{1}{2}\mathcal{I}_{kk},\end{aligned} \tag{25}$$

where the diagonal terms $\mathcal{I}_{jj}$ and $\mathcal{I}_{kk}$ can be estimated in a trivially symmetric way. This gives fast stochastic estimators of all quantities necessary for gradient-based and second-order optimization solvers, which can be computed in $\mathcal{O}(snp+snb)$ complexity per entry since matrix–vector products with $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}$ and linear solves with $\boldsymbol{W}$ require $\mathcal{O}(np + nb)$ work, which we now show.

3.5 Symmetric factor computation

The remaining

concern is to obtain a rank-structured symmetric factor $\boldsymbol{W}$ for our two-level approximate covariance matrix $\tilde{\boldsymbol{\Sigma}}$. The principal challenge is the upper left block, which we hope to factorize as

$$\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^{\top} + \boldsymbol{D} = \Big(\boldsymbol{X}\boldsymbol{Y}^{\top} + \boldsymbol{B}\Big)\Big(\boldsymbol{X}\boldsymbol{Y}^{\top} + \boldsymbol{B}\Big)^{\top} \tag{26}$$

for $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{(n-p)\times p}$ and $\boldsymbol{B}$ a block diagonal matrix with the same block structure B' as $\boldsymbol{D}$. As discussed in Sect. 2, our matrix structure is a two-level special case of the HODLR format; thus we use a single step of the symmetric factorization algorithm of Ambikasaran et al. (2014), which can be written concisely by computing Cholesky factors

$$\begin{aligned}
\boldsymbol{B}\boldsymbol{B}^{\top} &= \boldsymbol{D} \\
\boldsymbol{L}\boldsymbol{L}^{\top} &= \big(\boldsymbol{B}^{-1}\boldsymbol{\Sigma}_{QP}\big)^{\top}\big(\boldsymbol{B}^{-1}\boldsymbol{\Sigma}_{QP}\big) \\
\boldsymbol{M}\boldsymbol{M}^{\top} &= \boldsymbol{I} + \boldsymbol{L}^{\top}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{L} \\
\boldsymbol{X} &= \boldsymbol{\Sigma}_{QP} \\
\boldsymbol{Y} &= \boldsymbol{L}^{-\top}\big(\boldsymbol{M} - \boldsymbol{I}\big)\boldsymbol{L}^{-1}\big(\boldsymbol{B}^{-1}\boldsymbol{\Sigma}_{QP}\big)^{\top}.
\end{aligned} \tag{27}$$

This requires only Cholesky factorizations of block diagonal matrices and $p \times p$ matrices, as well as linear solves and matrix-matrix products involving $p \times p$ and $(n-p) \times p$ matrices, and thus can be computed in $\mathcal{O}(np^2 + nb^2)$ time. The symmetric factor $\boldsymbol{W}$ is then given by

$$\boldsymbol{W} = \boldsymbol{\Pi}^{\top}\begin{bmatrix} \boldsymbol{X}\boldsymbol{Y}^{\top} + \boldsymbol{B} & 0 \\ \boldsymbol{Z}^{\top} & \boldsymbol{G} \end{bmatrix}\boldsymbol{\Pi} \tag{28}$$

where the remaining blocks are defined by

$$\begin{aligned}
\boldsymbol{Z} &= \Big(\boldsymbol{X}\boldsymbol{Y}^{\top} + \boldsymbol{B}\Big)^{-1}\boldsymbol{\Sigma}_{QP} \\
\boldsymbol{G}\boldsymbol{G}^{\top} &= \boldsymbol{\Sigma}_{PP} - \boldsymbol{Z}^{\top}\boldsymbol{Z}
\end{aligned} \tag{29}$$

with $\boldsymbol{Z} \in \mathbb{R}^{(n-p)\times p}, \boldsymbol{G} \in \mathbb{R}^{p\times p}$. Since $\boldsymbol{W}$ maintains the same permuted block diagonal plus low-rank structure as $\tilde{\boldsymbol{\Sigma}}$, we can compute matrix–vector products and linear solves with $\boldsymbol{W}$ in $\mathcal{O}(np + nb)$ time using the Sherman-Woodbury-Morrison formula. This facilitates the SAA approximations to the gradient, Fisher matrix, and Hessian discussed above and provides a fast sampling method for the process.

3.6 Prediction and conditional distributions

In addition to fast likelihood computations, the rank structure of $\tilde{\boldsymbol{\Sigma}}$ facilitates fast kriging and results in a conditional covariance matrix with the same rank structure as $\tilde{\boldsymbol{\Sigma}}$. Given a set of locations $\{\boldsymbol{x}_i^*\}_{i=1}^{n_*}$ indexed by N_* with corresponding process values $y_i^* = Z(\boldsymbol{x}_i^*)$, we define the vector $\boldsymbol{y}_* = [y_1^*, ..., y_{n_*}^*]$ and wish to compute the conditional distribution

$$\boldsymbol{y}_*|\boldsymbol{y} \sim N(\tilde{\boldsymbol{\Sigma}}_*^{\top}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}, \tilde{\boldsymbol{\Sigma}}_{**} - \tilde{\boldsymbol{\Sigma}}_*^{\top}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\Sigma}}_*). \tag{30}$$

where $\big(\tilde{\boldsymbol{\Sigma}}_*\big)_{ij} = \tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j^*)$ and $\big(\tilde{\boldsymbol{\Sigma}}_{**}\big)_{ij} = \tilde{k}(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*)$. We assign each point $\boldsymbol{x}_i^*$ to a block from the observed data, for example by taking the block with centroid nearest to $\boldsymbol{x}_i^*$, resulting in a collection of block index sets $B_* = [B_1^*, ..., B_m^*]$ that partition N_*. To simplify the discussion of computational complexity, we assume this results in blocks of equal size $b_* = \big|B_\ell^*\big|$. Recalling the permutation $\boldsymbol{\Pi}$ that orders the landmark points to the last indices, we obtain covariance matrices of the form

$$\begin{aligned}
\tilde{\boldsymbol{\Sigma}}_{**} = \boldsymbol{\Sigma}_{N_*P}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{N_*P}^{\top} + \operatorname{blkdiag}_{B_*} \\
\Big(\boldsymbol{\Sigma}_{N_*N_*} - \boldsymbol{\Sigma}_{N_*P}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{N_*P}^{\top}\Big)
\end{aligned} \tag{31}$$

$$\tilde{\boldsymbol{\Sigma}}_* = \boldsymbol{\Pi}^{\top}\begin{bmatrix} \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{N_*P}^{\top} + \operatorname{blkdiag}_{B'B_*} \\ \Big(\boldsymbol{\Sigma}_{QN_*} - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{N_*P}^{\top}\Big) \\ \boldsymbol{\Sigma}_{N_*P}^{\top} \end{bmatrix}, \tag{32}$$

where the nonsymmetric block diagonalization operator is defined by

$$\Big[\operatorname{blkdiag}_{B'B_*}(\boldsymbol{A})\Big]_{ij} = \begin{cases} \boldsymbol{A}_{ij} & \text{for} i \in B'_\ell \text{and} j \in B^*_\ell \text{for some} \ell \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

We see that these covariance matrices have rank structures that are minor variations on the block diagonal plus low-rank structure we have used in the observed data covariance matrix, its derivatives, and its symmetric factor.

To compute the term $\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}$ in the conditional mean, we use the linear solve (17) followed by a straightforward block matrix–vector product with $\tilde{\boldsymbol{\Sigma}}_*$, which has complexity $\mathcal{O}(nb_* + np + n_*p)$.

To compute the term $\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\Sigma}}_*$ in the conditional covariance, we use the first column of the structured matrix-matrix solve (21). This yields a matrix with the same block structure as (32). We then compute the block matrix-matrix product with $\tilde{\boldsymbol{\Sigma}}_*^{\top}$, which has complexity $\mathcal{O}(n\,b_*^2 + np^2 + n_*p)$. Importantly, the resulting conditional covariance matrix is a block diagonal plus a matrix of rank at most $4p$. Thus we can afford to compute and store it, facilitating further computations such as symmetric factorization using (27) and thus yielding conditional simulations in linear complexity.

