Journal of Data Science 20 (4), 512–532
October 2022
DOI: 10.6339/22-JDS1075
Statistical Data Science

Geostatistics for Large Datasets on Riemannian Manifolds: A Matrix-Free Approach

Mike Pereira[1,2,*], Nicolas Desassis[2], and Denis Allard[3]

[1]*Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, 412 96 Gothenburg, Sweden*
[2]*Mines Paris, PSL University, Centre for geosciences and geoengineering, 77300 Fontainebleau, France*
[3]*Biostatistics and Spatial Processes (BioSP), INRAE, 84914 Avignon, France*

Abstract

Large or very large spatial (and spatio-temporal) datasets have become common place in many environmental and climate studies. These data are often collected in non-Euclidean spaces (such as the planet Earth) and they often present nonstationary anisotropies. This paper proposes a generic approach to model Gaussian Random Fields (GRFs) on compact Riemannian manifolds that bridges the gap between existing works on nonstationary GRFs and random fields on manifolds. This approach can be applied to any smooth compact manifolds, and in particular to any compact surface. By defining a Riemannian metric that accounts for the preferential directions of correlation, our approach yields an interpretation of the nonstationary geometric anisotropies as resulting from local deformations of the domain. We provide scalable algorithms for the estimation of the parameters and for optimal prediction by kriging and simulation able to tackle very large grids. Stationary and nonstationary illustrations are provided.

Keywords *anisotropy; finite elements; Gaussian process; Laplace-Beltrami operator; nonstationarity*

1 Introduction

Large or very large spatial (and spatio-temporal) datasets have become common place in many environmental and climate studies. Various approaches have been proposed by the statistical community to tackle the "big N" challenge in ways that properly acknowledge the spatial and spatio-temporal dependence structures usually observed in these datasets. Recently, several competitions have been organized to compare methods and algorithms in this context (Heaton et al., 2019, Huang et al., 2021) which provide an excellent overview of state-of-the-art methods for analyzing large spatial datasets. Most algorithms rely on approximation methods of Gaussian Processes, also known as Gaussian Random Fields (GRFs). In their conclusion, the organizers of the first competition note that "spatial data may exhibit anisotropy, nonstationarity, large and small range spatial dependence as well". Despite its long history, modeling nonstationary spatial datasets remains a challenge. Sampson and Guttorp (1992) proposed a space deformation approach further developed for instance in Perrin and Senoussi (2000) and Fouedjio et al. (2015). The main difficulty with this approach is to estimate a valid, global, deformation of the domain, which in practice is not guaranteed to exist. Paciorek and Schervish (2006) (see also

*Corresponding author. Email: mike.pereira@minesparis.psl.eu.

Received August 2, 2022; Accepted October 17, 2022

Fouedjio et al. (2016)) introduced a class of nonstationary covariance functions based on the kernel convolution approach of Higdon et al. (1999), later generalized in Porcu et al. (2009). Our approach rather builds on the so-called "SPDE approach" introduced in the seminal work of Lindgren et al. (2011), which relates GRFs characterized with Matérn covariance functions to stationary solutions of a specific Stochastic Partial Differential Equation (SPDE), see also Carrizo Vergara et al. (2022) for models beyond Whittle–Matérn GRFs. This approach can be extended to the nonstationary case by allowing spatially varying coefficients of the SPDE, see for instance Fuglstad et al. (2015a;b) among other possible references. Another challenge often faced when analyzing environmental or climate data is to work on non-Euclidean domains. In particular, methods for analyzing data on spheres have received a lot of attention, see Marinucci and Peccati (2011), Jeong et al. (2017) and Porcu et al. (2021) for recent reviews and Rayner et al. (2020) for a recent application using the SPDE approach dealing with extremely large datasets. Methods developed for spheres depend usually on specific properties, such as expansion into the spherical harmonics (Emery and Porcu, 2019, Lang and Schwab, 2015, Lantuéjoul et al., 2019) or the use of arc distances to define valid covariance models (Gneiting, 2013, Huang et al., 2011).

This paper aims at bridging the gap between existing works on nonstationary GRFs and random fields on manifolds. Specifically, we propose a generic approach to model GRFs on compact Riemannian manifolds and we provide scalable algorithms for their optimal prediction by kriging and simulation. Our approach is based on two main ingredients. First, random fields are defined through expansions in the eigenfunctions of the Laplace–Beltrami operator on the Riemannian manifold which are, in some cases, solutions to some particular SPDEs. Then, we build finite element approximations of these GRFs. This construction allows to perform optimal prediction, simulation (including conditional) and estimation of the parameters using scalable algorithms.

For this purpose, we define a Riemannian metric that accounts for the preferential directions of correlation of the nonstationary GRF. This method yields an interpretation of the "local anisotropies" as resulting from "local" deformations of the domain, in striking contrast to both the space deformation and the kernel convolution approaches. The resulting fields can be seen as a direct generalization of the construction for nonstationary random fields proposed in Fuglstad et al. (2015a).

Our approach can be applied to any smooth compact manifold, and in particular to any compact surface or hypersurface. It shares clear similarities with the work of Borovitskiy et al. (2020), but in contrast, it is not restricted to Whittle–Matérn fields since in our approach the GRF is characterized by its spectral density, whose inverse is restricted to belong to the family of positive polynomials. Besides, our approach does not rely on the explicit computation of the eigenfunctions and eigenvalues of the Laplace–Beltrami operator, and can be seen as providing a theoretical motivation to the method developed in Borovitskiy et al. (2021) to deal with Gaussian processes on graphs.

The flexibility of our approach does not result in increased computational costs. Indeed, we show how prediction and conditional simulations can be performed through a so-called "matrix-free" approach. This approach, unlike classical geostatistical algorithms, does not require to build and store possibly large covariance matrices, but instead relies only on products between some sparse matrices and vectors. This in turn ensures the scalability of this method, thus paving the way to efficient nonstationary geostatistics for large datasets. We illustrate our approach with 2D and 3D synthetic examples and grids with more that 10^7 nodes for the 3D cases.

The organization of the paper is the following. The GRF model is presented in Section 2,

along with its finite element approximation and covariance function. Kriging and simulation algorithms are provided in Section 3. Section 4 shows how the parameters can be estimated using maximum likelihood. All methods are summarized as Algorithms. Stationary and nonstationary illustrations are then provided in Section 5. We conclude with some final words in Section 6. Proofs and technical details are deferred to the Appendix. Throughout this paper, vectors and matrices will be denoted in bold fonts. The superscript ^{T} denotes the transpose operation on matrices or vectors. Diag is the operator that transforms a vector of length n into an $n \times n$ matrix whose diagonal elements are those of the vector and whose off-diagonal elements are 0. $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. $|\boldsymbol{A}|$ is the determinant of the square matrix $\boldsymbol{A}$ and $\|\cdot\|$ denotes the Euclidean norm.