3.7 Numerical verification of $\mathcal{O}(n)$ complexity

Before applying the methods developed above to a nonstationary process, we demonstrate their linear scaling using a

simple stationary process. We consider fitting n observations at locations selected uniformly at random in $[0, 1]^2$ using the stationary isotropic Matérn covariance (5) with fixed $\nu = 1$, where $\boldsymbol{\theta} = [\sigma^2, \rho]$ are being estimated. We fix the block size $b = 128$ and the Nyström rank $p = 32$. For various n we then time the computation of the approximate covariance matrix $\tilde{\boldsymbol{\Sigma}}$, the likelihood ℓ, the symmetric factor $\boldsymbol{W}$, the gradient $\nabla\ell$, and the Fisher information matrix $\mathscr{I}$, as well as computation of the conditional distribution of $\boldsymbol{y}_*|\boldsymbol{y}$ consisting of the mean and rank-structured conditional covariance matrix at 512 sites also selected uniformly at random in $[0, 1]^2$. Figure 1 shows that all the aforementioned computations scale linearly as expected and gives timings for our implementations on a single core of an Intel Xeon CPU E5-2650 @ 2.00 GHz machine.

4 Numerical results

We now test these scalable computations in the nonstationary setting, performing parameter estimation for both a simulated dataset as well as a large satellite sea surface temperature dataset. The effect of the Nyström rank p and block size b parameters on the approximation quality is studied, and our approach is compared to an alternative global fitting method (Guinness 2021) from the literature.

4.1 Covariance model and parameterization

Recall the nonstationary anisotropic model described in Sect. 1, which is given by

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 \frac{|\boldsymbol{\Lambda}(\boldsymbol{x}_i)|^{\frac{1}{4}} \left|\boldsymbol{\Lambda}(\boldsymbol{x}_j)\right|^{\frac{1}{4}}}{\left|\frac{\boldsymbol{\Lambda}(\boldsymbol{x}_i)+\boldsymbol{\Lambda}(\boldsymbol{x}_j)}{2}\right|^{\frac{1}{2}}} \mathscr{M}_\nu\left(\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \left(\frac{\boldsymbol{\Lambda}(\boldsymbol{x}_i) + \boldsymbol{\Lambda}(\boldsymbol{x}_j)}{2}\right)^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)}\right). \quad (6)$$

To estimate the spatially-varying anisotropy function $\boldsymbol{\Lambda}(\cdot)$, we expand it in a normalized radial basis

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \sum_{i=1}^{m} \phi_i(\boldsymbol{x})\, \boldsymbol{\Lambda}_i \quad (39)$$

$$\phi_i(\boldsymbol{x}) = \frac{e^{-\|\boldsymbol{x}-\boldsymbol{a}_i\|^2/c^2}}{\sum_{j=1}^{m} e^{-\|\boldsymbol{x}-\boldsymbol{a}_j\|^2/c^2}} \quad (40)$$

using squared exponential bases with a width parameter c, and we estimate the positive definite matrices $\boldsymbol{\Lambda}_i$. In particular, we express the anisotropy matrices in terms of their Cholesky factors

$$\boldsymbol{\Lambda}_i = \boldsymbol{L}_i \boldsymbol{L}_i^\top \quad (41)$$

$$\boldsymbol{L}_i = \begin{bmatrix} \ell_i^{(1,1)} & 0 \\ \ell_i^{(2,1)} & \ell_i^{(2,2)} \end{bmatrix}, \quad (42)$$

which guarantees positive definiteness. We find also that it is necessary to parameterize the log of the diagonal elements in order to enforce uniqueness of the Cholesky factors and avoid identifiability issues. This results in the parameter vector

$$\boldsymbol{\theta} = [\log \ell_1^{(1,1)}, \ell_1^{(2,1)}, \log \ell_1^{(2,2)}, \ldots, \log \ell_m^{(1,1)}, \ell_m^{(2,1)}, \log \ell_m^{(2,2)}] \in \mathbb{R}^{3m}. \quad (43)$$

Since the nonnegative linear combination of positive definite matrices is positive definite, we have obtained a parameterization in which $\boldsymbol{\Lambda}(\boldsymbol{x})$ is positive definite for all $\boldsymbol{x} \in \Omega$. We fix the smoothness parameter $\nu = 1$ because preliminary estimation of a stationary Matérn model yielded an MLE near this value, and we are interested here primarily in the nonstationary anisotropy parameters. We estimate the scale parameter σ^2 using the *profile likelihood* by fixing $\sigma^2 = 1$, estimating the anisotropy parameters $\boldsymbol{\theta}$ and then computing the optimal scale parameter, which is given in closed form by $\sigma^2 = \frac{1}{n}\boldsymbol{y}^\top \tilde{\boldsymbol{\Sigma}}(1, \boldsymbol{\theta})^{-1}\boldsymbol{y}$.

4.2 Second order trust-region optimization

To compute the MLE for all models in the following sections, we use our own implementation of the trust-region algorithm adapted directly from Wright and Nocedal (1999). At iteration k this algorithm minimizes a quadratic approximation to the negative log-likelihood

$$\min_{\boldsymbol{p}\in\mathbb{R}^{3m}} -\ell(\boldsymbol{\theta}^{(k)}) - \nabla\ell(\boldsymbol{\theta}^{(k)})^\top \boldsymbol{p} - \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{B}_k \boldsymbol{p} \quad \text{s.t.} \quad \|\boldsymbol{p}\| \le \Delta_k \quad (44)$$

where $\boldsymbol{\theta}^{(k)}$ are the current parameters, $\boldsymbol{B}_k$ is an approximation to the Hessian $\nabla^2\ell(\boldsymbol{\theta}^{(k)})$, and Δ_k is a radius parameter which indicates the size of the region in that this quadratic objective is a good approximation to the true negative log-likelihood. Trust-region algorithms are known to converge to stationary points for various approximate solutions of the subproblem (44) and for various Hessian approximations $\boldsymbol{B}_k$ as long as $\boldsymbol{B}_k$ is bounded. In the spirit of Fisher scoring, we use the symmetrized stochastic estimate of the full Fisher matrix $\boldsymbol{B}_k = \mathscr{I}(\boldsymbol{\theta}^{(k)})$ discussed in Sect. 3.4 and solve the subproblem (44) using a Newton-based iterative method (Wright and Nocedal (1999), Section 4.3) which is inexpensive and effective for problems of this size.

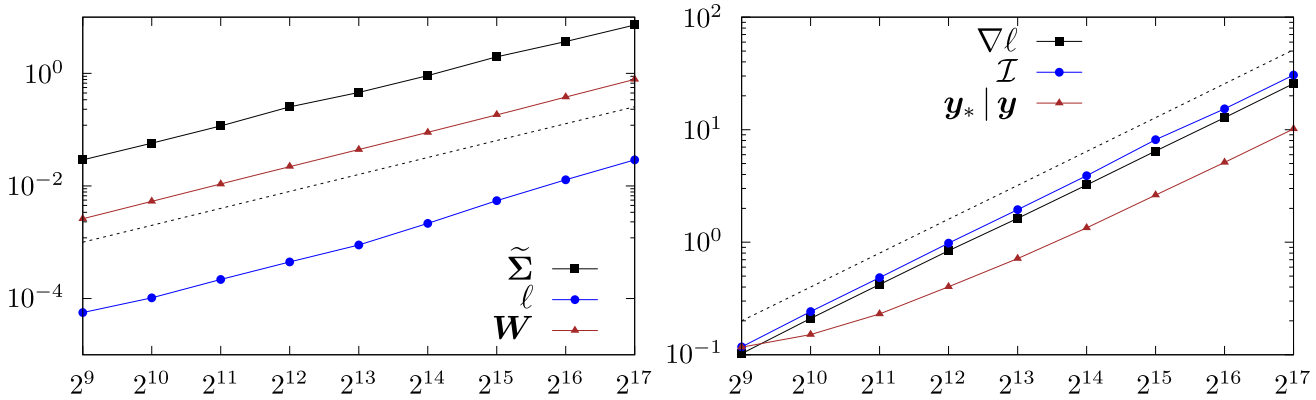

Fig. 1 Linear scaling of covariance matrix construction, symmetric factorization, and log-likelihood evaluation (left); and of gradient, Fisher matrix, and conditional distribution computation at 512 unobserved locations (right). Dotted black lines show $\mathcal{O}(n)$ scaling for reference

4.3 Approximation accuracy study

In order to compare the performance of parameter estimation using the block full-scale approximation with estimation using the exact likelihood, we generate synthetic nonstationary data using the covariance function (6). We fix the true scale and smoothness $\sigma^2 = \nu = 1$ and specify the true local anisotropy $\mathbf{\Lambda}(\cdot)$ using smooth functions of its rotation angle and eigenvalues. See the appendix for the full specification. We compute $S = 10$ i.i.d. samples from this model using points on a 64×64 grid in $[0, 1]^2$. To estimate $\mathbf{\Lambda}(\cdot)$ from each of these independent samples, we use the Choleksy-based RBF model described in Sect. 4.1, fix $\sigma^2 = \nu = 1$, and place RBFs with width $c = 0.2$ on a 3 by 3 grid. See the left plot in Fig. 2 for a plot of one of the sample paths with RBF centers. This results in 27 parameters to fit with maximum likelihood using the trust-region algorithm described in the previous section and the block full-scale computations outlined in Sects. 3.1–3.3.

We provide two metrics for the quality our parameter estimates as we vary the block size b and Nyström rank p in the block full-scale approximation. For the first metric, we generate i.i.d. samples $\boldsymbol{y}^{(s)}$ for $s = 1, ..., S$ from our nonstationary spatial model and compute the approximate MLE $\widehat{\boldsymbol{\theta}}^{(s)}_{b,p}$ for each sample independently. We then compute the the exact log-likelihood $\ell(\cdot)$ and the approximate log-likelihood $\ell_{b,p}(\cdot)$ evaluated at each approximate MLE $\widehat{\boldsymbol{\theta}}^{(s)}_{b,p}$. From these log-likelihoods we obtain an estimate of the expected difference between the exact and approximate log-likelihoods at the approximate MLE

$$\frac{1}{S}\sum_{s=1}^{S}\left|\ell(\widehat{\boldsymbol{\theta}}^{(s)}_{b,p}) - \ell_{b,p}(\widehat{\boldsymbol{\theta}}^{(s)}_{b,p})\right| \approx \mathbb{E}_{\boldsymbol{y}}\left|\ell\left(\widehat{\boldsymbol{\theta}}_{b,p}(\boldsymbol{y})\right) - \ell_{b,p}\left(\widehat{\boldsymbol{\theta}}_{b,p}(\boldsymbol{y})\right)\right| \quad (45)$$

The second metric is based on a Monte Carlo approximation to the expected norm of the score obtained by averaging the gradient of the exact log-likelihood at the approximate MLEs $\widehat{\boldsymbol{\theta}}^{(s)}_{b,p}$. These gradients should be close to zero because they would be exactly zero if the gradients were evaluated at the exact MLEs. Thus we consider the statistic

$$\frac{1}{S}\sum_{s=1}^{S}\left\|\nabla\ell\left(\widehat{\boldsymbol{\theta}}^{(s)}_{b,p}\right)\right\| \approx \mathbb{E}_{\boldsymbol{y}}\left\|\nabla\ell\left(\widehat{\boldsymbol{\theta}}_{b,p}(\boldsymbol{y})\right)\right\|. \quad (46)$$

When the this criteria is small, it indicates that the block full-scale approximation does not significantly influence parameter estimation in the case of a single sample of the process, because using the exact likelihood instead would have little impact on gradient-based optimization.

Figure 2 indicates a number of important trends which guide our choice of approximation parameters in the following application to real data. Primarily, the block size b tends to have a much larger impact on our metrics than the rank p. In particular, increasing the block size from 8 to 128 with a fixed rank reduces the expected gradient norm by an order of magnitude, while increasing the rank from 8 to 128 with a fixed block size yields virtually no improvement. This indicates that for fairly rough processes, preserving local interactions by using larger neighborhoods gives better tradeoffs in likelihood approximation than preserving long range interactions by increasing the rank or by careful selection of the Nyström landmark points. Thus it appears a modest rank is sufficient to couple local neighborhoods, and one should focus computational effort on choosing these neighborhoods to be as large as possible.

4.4 Sea surface temperature modeling application

To test our computational framework in the relevant setting of nonstationary modeling with many parameters, we study a large sea surface temperature anomaly dataset from the NOAA Coral Reef Watch database consisting of a $40° \times 40°$ domain in the central Pacific Ocean. These reanalysis data are de-meaned sea surface temperature measurements

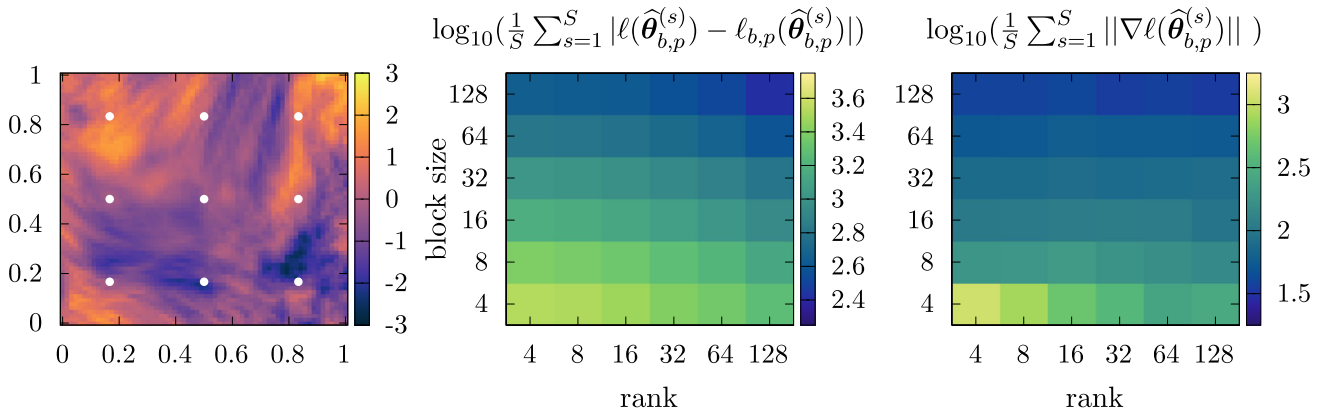

Fig. 2 A sample from the simulated data with RBF centers denoted with white dots (left). The difference in exact and approximated log-likelihoods given by Eq. 45 on a log scale for various Nyström ranks and block sizes (center). The expected score norm given in Eq. 46 on a log scale for the same combination of ranks and block sizes (right)

generated by interpolating polar-orbiting and geostationary satellite data from multiple sources to a 0.05° (∼ 5 km) grid using a Kalman filter-like approach (Maturi et al. 2017; Khellah et al. 2004). To study such a large spatial domain, we subsample this data to a 0.1° grid. We then use local averaging on NASA's MODIS Cloud Mask product (Ackerman and Frey 2015) to compute a holdout set on the 0.1° grid, which will serve as a testing set for evaluating predictions and uncertainties. This leaves 107,600 non-cloudy observations for maximum likelihood estimation. See Fig. 3. These cloud-masked data provide a realistic setting for interpolation in atmospheric science applications.

4.4.1 Fitting disjoint local neighborhoods

We start by partitioning our data into 64 disjoint subregions using a k-d tree, and we use the centroid of each of these regions as the center of an RBF within the full model, as shown in Fig. 3. To determine a suitable initial guess for our global model optimization and to compare against the current state of the art in which locally fitted parameters are plugged into the RBF expansion (39), we fit a local anisotropy $\boldsymbol{\Lambda}_i$ in each subregion using the trust region method described in Sect. 4.2. Since each subregion contains only 1681 observations, we can afford to use exact linear algebra and do not require any covariance matrix approximation.