2 Random Fields on Riemannian Manifolds

2.1 Definition and Finite Element Approximation

A generic approach to define and characterize GRFs on Riemannian manifolds has been proposed in Borovitskiy et al. (2020) and in Lang and Pereira (2021) and is now briefly recalled. Let $\mathcal{D}$ be a smooth compact manifold of dimension d equipped with a Riemannian metric g, and let $f : \mathbb{R}_+ \to \mathbb{R}_+$ be a function such that for some $\beta > d/2$, $|\lambda^\beta f(\lambda)|$ is bounded as $\lambda \to +\infty$. A centered GRF $\mathcal{Z}$ is constructed on the resulting Riemannian manifold $(\mathcal{D}, g)$ through the expansion

$$\mathcal{Z} = \sum_{k\in\mathbb{N}} f^{1/2}(\lambda_k) w_k e_k, \tag{1}$$

where $f^{1/2} : \mathbb{R}_+ \to \mathbb{R}$ is a function such that $\left(f^{1/2}\right)^2 = f$ on $\mathbb{R}_+$, $\{w_k\}_{k\in\mathbb{N}}$ is a sequence of independent standard Gaussian variables, $\{\lambda_k\}_{k\in\mathbb{N}}$ denote the set of eigenvalues of the Laplace–Beltrami operator $-\Delta$ on $(\mathcal{D}, g)$, and $\{e_k\}_{1\leqslant k\leqslant\infty}$ denote the associated eigenfunctions. In order to get a feeling of what Eq. (1) represents, one could recall that on $\mathbb{R}^d$ the eigenfunctions of the Laplacian are all the members of the uncountable family of functions $\{e^{-i\langle\boldsymbol{\omega},\boldsymbol{x}\rangle} : \boldsymbol{\omega} \in \mathbb{R}^d\}$. In this case, the Whittle–Matérn random fields are solution to the SPDE $(\kappa^2 - \Delta)^{\alpha/2}\mathcal{Z} = \mathcal{W}$ where $\mathcal{W}$ is the white noise process on $\mathbb{R}^d$ and the associated spectral density is $f^{1/2}(\|\boldsymbol{w}\|) = (\kappa^2 + \|\boldsymbol{\omega}\|)^{-\alpha/2}$.

Going back to our definition of GRFs on compact manifolds, the eigenvalues and eigenfunctions of the Laplace–Beltrami operator are not known in general. A first approach, proposed by Borovitskiy et al. (2020), consists of computing them numerically by solving (approximately) the corresponding eigenvalue problems. Instead, we approximate $\mathcal{Z}$ using a finite element approach as in Lindgren et al. (2011) and Lang and Pereira (2021), as this method will naturally yield scalable algorithms for prediction and sampling tasks. First, the manifold $\mathcal{D}$ is triangulated using n nodes $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n \in \mathcal{D}$ and a family of compact support approximation functions $\{\psi_i\}_{1\leqslant i\leqslant n}$ is defined over $\mathcal{D}$, where each ψ_i is the piecewise linear function equal to 1 at the node $\boldsymbol{s}_i$ and 0 at all the other nodes. Then, $\mathcal{Z}$ is approximated by a linear combination Z defined as

$$Z = \sum_{i=1}^{n} z_i \psi_i, \tag{2}$$

where for any $i \in \{1, \ldots, n\}$, $z_i = Z(\boldsymbol{s}_i)$ is the weight associated with the basis function ψ_i.

The weights $\boldsymbol{Z} = (z_1, \ldots, z_n)^T$ are chosen so that Z can also be written using the same expansion as the one defining the original field $\mathcal{Z}$ in (1), but replacing now the eigenfunctions $\{e_k\}_{k\in\mathbb{N}}$ and eigenvalues $\{\lambda_k\}_{k\in\mathbb{N}}$ of $-\Delta$ by those of its Galerkin approximation (see Appendix D

for more details). This particular choice yields an explicit formula to compute these weights, see Proposition 2.1 below. We first introduce $\boldsymbol{C}$ and $\boldsymbol{F}$, the mass and stiffness matrices respectively defined by

$$[\boldsymbol{C}]_{ij} = (\psi_i, \psi_j), \quad [\boldsymbol{F}]_{ij} = (\nabla\psi_i, \nabla\psi_j), \quad 1 \leqslant i, j \leqslant n, \tag{3}$$

where $(\cdot, \cdot)$ denotes the L^2 inner product on the Riemannian manifold (see Appendix C). Note that since each basis function ψ_i $(1 \leqslant i \leqslant n)$ is zero for every node of the triangulation except one, the resulting matrices $\boldsymbol{C}$ and $\boldsymbol{F}$ are sparse. Let then $\sqrt{\boldsymbol{C}} \in \mathbb{R}^{n\times n}$ be a matrix such that $\sqrt{\boldsymbol{C}}(\sqrt{\boldsymbol{C}})^T = \boldsymbol{C}$, and let $\boldsymbol{S}$ be the matrix defined by

$$\boldsymbol{S} = (\sqrt{\boldsymbol{C}})^{-1}\boldsymbol{F}(\sqrt{\boldsymbol{C}})^{-T}. \tag{4}$$

Note that since $\boldsymbol{S}$ is real, symmetric and positive semi-definite, it is diagonalizable with non-negative eigenvalues. Therefore it can be written as

$$\boldsymbol{S} = \boldsymbol{V}\mathrm{Diag}(\lambda_1, \ldots, \lambda_n)\boldsymbol{V}^T,$$

where $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $\boldsymbol{S}$ and $\boldsymbol{V}$ is an orthogonal matrix whose columns are eigenvectors of $\boldsymbol{S}$.

Proposition 2.1. *Let $\boldsymbol{Z}$ be the vector of weights as in Eq. (2). Then, the vector $\boldsymbol{Z}$ is a centered Gaussian vector with covariance matrix $\boldsymbol{\Sigma}$ given by*

$$\boldsymbol{\Sigma} = (\sqrt{\boldsymbol{C}})^{-T} f(\boldsymbol{S})(\sqrt{\boldsymbol{C}})^{-1}, \tag{5}$$

where $f(\boldsymbol{S})$ is a matrix function, defined from the eigendecomposition of $\boldsymbol{S}$ as

$$f(\boldsymbol{S}) = \boldsymbol{V}\mathrm{Diag}\big(f(\lambda_1), \ldots, f(\lambda_n)\big)\boldsymbol{V}^T.$$

This result is shown in Pereira (2019) and Lang and Pereira (2021). In the latter reference, a convergence result of the approximation of $\mathcal{Z}$ by Z as the mesh size of the triangulation decreases is provided, thus further justifying this approach.

The matrix square-root $\sqrt{\boldsymbol{C}}$ can be computed using matrix functions or through a Cholesky decomposition. In practice however, $\sqrt{\boldsymbol{C}}$ is replaced by the so-called mass lumping approximation defined as the diagonal matrix with entries:

$$[\sqrt{\boldsymbol{C}}]_{ii} = \sqrt{(\psi_i, 1)}, \quad 1 \leqslant i \leqslant n. \tag{6}$$

To ease the notations, but at the cost of a slight abuse of notation, the mass lumping approximation will also be denoted $\sqrt{\boldsymbol{C}}$ in the remainder of this text. As shown in Lindgren et al. (2011), this approximation comes with negligible effect on the covariance of the resulting random field. It allows to readily have access to the inverse of the square-root matrix $\sqrt{\boldsymbol{C}}$ and yields

$$[\boldsymbol{S}]_{ij} = \frac{(\nabla\psi_i, \nabla\psi_j)}{\sqrt{(\psi_i, 1)}\sqrt{(\psi_j, 1)}}, \quad 1 \leqslant i, j \leqslant n, \tag{7}$$

which ensures that the matrix $\boldsymbol{S}$ is also sparse.

The expression of the covariance matrix $\boldsymbol{\Sigma}$ in Eq. (5) involves a matrix function defined from the eigendecomposition of $\boldsymbol{S}$, which is notoriously computationally expensive. To avoid this, two

particular cases can be considered. If the function f is approximated by a polynomial, the resulting matrix function becomes a matrix polynomial, which can be computed without requiring eigendecompositions. This is the rationale behind the Galerkin–Chebyshev approach proposed in Lang and Pereira (2021), where the function f is replaced by its Chebyshev polynomial approximation over an interval containing the eigenvalues of $\boldsymbol{S}$.

We propose here an alternative approach in the spirit of Lindgren et al. (2011) and Rue and Held (2005). We assume that f is the inverse of a polynomial P taking positive values over $\mathbb{R}_+$, i.e. $f = 1/P$. Then the resulting precision matrix $\boldsymbol{Q}$ of the weights can be expressed as

$$\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1} = (\sqrt{\boldsymbol{C}})P(\boldsymbol{S})(\sqrt{\boldsymbol{C}})^T, \tag{8}$$

which again involves a matrix polynomial instead of a matrix function. Computing the matrix $\boldsymbol{Q}$ can then be done by summing iterates of the matrix $\boldsymbol{S}$, resulting in a matrix whose sparsity depends on the degree of P: the higher the degree of P, the less sparse $\boldsymbol{Q}$ is. This approach, which we refer to as a "matrix free" approach, will be adopted in the algorithms presented in Section 3 for prediction and simulation.

2.2 Second-order Characterization

2.2.1 Stationary and Approximately Stationary Covariance Functions

Let us consider the GRF $\mathcal{Z}$ defined by Eq. (1) on some compact Riemannian manifold $(\mathcal{D}, g)$. It is straightforward to show that its covariance function $C_{\mathcal{Z}}$ can be written as:

$$C_{\mathcal{Z}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{k\in\mathbb{N}} f(\lambda_k)e_k(\boldsymbol{x}_1)e_k(\boldsymbol{x}_2), \quad \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}. \tag{9}$$

In the particular case where $(\mathcal{D}, g)$ is an Euclidean domain equipped with the usual metric, Solin and Särkkä (2020) show that $C_{\mathcal{Z}}$ approximates the covariance of a random field on $\mathcal{D}$ with radial spectral density $f^{1/2}$ (defined in Eq. (1)). They also provide a uniform bound on the error between the actual covariance function of $\mathcal{Z}$ and the covariance function associated with $f^{1/2}$ which shows that the approximation improves as we move further away from the boundary of $\mathcal{D}$. Hence, in this case, $\mathcal{Z}$ approximates an isotropic GRF with covariance C given by

$$C_{\mathcal{Z}}(\boldsymbol{x}_1, \boldsymbol{x}_2) \approx C(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|), \quad \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}, \tag{10}$$

where C is the inverse Fourier transform of $\boldsymbol{w} \mapsto f^{1/2}(\|\boldsymbol{w}\|)$.

More generally, expansions similar to (9) have been used to characterize the covariance of random fields on (Riemannian) manifolds. For instance, Lang and Schwab (2015) use it to describe covariance functions of random fields on the sphere (endowed with its usual metric), and show an explicit link between the regularity of the resulting field and the decay of the sequence $\{f^{1/2}(\lambda_k)\}_{k\in\mathbb{N}}$. On general compact Riemannian manifolds, Borovitskiy et al. (2020) characterize their "Matérn Gaussian processes in the sense of Whittle" through covariance functions of the form (9), by taking $f^{1/2}$ to be the spectral density of the usual Matérn covariance function (i.e. as defined for isotropic random fields on $\mathbb{R}^d$). Hence, the field $\mathcal{Z}$ defined by (1) with covariance function given by (9) can be seen as the counterpart, on the Riemannian manifold $(\mathcal{D}, g)$, of the random fields with radial spectral density $f^{1/2}$ on $\mathbb{R}^d$ and covariance function given by the inverse Fourier transform of $f^{1/2}$. Examples of sampled GRFs with Matérn covariance on different domains are presented in Figure 1.

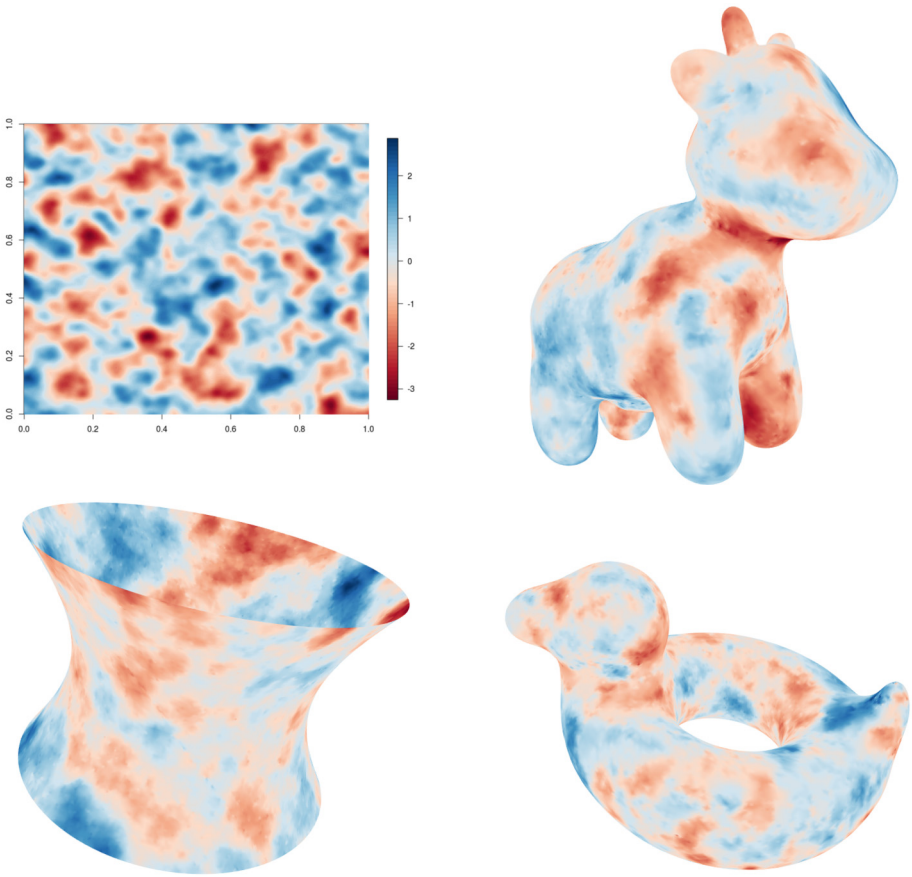

Figure 1: Example of GRFs with Matérn covariance function on the sphere and on surfaces shaped like a cow, a paraboloid and a duck-shaped swim ring.

2.2.2 Nonstationary Covariances

The general construction of random fields on Riemannian manifold presented in the previous section can be used to define nonstationary models of GRFs, and in particular fields that exhibit *local anisotropies*. Such fields are defined on Euclidean domains of dimension $d \in \{2, 3\}$ as follows: around each point of the domain, there is a preferential direction along which the range of highly correlated values is maximal, whereas it is minimal in the orthogonal direction(s). The angles defining the preferential directions are called anisotropy angles and the size of the ranges are called anisotropy ranges. These anisotropy parameters can be graphically represented by an ellipse/ellipsoid whose axes length and direction are respectively given by the anisotropy ranges and angles.