4.4.2 Fitting the global model

Following the local fitting of anisotropy parameters, we optimize the full set of global model parameters $\boldsymbol{\theta}$. Here we consider the full data and thus require the block full-scale approximation and accompanying computations discussed in Sects. 2 and 3. We use the local neighborhoods around each RBF as the blocks in the approximation and use the k-d tree to select 72 Nyström points for the low-rank portion that are approximately equispaced over the spatial domain. We can therefore think of the global model as a combination of the exact local neighborhoods on which the local parameters were estimated, plus a low-rank term coupling these neighborhoods. Comparing the likelihood under a disjoint neighborhood model using the locally estimated anisotropy parameters (i.e. a block diagonal covariance) to the likelihood under the Paciorek-Schervish model with the block diagonal plus low-rank covariance approximation using the locally estimated anisotropy parameters as the Λ_i in Eq. (39), we observe a log-likelihood increase of 334,342 units. This large likelihood improvement indicates that observations in adjacent neighborhoods are in fact highly correlated and that the low-rank Nyström term captures some of this dependence.

Preparing for optimization of the global model parameters, we choose the width parameter c of the radial basis functions to be half of the minimum distance between RBF centers. This choice results in bases that are fairly localized, helping avoid identifiability problems between adjacent RBFs caused by more diffuse bases with larger c values. In principle, estimating the full parameter vector $\boldsymbol{\theta}$ allows one to also perform maximum likelihood for the parameter c, which is impossible with local estimation. We find, however, that if c is included as a parameter alongside the locally estimated parameters, the derivative in c is much larger than the derivatives in the anisotropy parameters in $\boldsymbol{\theta}$, and thus the solver moves in the c direction without materially improving the fit. If we instead fix a localized c and estimate $\boldsymbol{\theta}$ only, we find that the likelihood surface in c becomes uninformative because the basis width does not have a large impact on the model fit if the local parameters are well selected.

After making the block full-scale approximation and selecting the basis width, the key to fitting these 192 parameters in practice using the trust-region-based Fisher scoring

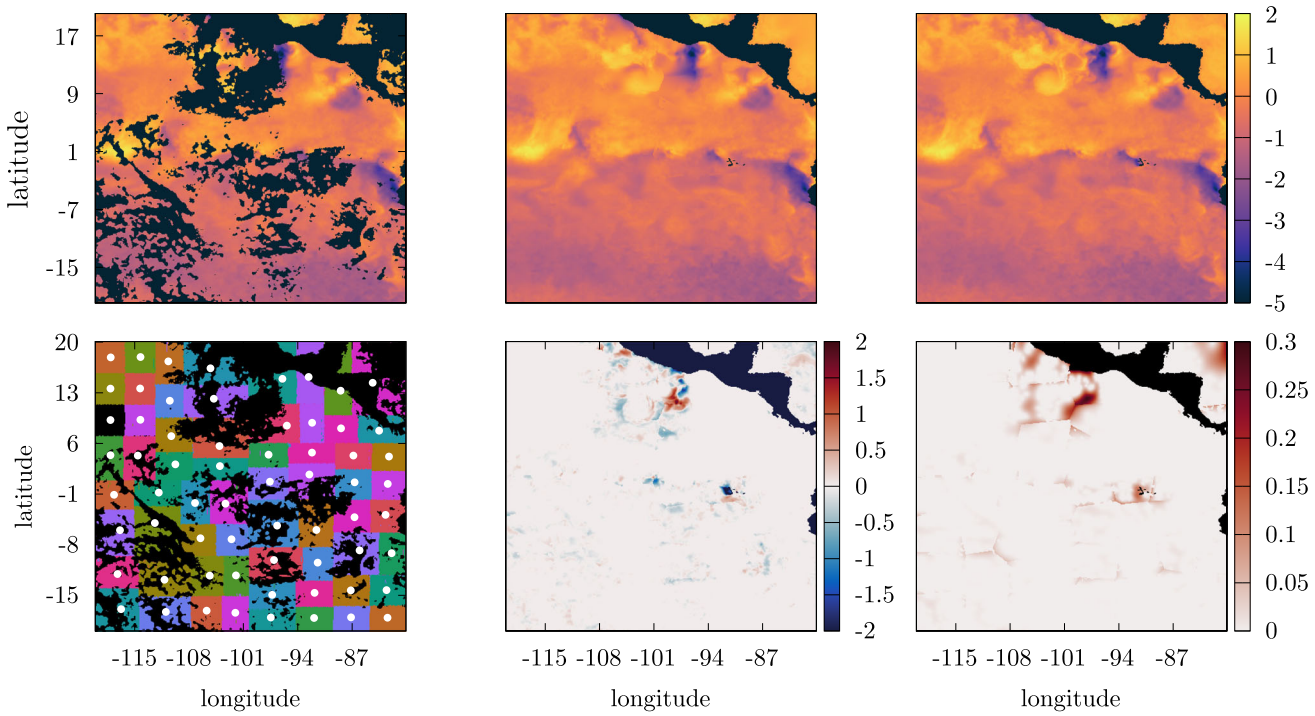

Fig. 3 Artificially cloud-masked data (top left), interpolant (top center), full data (top right), neighborhoods used for local fitting and approximation blocks with RBF centers as white dots (bottom left), interpolant error (bottom center), and interpolant standard deviations (bottom right)

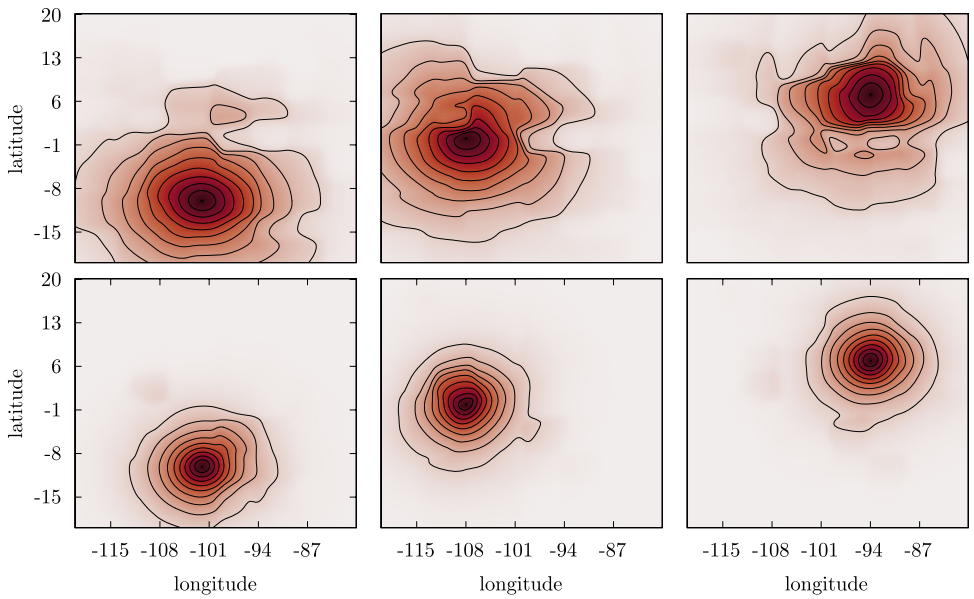

Fig. 4 Correlation functions at three points using parameters fit globally (top row) and locally (bottom row), with contours at levels 0.1, 0.2, ..., 0.9

algorithm described above lies in the use of the SAA methods discussed in Sect. 3.4 to compute the gradient and Fisher matrix in a single pass over the derivative matrices $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}$. The crucial observation is that the symmetrized estimators for both the gradient and Fisher entries (24, 25) require only inner products of terms of the form $\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell$ and $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell$. Thus, we can compute each $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}$ and matrix–vector products with $\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell$ efficiently and independently in parallel. We can then compute the necessary inner products for each entry in the gradient and Fisher matrix. We use 150 SAA vectors for this numerical experiment, which leads to a dramatic reduction both in the computational cost, because the inner products are less expensive than matrix-matrix products, and in memory, because the vectors $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell$ are significantly smaller than the full derivative matrices $\frac{\partial \tilde{\boldsymbol{\Sigma}}}{\partial \theta_j}$. While both exact

and SAA computations require $\mathcal{O}(n)$ time and storage, SAA provides a dramatic reduction in prefactors. On an Intel Xeon CPU E5-2650 @ 2.00 GHz machine, computing each Fisher matrix entry using the exact form takes about 40 s in our implementation after the matrices $\tilde{\boldsymbol{\Sigma}}^{-1}\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}$ have been computed, whereas each entry takes 0.05 s using SAA after the vectors $\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}\boldsymbol{W}^{-\top}\boldsymbol{u}_\ell$ have been computed. As a result, the full exact Fisher matrix requires about 206 h, whereas the SAA approximation requires only 15 min. In our implementation, parallelizing the construction of the derivatives of the covariance matrix $\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}$ on 16 cores leads to approximately 80 s for each of the 192 derivative matrices, resulting in a total of about 4 h. This easily dominates the computational cost at each iteration, far more demanding than computing the SAA gradient and Fisher matrix.

4.4.3 Comparison of local and global fits

Comparing the log-likelihood of the global Paciorek-Schervish model using the initial, locally fitted parameters with the log-likelihood of the same model using the final, global parameters given by our Fisher scoring trust-region algorithm, we observe an increase of 48,679 units. In total, parallelized over 16 cores, the time to optimize the local models in disjoint subregions was about 7 h, and the time to optimize the global model which stagnates after 16 iterations, was about 74 h. The substantial likelihood difference indicates that solving the high-dimensional optimization problem to fit all parameters simultaneously can produce a significant improvement in model fit when compared with purely local parameter estimation. The difference in covariance structure can be seen in Fig. 4, which shows the estimated correlation function obtained using local parameter estimates versus optimized global parameter estimates. In particular, we notice that the global model fitting captures larger east–west correlations above and below the vortices caused by equatorial currents and little correlation across these currents.

After maximum likelihood estimation is completed using the inexpensive SAA gradients and Fisher matrices, we can efficiently compute interpolants and the rank-structured conditional covariance matrix. Figure 3 shows the interpolation results along with standard errors. In addition, we compute the more expensive exact Fisher matrix at the MLE to evaluate parameter uncertainties. We see that the inverse Fisher matrix contains non-negligible terms away from the diagonal, indicating that some interaction exists between spatially proximal parameters. See Fig. 5, which shows the correlation matrix given by normalizing the inverse Fisher matrix to have unit diagonal entries. If one were to fit parameters only locally, the resulting Fisher matrix would be block diagonal with 3×3 blocks, which ignores these off-diagonal contributions (also shown in Fig. 5). Beyond the significantly improved likelihoods, this additional second-order information about parameter point estimates themselves is a material advantage of the global model formulation used here. Interpolation can be nearly optimal even with a completely misspecified covariance function (Stein 1999) and may not be significantly improved by a global model. We find negligible differences in the mean squared error of interpolants computed in disjoint local neighborhoods with stationary models and the interpolants shown in Fig. 3 that are computed with the global nonstationary model. However, uncertainties for estimated parameters *can* be significantly underestimated when global dependence is not accounted for. Especially in an RBF model such as the one presented here, the MLE and expected Fisher matrix from the global model may also serve as a reasonable approximation to a Bayesian posterior, although one needs to be careful when appealing to asymptotic results that depend on the consistency of parameter estimates when working with spatial data (Zhang 2004). The difference in parameter correlation structure can be seen in Fig. 6, which shows the correlation between the log of the upper left Cholesky entry $\log \ell_i^{(1,1)}$ (which serves as one of the nonstationary model parameters, see (43)) at various spatial locations for the global Paciorek-Schervish model and the disjoint local neighborhood model. Note the meaningful negative correlations between adjacent parameters in some regions when using the global model, which cannot be captured by disjoint local models.

To further identify where significant likelihood improvements have been made, we investigate quality of fit at the lower tail of the spectrum of the covariance matrix. If a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ has orthonormal eigenvectors $\boldsymbol{q}_j$ and corresponding positive eigenvalues λ_j, then a vector $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ can be represented as $\boldsymbol{z} \sim \sum_{j=1}^{n}\sqrt{\lambda_j}\varepsilon_j\boldsymbol{q}_j$, where $\varepsilon_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. By virtue of the orthogonality of the eigenvectors, it follows that under this model $\boldsymbol{q}_j^T\boldsymbol{z} \sim \mathcal{N}(0,\lambda_j)$, and $\boldsymbol{q}_j^T\boldsymbol{z}$ is independent of $\boldsymbol{q}_k^T\boldsymbol{z}$ for all $j \neq k$.

Because $\boldsymbol{\Sigma}$ is inverted in the quadratic form that appears in the likelihood, it is particularly revealing of log-likelihood improvements to inspect these inner products for eigenvectors corresponding to the smallest eigenvalues of $\boldsymbol{\Sigma}$. Figure 7 shows the quantity

$$Z_j = \left|\frac{\boldsymbol{q}_j(\boldsymbol{\theta})^T\boldsymbol{z}}{\sqrt{\lambda_j(\boldsymbol{\theta})}}\right|, \tag{47}$$

the absolute value of a standard Z-score, using the 100 smallest eigenvalues and corresponding eigenvectors for the model parameters $\boldsymbol{\theta}$ estimated locally in disjoint regions and simultaneously in the global model.

As can immediately be seen, even after taking into account small-sample variability, the Z-scores using the local plug-in parameters are excessively dispersed. This reflects a signif-

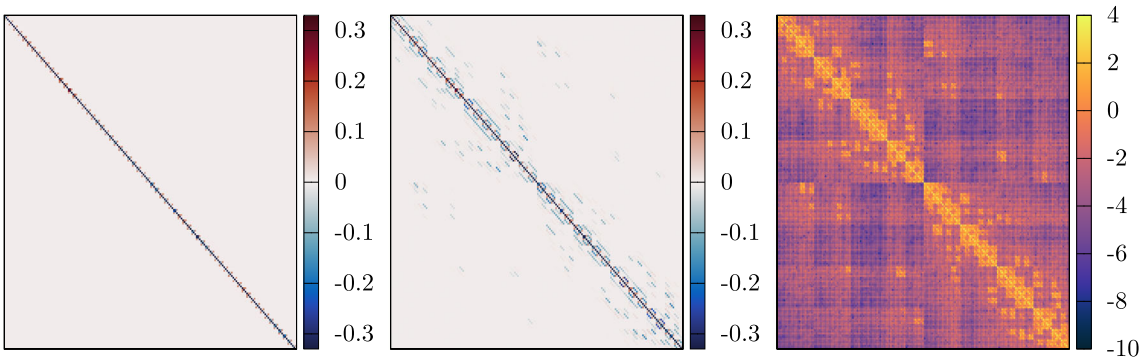

Fig. 5 Inverse Fisher matrix normalized to have unit diagonal entries for the disjoint local neighborhood model (left) and for the global Paciorek–Schervish model (center) shown with a reduced color range to emphasize off-diagonal elements, as well as the logs of the magnitudes of the entries of the latter Fisher matrix (right). Parameter blocks are ordered using a k-d tree on the RBF centers