Following the approach described in Pereira (2019), a GRF with local anisotropies on some bounded Euclidean domain $\mathcal{D}$ can be built by defining a GRF on a specific Riemannian manifold: anisotropy angles and ranges can be used to define a metric tensor at each point of $\mathcal{D}$. In other words, at each $\boldsymbol{p} \in \mathcal{D}$, the metric is chosen so that it locally "deforms" $\mathcal{D}$ into a local domain where the anisotropy reduces to isotropy thanks to the composition of a rotation and a scaling that would turn an ellipse/ellipsoid into a circle/sphere (see Figure 2). This transformation

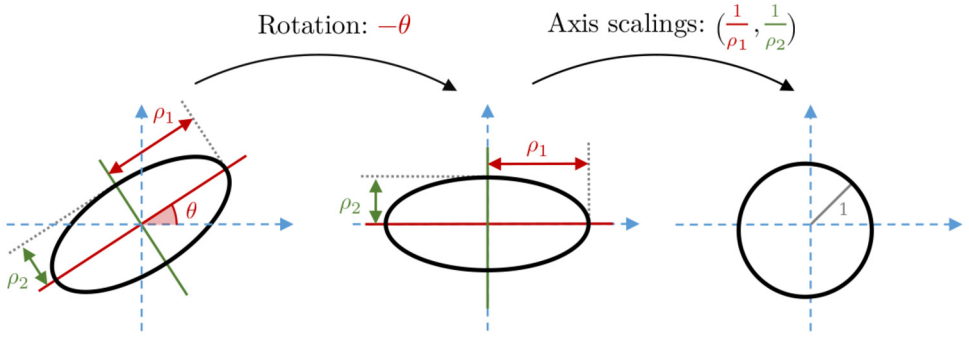

Figure 2: Deformation turning an anisotropy ellipse (with range parameters (ρ_1, ρ_2) and angle θ) into a circle.

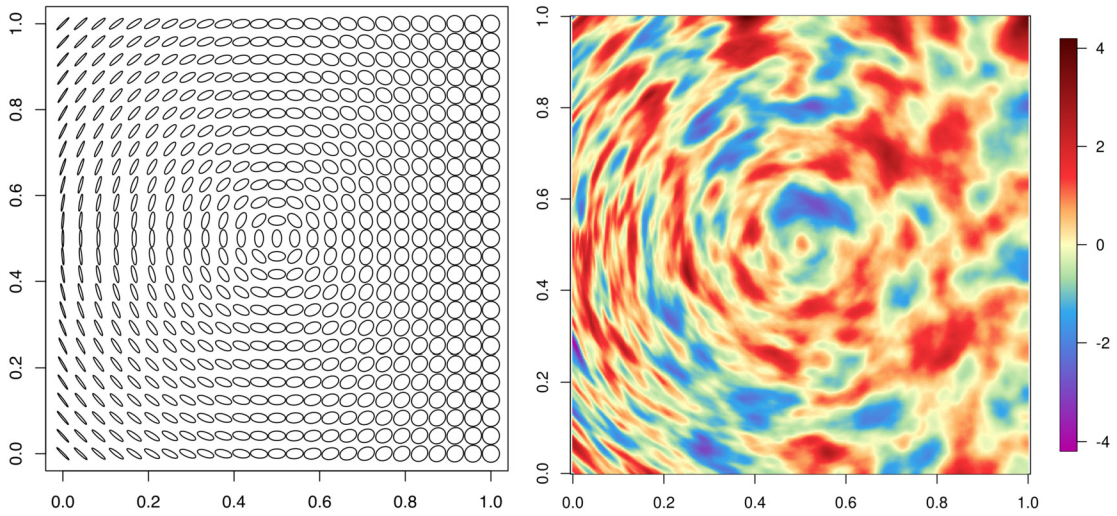

Figure 3: Example of anisotropy parameters (left) and corresponding random field simulation obtained using our method (right), on the unit square.

results in a metric defined as

$$g_{\boldsymbol{p}}(\boldsymbol{u}, \boldsymbol{v}) = \big(\boldsymbol{D}(\boldsymbol{p})^{-1}\boldsymbol{R}(\boldsymbol{p})^{-1}\boldsymbol{u}\big)^T\big(\boldsymbol{D}(\boldsymbol{p})^{-1}\boldsymbol{R}(\boldsymbol{p})^{-1}\boldsymbol{v}\big), \quad \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d, \tag{11}$$

where $\boldsymbol{D}(\boldsymbol{p})$ is the diagonal matrix whose entries are the anisotropy ranges at $\boldsymbol{p} \in \mathcal{D}$ and $\boldsymbol{R}(\boldsymbol{p})$ is the rotation matrix defined from the anisotropy angles at $\boldsymbol{p}$.

To get the covariance properties of the field in the original domain $\mathcal{D}$ equipped with the metric (11), we can first apply the deformation and then use (10), to obtain

$$C_Z(\boldsymbol{p}, \boldsymbol{p}+\boldsymbol{h}) \approx C(g_{\boldsymbol{p}}(\boldsymbol{p}, \boldsymbol{p}+\boldsymbol{h})) = C(\|\boldsymbol{D}(\boldsymbol{p})^{-1}\boldsymbol{R}(\boldsymbol{p})^{-1}\boldsymbol{h}\|), \tag{12}$$

where $\boldsymbol{h} \in \mathbb{R}^d$ is some infinitesimal displacement vector around $\boldsymbol{p}$. It is then straightforward to check that such a covariance locally reproduces the desired anisotropy properties around $\boldsymbol{p}$ (see Chilès and Delfiner (2012) for details). An example of the type of nonstationary fields that can be sampled using this method is presented in Figure 3.

In conclusion, given a compact Euclidean domain $\mathcal{D}$ and a field of anisotropy parameters on $\mathcal{D}$, defining nonstationary random fields with the corresponding anisotropy properties can be done by applying the approach described in Section 2.1 to a tailored Riemannian manifold (namely $\mathcal{D}$ equipped with the metric (11)). Note that similar ideas could be applied to define random fields with varying covariance structure on more general surfaces if one can define coherent fields of anisotropy parameters on such surfaces.

3 Prediction on Riemannian Manifolds

We now show how the construction presented in Section 2 provides efficient prediction algorithms in a quite general setting, which includes nonstationary covariances, non-Euclidean support and non-Matérn covariance functions. Given some spatial domain $\mathcal{D}$, we assume that we observe some real-valued variable Y at $p \geqslant 1$ locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p \in \mathcal{D}$. These observations are modeled as

$$Y(\boldsymbol{x}_i) = \mathcal{Z}(\boldsymbol{x}_i) + \tau\epsilon_i, \quad 1 \leqslant i \leqslant p,$$

where $\epsilon_1, \ldots, \epsilon_p$ are independent standard Gaussian variables, $\tau > 0$, and $\mathcal{Z}$ denotes some GRF on $\mathcal{D}$ acting as a latent variable. Hence, they can be seen as observations of the latent field $\mathcal{Z}$ affected by some independent centered Gaussian noise with variance τ^2.