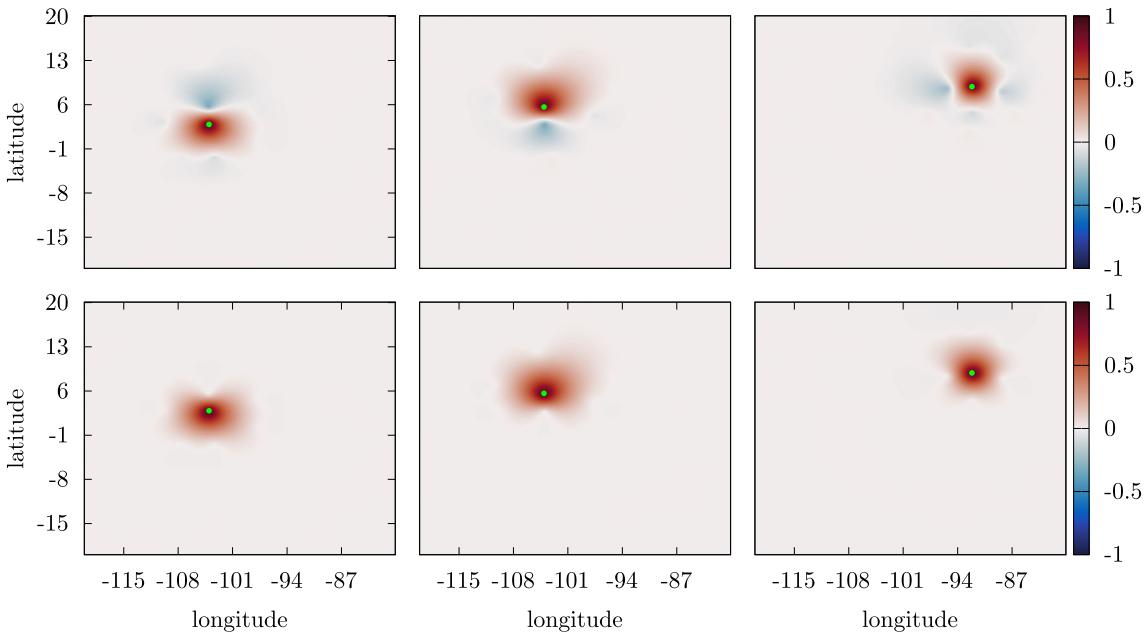

Fig. 6 Spatially indexed parameter correlation for $\log \ell_i^{(1,1)}$ with the value located at the green dot using the global Paciorek-Schervish model (top row) and the disjoint local neighborhood model (bottom row)

icant misfit, and the corresponding values for the globally optimized parameters much more closely resemble the standard half-normal density. Because the smallest eigenvectors for most standard covariance models generally reflect high-frequency energy and are comparable to some form of higher-order differencing, this indicates that the global optimization leads to the model capturing fundamentally local behavior better than what is achieved with plug-in local parameters. Since the model is necessarily somewhat misspecified, this behavior is likely explained at least in part as the global optimization sacrificing fit quality in some parts to make significant enough improvement in other parts such that the total likelihood is improved. Considering that almost every model used for serious applications with environmental data will be to some degree misspecified, we find this to be a valuable functionality that is not possible with plug-in local parameters.

Overall, we see that the combination of the covariance matrix approximation and second-order trust-region-based Fisher scoring algorithm presented here, which scale favorably with the data size and the number of parameters, can produce meaningfully different fits and capture inter-parameter dependence structure for many-parameter models. Therefore these methods represent an important consideration when modeling nonstationary data at scale.

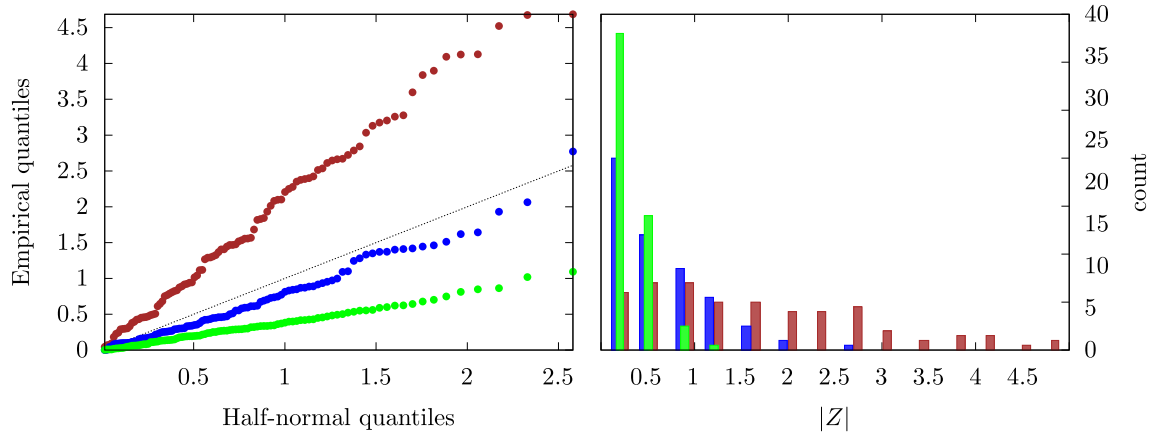

Fig. 7 Absolute values of standardized Z-scores for the eigenvector inner products given by Eq. (47) corresponding to the 100 smallest eigenvalues for the global model covariance matrix using plug-in local parameter estimates (red), globally optimized parameters (blue), and the nonstationary scale model discussed in Sect. 4.5 (green)

4.5 Comparison with a nonstationary scale

A nonstationary covariance model which is much easier to fit but still offers some degree of flexibility can be generated by allowing only the scale parameter to vary in space. This category of covariance function was used, for example, in Guinness (2021), and demonstrated a notable improvement in the terminal likelihood in their application. In this section, we briefly compare the primary model here, which uses a spatially varying anisotropy, with this class of variable scale covariance functions.

After experimentation, for this particular dataset an RBF-based interpolation scheme between parameterized locations yielded higher likelihoods than the orthogonal basis function approach of Guinness (2021), so we use the spatially varying scale function

$$\sigma(\boldsymbol{x}) = \left(\sum_{i=1}^{S} w_i(\boldsymbol{x})\right)^{-1} \sum_{i=1}^{S} w_i(\boldsymbol{x})\sigma_i, \tag{48}$$

where $w_i(\boldsymbol{x}) = \exp\left(-\sqrt{(\boldsymbol{x}-\boldsymbol{x}_j)^T \boldsymbol{\Gamma}^{-1}(\boldsymbol{x}-\boldsymbol{x}_j)}\right)$ is a simple weighting function, σ_i is the directly parameterized scale at location $\boldsymbol{x}_i$, and $\boldsymbol{\Gamma}$ is a diagonal matrix chosen to compensate for the rectangularity of the spatial domain. The full kernel is then given by

$$k_{\text{scale}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma(\boldsymbol{x}_i)\sigma(\boldsymbol{x}_j)\mathcal{M}_1 \left(\sqrt{(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j)}\right), \tag{49}$$

where $\boldsymbol{\Lambda} = \boldsymbol{L}\boldsymbol{L}^\top$ is a globally stationary anisotropy matrix parameterized via its Cholesky factor

$$\boldsymbol{L} = \begin{bmatrix} \ell^{(1,1)} & 0 \\ \ell^{(2,1)} & \ell^{(2,2)} \end{bmatrix} \tag{50}$$

and $\mathcal{M}_1$ is the Matérn covariance function with its smoothness fixed to $\nu = 1$. Following Guinness (2021), we picked 10 spatial anchor locations to cover the domain, making manual corrections to be sure that every anchor location had a good amount of non-missing data around it. Thus our full parameter vector for this model is $\boldsymbol{\theta} = [\log\sigma_1, ..., \log\sigma_{10}, \log\ell^{(1,1)}, \ell^{(2,1)}, \log\ell^{(2,2)}] \in \mathbb{R}^{13}$.

Table 8 shows the results of estimation for this model using the same covariance matrix approximation scheme, block size, and off-diagonal rank as the nonstationary scale model above. As can be seen, in terms of likelihoods, our model with varying anisotropy does significantly better, even if one compensates for the significant difference in the number of parameters. The prediction metrics are much closer, although still slightly prefer our model. The degree of similarity between these two models in the prediction setting is interesting, although with data this dense and with this level of dependence it is not entirely surprising.

In addition to these comparisons, we also include in Fig. 7 the eigen-Z statistics obtained from the tail of the spectrum of the fitted covariance matrix under this model (results in green). Interestingly, the direction of the misfit for this model is that the variance of those linear functions of the data is over-estimated by the model. Inspecting those eigenvectors shows that they resemble high-order finite differences, primarily across latitude, in the upper northwest portion of the domain, and are often near the large primary cloud mask. This is an interesting region because the data is particularly

Model	$\ell(\hat{\boldsymbol{\theta}})$	RMSE	max. abs. error
Nonstationary anisotropy (global)	**0.0**	**0.1281**	1.790
Nonstationary anisotropy (local)	−48,679	0.1314	**1.767**
Nonstationary scale	−32,194	0.1284	1.768

Fig. 8 A comparison of estimation and prediction metrics using our nonstationary anisotropy model with both locally and globally estimated parameters, and model using a nonstationary scale parameter. Log-likelihoods at the MLE are normalized to zero by the highest value, and the prediction regions used in the RMSE and maximum error columns are the cloud-masked regions of the domain using the NOAA data product as the ground truth

smooth, but there is a lot of structure being obscured by those clouds. Considering that that portion of the domain was also a hotspot for serious prediction error in our nonstationary anisotropy model, a possible explanation for this behavior in the eigen-Z scores is that these estimated parameters do indicate more variability in that area than just the available data would suggest, and considering that underestimating variability punishes a likelihood much worse than over-estimating, a model that is conservative about the variance of those differences would naturally do better.

5 Discussion

In this work we present a complete set of linear cost computations for second-order maximum likelihood estimation of Gaussian process parameters using the block full-scale approximation with application to fitting many-parameter nonstationary models. In particular, we give exact algorithms for computing the Gaussian log-likelihood and its gradient, Fisher information matrix, and Hessian, as well as methods for highly efficient stochastic approximation. The ability to compute derivatives of the log-likelihood is crucial especially in fitting expressive covariance models with many parameters, in which solvers must navigate a complex, high-dimensional, nonconvex objective landscape. We demonstrate using a large sea surface temperature dataset that our methods facilitate parameter estimation for nonstationary models with a very large number of parameters. Further, we demonstrate the value of complex global parameterizations by the significantly improved likelihood and additional inferred covariance structure between parameters, motivating the need for methods that are designed to be scalable with respect to both data size and parameter size.

The block diagonal plus low-rank approximation we use is well known in the literature as the partially independent conditional (Snelson and Ghahramani 2007) or block full-scale approximation (Sang et al. 2011). It is also a special case of the Vecchia approximation (Katzfuss and Guinness 2021) and is a two-level case of the more general hierarchical models of both Katzfuss (Katzfuss 2017; Katzfuss and Gong 2020) and Chen (Chen and Stein 2021). While these more sophisticated approaches can achieve more accurate approximations to the log-likelihood (Katzfuss and Gong 2020; Chen and Stein 2021), they provide no efficient methods for computing its derivatives, making high-dimensional parameter estimation extremely challenging. Our principal contributions are to demonstrate that the block full-scale approximation yields efficient computations with and without a nugget using Schur complements in a permuted covariance matrix and to apply these computations to high-dimensional parameter estimation problems. The utility and efficiency of computations with the block diagonal plus low-rank structure are due to the fact that this structure is closed with respect to matrix-matrix addition, multiplication, and inversion. Therefore, by making a single algebraic approximation to the covariance matrix, we obtain rank-structured derivatives matrices, symmetric factors, and conditional covariance matrices that can be assembled using only $\mathcal{O}(n)$ computation and storage.

Although we find that this covariance matrix approximation in conjunction with the SAA gradient and Fisher matrix methods makes parameter estimation possible for hundreds of parameters, it has limitations in some circumstances. With respect to the covariance approximation, Katzfuss and Gong (2020) show that the conditional mean can suffer from discontinuities at block boundaries, which are noticeable upon close inspection of the interpolant in Fig. 3. Although these can in theory be alleviated by tapering at block boundaries with a sufficiently smooth compactly supported function, this comes at the cost of approximation accuracy. Regarding SAA and parameter dimension, the full Fisher matrix contains $\mathcal{O}(m^2)$ many entries, where m is the number of parameters. Due to the incredible efficiency of computing SAA inner products, this is not a computational bottleneck in the application shown above. However, if one were to need thousands of parameters with hundreds of thousands of observations, or if exact Fisher entries were required for extremely high-precision optimization, this quadratic scaling in parameter dimension may become prohibitive. In this regime, some form of structured quasi-Newton that approximates the Fisher matrix using fewer entries may become necessary. Additionally, we note that the cost of evaluating the basis function expansion (39) to compute the covariance itself has cost $\mathcal{O}(m)$, so computing the m derivative matrices $\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}$ requires $\mathcal{O}(m^2 n)$ effort, which explains why forming these matrices is the bottleneck in our application. This could potentially be alleviated through the use of compactly supported basis functions as in Huang et al. (2021), although these authors use purely local parameter estimation in place of the global model estimation we have developed here.

Finally, we note that while this paper is exclusively concerned with the maximum likelihood estimation of direct

Gaussian process models, scalable approximations that are also designed to accommodate models with many parameters may be useful in other domains and modeling paradigms. Hamiltonian Monte Carlo (Neal 2011) methods benefit from derivative information, and so applying this methodology that prioritizes derivatives may be useful. Similarly, the application of latent Gaussian processes in hierarchical models that involve sampling may also benefit from such approximations, although as always, one should use caution when applying kernel matrix approximation techniques in higher dimensions.

Acknowledgements This material was based upon work supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) under Contract DE-AC02-06CH11347. We would also like to thank the anonymous referees for many helpful comments which led to significant improvements to the manuscript.

Appendix

Computing the Hessian

To employ second-order Newton solvers instead of Fisher scoring to compute the MLE, one must compute the Hessian, whose entries are given by

$$\begin{aligned}\left[-\nabla^2\ell(\boldsymbol{\theta})\right]_{jk} = -\mathscr{I}_{jk} &+ \frac{1}{2}\operatorname{tr}\left[\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial^2\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j\partial\theta_k}\right)\right] \\ &+ \boldsymbol{y}^\top\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}\right)\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_k}\right)\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y} \\ &- \frac{1}{2}\boldsymbol{y}^\top\tilde{\boldsymbol{\Sigma}}^{-1}\left(\frac{\partial^2\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j\partial\theta_k}\right)\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{y}. \end{aligned} \tag{51}$$

We can again apply basic matrix differentiation rules to equation (19) to obtain the second derivatives of the Nyström approximation, where the second derivative of the rank-p Nyström approximation

$$\begin{aligned}&\frac{\partial^2}{\partial\theta_j\partial\theta_k}\left(\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top\right) \\ &= \left(\frac{\partial^2\boldsymbol{\Sigma}_{QP}}{\partial\theta_j\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top \\ &\quad - \left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top \\ &\quad + \left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{QP}^\top}{\partial\theta_k}\right) \\ &\quad - \left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top \\ &\quad + \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top \\ &\quad - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial^2\boldsymbol{\Sigma}_{PP}}{\partial\theta_j\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top \\ &\quad + \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top \\ &\quad - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_j}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{QP}^\top}{\partial\theta_k}\right) \\ &\quad + \left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\right)^\top \\ &\quad - \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{PP}}{\partial\theta_k}\right)\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial\boldsymbol{\Sigma}_{QP}}{\partial\theta_j}\right)^\top \\ &\quad + \boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\left(\frac{\partial^2\boldsymbol{\Sigma}_{QP}}{\partial\theta_j\partial\theta_k}\right)^\top \end{aligned} \tag{52}$$

has rank at most $4p$. The second derivative of the approximate covariance matrix $\tilde{\boldsymbol{\Sigma}}$ is then given by