We aim at making the Best Linear Unbiased Prediction (BLUP) of the variable $\mathcal{Z}$ at q locations $\boldsymbol{x}_{p+1}, \ldots, \boldsymbol{x}_{p+q} \in \mathcal{D}$. Under a Gaussian assumption, the BLUP is equal to the conditional expectation (Tong, 2012), i.e. the optimal prediction in a L^2-sense. In the geostatistical literature, this prediction is referred to as kriging (Chilès and Delfiner, 2012). In most geostatistical approaches, the GRF is either characterized by a covariance function or by a precision matrix. Here, in contrast, the GRF is defined on a triangulation of $\mathcal{D}$ and it is characterized by a positive polynomial P as per Eq. (8). In this section, to derive the kriging algorithm, we first suppose that this polynomial is known. We will show in the next section how τ^2 and the coefficients of P can be estimated from a single realization of the vector of observations $\boldsymbol{Y} = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_p))^T$.

Note that the approach we present here readily generalizes to the case where covariates are added to the model. In that case, the vector of observations $\boldsymbol{Y}$ should contain the residuals obtained after removing from the data points the trend defined by the covariates.

3.1 A "Matrix-Free" Kriging Algorithm

We start with some triangulation of $\mathcal{D}$ with n nodes $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n \in \mathcal{D}$, and $\mathcal{Z}$ is approximated by its finite element approximation Z associated to this triangulation as shown in Section 2. Following Eq. (2), the values of the field Z at the observed locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ can be expressed as a linear combination of the values taken at the triangulation nodes $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$. The vector of observations $\boldsymbol{Y}$ can thus be written as

$$\boldsymbol{Y} = \boldsymbol{M}_{\mathrm{D}}\boldsymbol{Z} + \tau\boldsymbol{\epsilon}, \tag{13}$$

where $\boldsymbol{Z} = (Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_n))^T$, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)^T$ and $\boldsymbol{M}_{\mathrm{D}} \in \mathbb{R}^{p \times n}$ is the so-called design matrix containing the weights defined by

$$[\boldsymbol{M}_{\mathrm{D}}]_{ij} = \psi_j(\boldsymbol{x}_i), \quad 1 \leqslant i \leqslant p, \quad 1 \leqslant j \leqslant n. \tag{14}$$

The next proposition provides an analytic expression for the kriging predictors $Z^*(\boldsymbol{x}_{p+1})$ at some given target location $\boldsymbol{x}_{p+i} \in \mathcal{D}$ $(1 \leqslant i \leqslant q)$. It is proven in Appendix A.1.

Proposition 3.1. *The conditional distribution of* $\boldsymbol{Z}$ *given* $\boldsymbol{Y}$ *is that of a Gaussian vector with mean* $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$ *and covariance matrix* $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$ *given by*

$$\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}] = \boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T(\boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T + \tau^2\boldsymbol{I}_p)^{-1}\boldsymbol{Y} = (\tau^2\boldsymbol{Q} + \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}})^{-1}\boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{Y}, \tag{15}$$

and

$$\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}] = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T(\boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T + \tau^2\boldsymbol{I}_p)^{-1}\boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma} = \tau^2(\tau^2\boldsymbol{Q} + \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}})^{-1}, \tag{16}$$

where $\boldsymbol{\Sigma}$ *is the covariance matrix of* $\boldsymbol{Z}$, $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$ *is its precision matrix, and* τ^2 *and* $\boldsymbol{M}_D$ *are defined in Eq. (13). Then, the vector of kriging predictors* $\boldsymbol{Z}^* = (Z^*(\boldsymbol{x}_{p+1}), \ldots, Z^*(\boldsymbol{x}_{p+q}))^T$ *can be written as*

$$\boldsymbol{Z}^* = \boldsymbol{M}_{\mathrm{T}}\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}], \tag{17}$$

where $\boldsymbol{M}_{\mathrm{T}}$ *is the target design matrix defined by*

$$[\boldsymbol{M}_{\mathrm{T}}]_{ij} = \psi_j(\boldsymbol{x}_{p+i}), \quad 1 \leqslant i \leqslant q, \quad 1 \leqslant j \leqslant n. \tag{18}$$

We have assumed that the function f characterizing the field $\mathcal{Z}$ is the inverse of P, a positive polynomial on $\mathbb{R}_+$. This implies in particular that the precision matrix $\boldsymbol{Q}$ in (15) can be expressed as in Eq. (8). Assuming that P is known, the kriging predictors $\boldsymbol{Z}^*$ in (17) can thus be computed using Algorithm 1.

Algorithm 1: Kriging prediction.

Require: Matrices $\sqrt{\boldsymbol{C}}$, $\boldsymbol{S}$, $\boldsymbol{M}_{\mathrm{D}}$ and $\boldsymbol{M}_{\mathrm{T}}$, as defined in Eqs. (6), (7), (14) and (18).
Require: Polynomial P.
Require: Parameter $\tau > 0$.
Require: Vector of observations $\boldsymbol{Y}$.
1: Solve for $\boldsymbol{X}$ the linear system

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{Y} \tag{19}$$

where

$$\boldsymbol{A} = \tau^2\boldsymbol{Q} + \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}} = \tau^2(\sqrt{\boldsymbol{C}})P(\boldsymbol{S})(\sqrt{\boldsymbol{C}})^T + \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}} \tag{20}$$

2: Return $\boldsymbol{Z}^* := \boldsymbol{M}_{\mathrm{T}}\boldsymbol{X}$.

Note that all the matrices $\sqrt{\boldsymbol{C}}$, $\boldsymbol{S}$, $\boldsymbol{M}_{\mathrm{D}}$ and $\boldsymbol{M}_{\mathrm{T}}$ are sparse, and that the matrix $\boldsymbol{Q}$ is also sparse when P has a low degree. A classical approach for solving the linear system in Algorithm 1 consists in first building and storing the matrix $\boldsymbol{A} = \tau^2\boldsymbol{Q} + \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}}$, and then using a method for solving sparse linear systems. For instance, one could use so-called sparse direct solvers. Such solvers start by factorizing $\boldsymbol{A}$ into sparse triangular factors (using LU or Cholesky decompositions) and then solving the resulting triangular systems. The Cholesky decomposition for large $\boldsymbol{A}$ can be done exactly or with very accurate decomposition using Tile Low Rank approximations with the ExaGeoStat software (Abdulah et al., 2018). Note however, that in this case the matrix $\boldsymbol{A}$ needs to be built and stored.

When these direct approaches are not possible due to size of the problem, an alternative approach using iterative solvers must be used (Saad, 2003). Such solvers only rely on products between the matrix $\boldsymbol{A}$ and vectors, and therefore can be used to approximately solve the linear system in Algorithm 1 without effectively requiring to build and store $\boldsymbol{A}$: only a routine performing the product between $\boldsymbol{A}$ and vectors is needed. Note that such products would only require products between the sparse matrices $\sqrt{\boldsymbol{C}}$, $\boldsymbol{S}$ and $\boldsymbol{M}_D$ and vectors (see Algorithm 2). This approach yields a "matrix-free" approach to kriging in the sense that it does not require to explicitly build and store the precision matrix $\boldsymbol{Q}$ of the field.