$$\frac{\partial^2\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j\partial\theta_k} = \boldsymbol{\Pi}^\top\begin{bmatrix}\frac{\partial^2}{\partial\theta_j\partial\theta_k}\left(\boldsymbol{\Sigma}_{QP}\boldsymbol{\Sigma}_{PP}^{-1}\boldsymbol{\Sigma}_{QP}^\top\right) + \frac{\partial^2\boldsymbol{D}}{\partial\theta_j\partial\theta_k} & \frac{\partial^2\boldsymbol{\Sigma}_{QP}}{\partial\theta_j\partial\theta_k} \\ \frac{\partial^2\boldsymbol{\Sigma}_{QP}^\top}{\partial\theta_j\partial\theta_k} & \frac{\partial^2\boldsymbol{\Sigma}_{PP}}{\partial\theta_j\partial\theta_k}\end{bmatrix}\boldsymbol{\Pi}, \tag{53}$$

which follows the same permuted block diagonal plus low-rank structure as $\tilde{\boldsymbol{\Sigma}}$ and $\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_j}$, now with rank at most $4p$ in the low-rank portion of the upper left block. Thus the trace term in each Hessian entry can be computed using equation (21) exactly as was done for the analogous term in the gradient. This results in a linear complexity algorithm for computing entries of the Hessian.

Nonstationary parameters for simulated data

For our approximation accuracy study in Sect. 4.3, we generate data from the Paciorek-Schervish model (6) using a continuously varying local anisotropy matrix $\boldsymbol{\Lambda}(\boldsymbol{x})$ given by

$$\begin{aligned}\boldsymbol{\Lambda}(\boldsymbol{x}) = &\begin{bmatrix}\cos(\theta(\boldsymbol{x})) & -\sin(\theta(\boldsymbol{x}))\\ \sin(\theta(\boldsymbol{x})) & \cos(\theta(\boldsymbol{x}))\end{bmatrix}\begin{bmatrix}\lambda_1(\boldsymbol{x}) & \\ & \lambda_2(\boldsymbol{x})\end{bmatrix} \\ &\times\begin{bmatrix}\cos(\theta(\boldsymbol{x})) & -\sin(\theta(\boldsymbol{x}))\\ \sin(\theta(\boldsymbol{x})) & \cos(\theta(\boldsymbol{x}))\end{bmatrix}^T, \end{aligned} \tag{54}$$

$$\theta(\boldsymbol{x}) = \operatorname{acos}\left(\min\left(1, \frac{\boldsymbol{x}^T\boldsymbol{x}_\theta^*}{||\boldsymbol{x}||||\boldsymbol{x}_\theta^*||}\right)\right), \tag{55}$$

$$\lambda_1(\boldsymbol{x}) = \exp(2\cos(4||\boldsymbol{x}-\boldsymbol{x}_1^*||) - 3), \quad \text{and} \tag{56}$$

$$\lambda_2(\boldsymbol{x}) = \exp(2\cos(4||\boldsymbol{x}-\boldsymbol{x}_2^*||) - 3), \tag{57}$$

where $\boldsymbol{x}_\theta^* = [1.75, 2.25]$, $\boldsymbol{x}_1^* = [0.75, 0.5]$, and $\boldsymbol{x}_2^* = [0.3, 0.2]$. These functions and values were chosen in an ad hoc way simply to provide sample paths with interesting features that resemble real data.

References

Ackerman, S., Frey, R.: Modis Atmosphere l2 Cloud Mask Product. NASA MODIS Adaptive Processing system. Goddard Space Flight Center (2015)

Ambikasaran, S., O'Neil, M., Singh, K.R.: Fast symmetric factorization of hierarchical matrices with applications (2014). arXiv preprint arXiv:1405.0223

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)

Anderes, E.B., Stein, M.L.: Local likelihood estimation for nonstationary random fields. J. Multivar. Anal. **102**(3), 506–520 (2011)

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

Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**(9), 509–517 (1975)

Börm, S., Garcke, J.: Approximating Gaussian processes with $\mathcal{H}^2$-matrices. In: European Conference on Machine Learning, pp. 42–53. Springer (2007)

Chen, J., Stein, M.L.: Linear-cost covariance functions for Gaussian random fields. J. Am. Stat. Assoc. **118**, 1–18 (2021)

Chen, J., Avron, H., Sindhwani, V.: Hierarchically compositional kernels for scalable nonparametric learning. J. Mach. Learn. Res. **18**(1), 2214–2255 (2017)

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

Eidsvik, J., Finley, A.O., Banerjee, S., Rue, H.: Approximate Bayesian inference for large spatial datasets using predictive process models. Comput. Stat. Data Anal. **56**(6), 1362–1380 (2012)

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

Geoga, C.J., Marin, O., Schanen, M., Stein, M.L.: Fitting Matérn smoothness parameters using automatic differentiation (2022). arXiv preprint arXiv:2201.00090

Geoga, C.J., Anitescu, M., Stein, M.L.: Scalable Gaussian process computations using hierarchical matrices. J. Comput. Graph. Stat. **29**, 1–11 (2019)

Guinness, J.: Gaussian process learning via fisher scoring of Vecchia's approximation. Stat. Comput. **31**(3), 1–8 (2021)

Huang, H., Blake, L.R., Katzfuss, M., Hammerling, D.M.: Nonstationary spatial modeling of massive global satellite data (2021). arXiv preprint arXiv:2111.13428

Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Commun. Stat. Simul. Comput. **18**(3), 1059–1076 (1989)

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

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

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

Katzfuss, M., Guinness, J.: A general framework for Vecchia approximations of gaussian processes. Stat. Sci. **36**(1), 124–141 (2021)

Khellah, F., Fieguth, P., Murray, M.J., Allen, M.: Statistical processing of large image sequences. IEEE Trans. Image Process. **14**(1), 80–93 (2004)

Li, Y., Sun, Y.: Efficient estimation of nonstationary spatial covariance functions with application to high-resolution climate model emulation. Stat. Sin. **29**(3), 1209–1231 (2019)

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

Litvinenko, A., Sun, Y., Genton, M.G., Keyes, D.E.: Likelihood approximation with hierarchical matrices for large spatial datasets. Comput. Stat. Data Anal. **137**, 115–132 (2019)

Maturi, E., Harris, A., Mittaz, J., Sapper, J., Wick, G., Zhu, X., Dash, P., Koner, P.: A new high-resolution sea surface temperature blended analysis. Bull. Am. Meteor. Soc. **98**(5), 1015–1026 (2017)

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)

Neal, R.M., et al.: MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, vol. 2(11), p. 2 (2011)

Paciorek, C.J., Schervish, M.J.: Spatial modelling using a new class of nonstationary covariance functions. Environ. Off. J. Int. Environ. Soc. **17**(5), 483–506 (2006)

Risser, M.D., Calder, C.A.: Local likelihood estimation for covariance functions with spatially-varying parameters: the convospat package for r. arXiv preprint (2015). arXiv:1507.08613

Sang, H., Huang, J.Z.: A full scale approximation of covariance functions for large spatial data sets. J. Roy. Stat. Soc. Ser. B (Stat. Methodol.) **74**(1), 111–132 (2012)

Sang, H., Jun, M., Huang, J.Z.: Covariance approximation for large multivariate spatial data sets with an application to multiple climate model errors. Ann. Appl. Stat. **20**, 2519–2548 (2011)

Snelson, E., Ghahramani, Z.: Local and global sparse Gaussian process approximations. Artif. Intell. Stat. 524–531 (2007)

Solin, A., Särkkä, S.: Hilbert space methods for reduced-rank Gaussian process regression. Stat. Comput. **30**(2), 419–446 (2020)

Stein, M.L., Chen, J., Anitescu, M.: Stochastic approximation of score functions for Gaussian processes. Ann. Appl. Stat. 1162–1191 (2013)

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

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

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

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

Wright, S., Nocedal, J.: Numerical optimization. Science **35**(67–68), 7 (1999)

Zhang, H.: Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. J. Am. Stat. Assoc. **99**(465), 250–261 (2004)