Algorithm 2: Product between a matrix $(\alpha \boldsymbol{D} P(\boldsymbol{B}) \boldsymbol{D}^T + \boldsymbol{M}^T \boldsymbol{M})$ and a vector.

Require: Matrices $\boldsymbol{B}, \boldsymbol{D} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{M} \in \mathbb{R}^{p \times n}$

Require: Polynomial $P(X) = \sum_{k=0}^{K} c_k X^k$ for some integer $K \geqslant 0$, $c_0, \ldots, c_K \in \mathbb{R}$

Require: Parameter $\alpha \in \mathbb{R}$

Require: Vector $\boldsymbol{v} \in \mathbb{R}^n$

Compute $\boldsymbol{x} := \boldsymbol{M}^T \boldsymbol{M} \boldsymbol{v} = \left((\boldsymbol{M} \boldsymbol{v})^T \boldsymbol{M} \right)^T$

Compute $\boldsymbol{w} := \boldsymbol{D}^T \boldsymbol{v} = (\boldsymbol{v}^T \boldsymbol{D})^T$

Compute $\boldsymbol{y} := c_K \boldsymbol{w}$

for $k = K - 1, \ldots, 0$ **do**

  Compute $\boldsymbol{y} \leftarrow c_k \boldsymbol{w} + \boldsymbol{S} \boldsymbol{y}$

end for

Return $\alpha \boldsymbol{D} \boldsymbol{y} + \boldsymbol{x}$

3.2 Conditional Simulations

Generating samples from the conditional distribution of $\mathcal{Z}$ (or rather of its approximation Z) given $\boldsymbol{Y}$ is a straightforward task. Since the distribution of Z is entirely specified through the distributions of its weights $\boldsymbol{Z}$, this amounts to sample from the conditional distribution of $\boldsymbol{Z}$ given $\boldsymbol{Y}$, which will be denoted $\pi_{\boldsymbol{Z}|\boldsymbol{Y}}$. Recall from Proposition 3.1 that $\pi_{\boldsymbol{Z}|\boldsymbol{Y}}$ is a multivariate Gaussian distribution with mean $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$ and covariance matrix $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$ given by Eqs. (15) and (16). Hence, sampling from the conditional distribution $\pi_{\boldsymbol{Z}|\boldsymbol{Y}}$ is straightforward, as shown in Algorithm 3.

Algorithm 3: Conditional simulation of the vector $\boldsymbol{Z}$ given the observations $\boldsymbol{Y}$.

Require: Covariance matrix $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$ defined in Eq. (16).

Require: Conditional mean $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$ defined in Eq. (15).

Sample a centered Gaussian vector $\boldsymbol{X}$ with covariance matrix $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$.

$\boldsymbol{X} \leftarrow \boldsymbol{X} + \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$.

Return $\boldsymbol{X}$.

The conditional mean $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$ required for Algorithm 3 is computed using the same methods as those described in Section 3 to compute the kriging predictors. Since an explicit formula is available for $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$ in Eq. (16), sampling the Gaussian vector of the first step in Algorithm 3 could be directly done by finding a factor $\boldsymbol{L}$ such that $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}] = \boldsymbol{L}\boldsymbol{L}^T$ and then returning the product $\boldsymbol{L}\boldsymbol{W}$ where $\boldsymbol{W}$ is a vector of independent standard Gaussian variables. Possibles candidates for $\boldsymbol{L}$ include the Cholesky decomposition of $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$, but also the matrix function $h(\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}])$ where h is the square-root function, and the matrix function $\tilde{h}(\tau^2 \boldsymbol{Q} + \boldsymbol{M}_{\mathrm{D}}^T \boldsymbol{M}_{\mathrm{D}})$ where $\tilde{h}$ is the inverse square-root function. In practice, and as before, both matrix functions could be approximated by matrix polynomials and the "matrix-free" algorithms presented above could be used.

Another method to sample the Gaussian vector of the first step in Algorithm 3 stems from the fact that the vector $\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$ is such a vector, and that the expression of $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$ does not explicitly depend on the vectors $\boldsymbol{Z}$ and $\boldsymbol{Y}$. Hence, by sampling new vectors $\boldsymbol{Z}'$ and $\boldsymbol{Y}'$ with the same distribution as $\boldsymbol{Z}$ and $\boldsymbol{Y}$, and by computing $\boldsymbol{Z}' - \mathbb{E}[\boldsymbol{Z}'|\boldsymbol{Y}']$ we retrieve a centered Gaussian

vector with covariance matrix $\mathrm{Cov}[\boldsymbol{Z}'|\boldsymbol{Y}'] = \mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$. This method is presented in Algorithm 4. It relies on being able to sample the vector $\boldsymbol{Z}'$ described above, which can be done by either:

- Computing the product $\boldsymbol{BW}$ where $\boldsymbol{B}$ is a square-root of $\boldsymbol{\Sigma}$ (e.g. the Cholesky factor of $\boldsymbol{\Sigma}$ or the matrix $(\sqrt{\boldsymbol{C}})^{-T}(1/\sqrt{P})(\boldsymbol{S})$).
- Solving for $\boldsymbol{X}$ the linear system $\tilde{\boldsymbol{B}}\boldsymbol{X} = \boldsymbol{W}$, where $\tilde{\boldsymbol{B}}$ is a square-root of $\boldsymbol{Q}$ (e.g. the Cholesky factor of $\boldsymbol{Q}$ or the matrix $(\sqrt{\boldsymbol{C}})\sqrt{P}(\boldsymbol{S})$).

In practice, the matrix functions appearing above are once again polynomially approximated. More details on the step 1 of Algorithm 4 can be found in Pereira and Desassis (2019).

Algorithm 4: Sampling a centered Gaussian vectors with covariance matrix $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$.

Require: Covariance matrix $\boldsymbol{\Sigma}$ in Eq. (5) or precision matrix in Eq. (8).
Require: Matrix $\boldsymbol{M}_{\mathrm{D}}$ and parameter τ defining the observations $\boldsymbol{Y}$ in Eq. (13).
1: Sample a centered Gaussian vector $\boldsymbol{Z}'$ with covariance matrix $\boldsymbol{\Sigma}$ or precision matrix $\boldsymbol{Q}$.
2: Sample a vector $\boldsymbol{\epsilon}'$ with independent standard Gaussian entries.
3: Compute $\boldsymbol{Y}' := \boldsymbol{M}_{\mathrm{D}}\boldsymbol{Z}' + \tau\epsilon'$.
4: Compute $\mathbb{E}[\boldsymbol{Z}'|\boldsymbol{Y}']$ by replacing $\boldsymbol{Y}$ by $\boldsymbol{Y}'$ in Eq. (15).
5: Return $\boldsymbol{Z}' - \mathbb{E}[\boldsymbol{Z}'|\boldsymbol{Y}']$.

4 Estimation of the Parameters

In order to apply the kriging and simulation procedures presented above, one needs to know the polynomial P characterizing the field $\mathcal{Z}$, as well as the parameter τ^2 defining the variance of the Gaussian noise. Usually, these parameters are not known, and they must be estimated from the observations $\boldsymbol{Y}$. We now show how a maximum likelihood approach can be efficiently implemented. Let us denote by $\boldsymbol{\theta}$ the vector of parameters containing the coefficients of P and the variance τ^2. Note then that, following Eq. (13), the vector $\boldsymbol{Y}$ is a centered Gaussian vector with covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{Y}}(\boldsymbol{\theta}) = \boldsymbol{M}_D\,\boldsymbol{Q}(\boldsymbol{\theta})^{-1}\boldsymbol{M}_D^T + \tau^2\boldsymbol{I}_p, \tag{21}$$

where $\boldsymbol{Q}(\boldsymbol{\theta}) = (\sqrt{\boldsymbol{C}})P(\boldsymbol{S})(\sqrt{\boldsymbol{C}})^T$. The log-likelihood of $\boldsymbol{Y}$ is thus

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2}\big(p\log 2\pi - \log|\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta})| + \boldsymbol{Y}^T\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta})\boldsymbol{Y}\big),$$

where $\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}_{\boldsymbol{Y}}(\boldsymbol{\theta})^{-1}$. Considering the matrix $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\theta})$ defined in (20), we can write

$$\boldsymbol{Y}^T\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta})\boldsymbol{Y} = \tau^{-2}\big(\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{M}_D\boldsymbol{A}(\boldsymbol{\theta})^{-1}\boldsymbol{M}_D^T\boldsymbol{Y}\big), \tag{22}$$

and

$$\begin{aligned}\log|\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta})| &= \log|\boldsymbol{Q}(\boldsymbol{\theta})| + (n-p)\log\tau^2 - \log|\boldsymbol{A}(\boldsymbol{\theta})| \\ &= \log|P(\boldsymbol{S})| + 2\log|\sqrt{\boldsymbol{C}}| + (n-p)\log\tau^2 - \log|\boldsymbol{A}(\boldsymbol{\theta})|.\end{aligned} \tag{23}$$

The precise computation of the log-likelihood deserves some comments. The quadratic form (22) is computed using the methods introduced to solve the linear system, as shown in Algorithm 1 and in Eq. (19). Evaluating the log-determinants in Eq. (23) could be done using a Cholesky (or LU) factorization of the matrices $\boldsymbol{Q}(\boldsymbol{\theta})$ and $\boldsymbol{A}(\boldsymbol{\theta})$ and then summing the log of the diagonal

elements of the resulting triangular factors. However, one can also use a "matrix-free" approach to compute an approximation of these log-determinants. This approach, detailed in the rest of this section, is a generalization to matrix functions of the results in Han et al. (2015) and can be seen as a particular instance of Hutchinson estimator (Hutchinson, 1989). It is based on the following result (proven in Appendix A.2).

Proposition 4.1. *Let* $\boldsymbol{B}$ *be a diagonalizable matrix and* $h : \mathbb{R} \to (0, \infty)$*. The log-determinant of the matrix function* $h(\boldsymbol{B})$ *satisfies the relation*

$$\log |h(\boldsymbol{B})| = \mathit{Trace}(\log h(\boldsymbol{B})) = \mathbb{E}[\boldsymbol{W}^T \log h(\boldsymbol{B}) \boldsymbol{W}],$$

where $\boldsymbol{W}$ *is a vector whose entries are independent zero-mean unit-variance random variables, and* $\log h(\boldsymbol{B})$ *is the matrix function defined from the function* $\log h$*.*

The matrix function $\log h(\boldsymbol{B})$ can in practice be approximated by a matrix polynomial $P_{\log h}(\boldsymbol{B})$ where $P_{\log h}$ denotes a polynomial approximation of $\log h$ over an interval containing the eigenvalues of $\boldsymbol{B}$ (and defined using for instance Chebyshev polynomial approximation). Then, we can approximate $\log |h(\boldsymbol{B})|$ as

$$\log |h(\boldsymbol{B})| \approx \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{W}_m^T P_{\log h}(\boldsymbol{B}) \boldsymbol{W}_m, \tag{24}$$

where $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_M$ denote M independent samples of $\boldsymbol{W}$ (defined for instance from a Gaussian or Rademacher distribution). Similarly to Algorithm 2, each quadratic form in Eq. (24) is computed in an iterative way while only requiring products between the matrix $\boldsymbol{B}$ and vectors. This approach can be used to compute the log-determinant (23) by noting that both $\log |P(\boldsymbol{S})|$ and $\log |\boldsymbol{A}(\boldsymbol{\theta})|$ can be written in the form $\log h(\boldsymbol{B})$ with $\boldsymbol{B} = \boldsymbol{S}$ and $h = P$ for the former, and $\boldsymbol{B} = \boldsymbol{A}(\boldsymbol{\theta})$ and h to be the identity map for the latter. The whole procedure to compute $\log |\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta})|$ is summarized in Algorithm 5. Note in particular that this algorithm requires to know the intervals containing the eigenvalues of $\boldsymbol{S}$ and those of $\boldsymbol{A}(\boldsymbol{\theta})$. Details on the computation of these intervals can be found in Appendix B. A discussion about the "matrix-free" approach to the computation of $\log |h(\boldsymbol{B})|$ is deferred to Section 6.

Since we can now evaluate the log-likelihood for any vector of parameters, we can plug Algorithm 5 into any optimization algorithm that only requires evaluations of an objective function to maximize it. Examples of such algorithms include the Nelder-Mead algorithm, and any gradient-descent algorithm for which the gradients would be numerically approximated by finite differences (Nocedal and Wright, 2006).

Finally, note that the procedure presented in this section naturally extends to the case where covariates are added to the model. Indeed, assuming now that the observations are modeled as $\tilde{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Y}$ where $\boldsymbol{Y}$ is defined as before, $\boldsymbol{X}$ is a matrix of covariates and $\boldsymbol{\beta}$ a vector containing the associated regression coefficients. Then, the maximum likelihood estimate of $\boldsymbol{\beta}$ is given by $\widehat{\boldsymbol{\beta}}(\boldsymbol{\theta}) = (\boldsymbol{X}^T \boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta}) \boldsymbol{X})^{-1} \boldsymbol{X} \boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta}) \tilde{\boldsymbol{Y}}$, and the estimation of the parameter can be carried out by maximizing the likelihood $\mathcal{L}$, where $\boldsymbol{Y}$ is now replaced by $\tilde{\boldsymbol{Y}} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}(\boldsymbol{\theta})$.

5 Illustration

The proposed algorithms are illustrated on synthetic very large data sets. Our aim is to show that our approach compares very well with the GMRF approximation in Lindgren et al. (2011) on very large grids, even in cases that are favorable to the GMRF approximation.

Algorithm 5: Computation of $\log|\boldsymbol{Q}_{\boldsymbol{Y}}(\boldsymbol{\theta})|$ defined in Eq. (23).

Require: Matrices $\sqrt{\boldsymbol{C}}, \boldsymbol{S} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{M}_{\mathrm{D}} \in \mathbb{R}^{p\times n}$ as defined in Eqs. (6), (7) and (14)
Require: Vector $\boldsymbol{\theta}$ containing the coefficients of P and the variance parameter τ^2
Require: Number of samples M
Compute the coefficients of a polynomial approximation $P_{\log P}$ of the function $\lambda \mapsto \log P(\lambda)$ over an interval containing the eigenvalues of $\boldsymbol{S}$
Compute the coefficients of a polynomial approximation $P_{\log}$ of the function $\lambda \mapsto \log\lambda$ over an interval containing the eigenvalues of the matrix $\boldsymbol{A}(\theta)$ defined in Eq. (20)
Set $Q_1 = Q_2 := 0$
for $m = 1, \ldots, M$ **do**
    Sample a vector $\boldsymbol{W}$ with independent identically distributed entries with mean 0 and variance 1
    Compute $\boldsymbol{u} := P_{\log P}(\boldsymbol{S})\boldsymbol{W}$ using Algorithm 2
    Set $Q_1 \leftarrow Q_1 + \boldsymbol{W}^T\boldsymbol{u}$
    Compute $\boldsymbol{v} := P_{\log}(\boldsymbol{A}(\boldsymbol{\theta}))\boldsymbol{W}$ using Algorithm 2
    Set $Q_2 \leftarrow Q_2 + \boldsymbol{W}^T\boldsymbol{v}$
end for
Compute $L = (Q_1 - Q_2)/M + (n-p)\log\tau^2 + 2\sum_{i=1}^{n}\log[\sqrt{\boldsymbol{C}}]_{ii}$
return L

5.1 Simulation and Kriging

We first consider the case of 3D standardized GRF with varying anisotropies in the horizontal plane and an exponential covariance, which corresponds in 3D to the choice $P(\lambda) = (\kappa^2 + \lambda)^2$. Following the remarks of Section 3.2, it is sampled by using a Chebyshev polynomial approximation of $\lambda \mapsto 1/\sqrt{P}(\lambda) = (\kappa^2 + \lambda)^{-1}$ of degree 268 (see e.g. Pereira and Desassis, 2019, for details). We chose a mesh built from 6 tetrahedrons in each cell of a regular grid with lags suitably chosen to fit with the ranges of the GRF. The resulting size of the discretized vector $\boldsymbol{Z}$ is about 1.5×10^7. On this extremely large grid, the simulation takes only 30 seconds on a laptop running at 1.9 Ghz on 8 cores. Most of the time is spent for the matrix-vector products implied by the polynomial approximation as shown in Algorithm 2.

A sub-sampling of this simulation is done to obtain 10^5 randomly located observations which are used to perform kriging. Since interpolation by kriging is smoother than a simulation, a coarser mesh can be used. The size of the resulting kriging system is about 7×10^6. A measurement error with a variance $\tau^2 = 0.01$ is added to the model. Conjugate gradient without preconditioning is used to solve the system of Eq. (19) in Algorithm 1. The algorithm converges in 1098 iterations for a computing time around 400 seconds. Results are displayed in Figure 4. Note that in most applications, the variance τ^2 is greater than 1% of the variance. In these cases, the system of Eq. (19) is better conditioned, leading to a faster convergence of the conjugate gradient. Kriging is done here in a nonstationary context, with a stationary mesh which is thus far from optimality in most locations. If the simulation were stationary, the mesh could be tailored to the specific model at hand. The grid could be oriented according to the anisotropy tensor and the lag in each direction could be chosen according to the associated directional range of the model. As a result, a given accuracy would be obtained with a lower degree of the Chebyshev polynomial and system Eq. (19) would be better conditioned, thus leading to faster

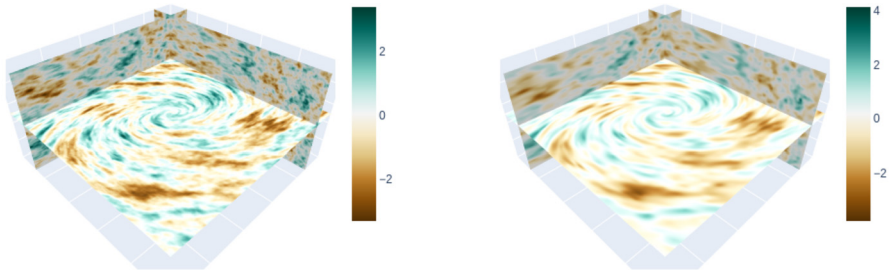

Figure 4: Left: 3D simulation of a GRF with varying anisotropies. Right: Kriging estimate using 10^5 randomly located samples from the simulation on the left.

Table 1: Coefficients of the polynomials (true, estimated, and initial).

Degree	0	1	2	3
True	1	−0.75	−0.75	1
Estimated	0.55	2.42	−5.17	2.60
Initial	0.001	0	0	0

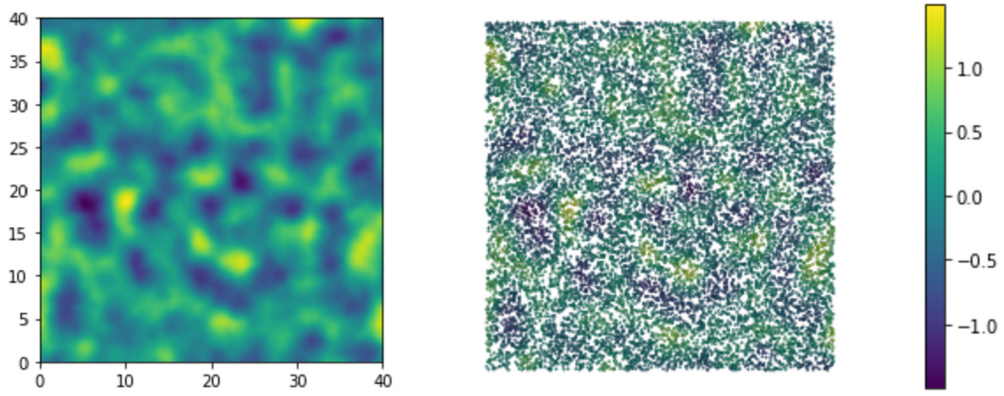

Figure 5: Left: Realization of the GMRF on the square. Right: Sampling of 1.5×10^4 noisy observations from the simulation on the left.

computations. In our experience, stationary simulations and kriging are usually 10 times faster than nonstationary ones, all other settings being equal.

5.2 Estimation of the Parameters of a GMRF

In this example, the coefficients of a polynomial P characterizing an isotropic GMRF and the measurement error τ^2 are estimated by the approach described in Section 4. The coefficients of the true P are given in Table 1. It does not correspond to a Matérn type. 1.5×10^4 synthetic observations are sampled from the model at random locations of a square of size 40 and a Gaussian noise with variance $\tau^2 = 0.01$ is added (see Figure 5).

To ensure that P takes strictly positive values over $\mathbb{R}_+$ during the estimation, the problem

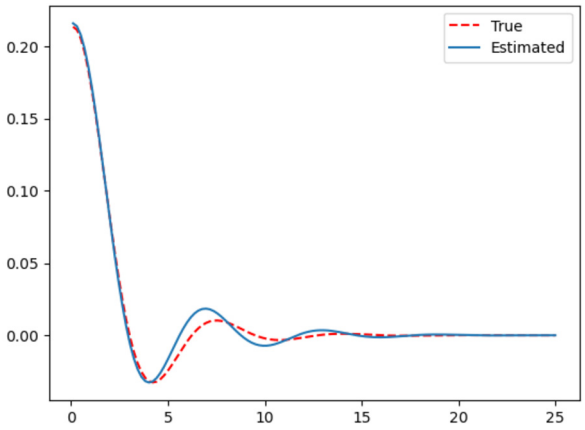

Figure 6: Covariances obtained from P by FFT (true model and estimated).

is re-parameterized by using two arbitrary polynomials P_1 and P_2 as follows:

$$P(x) = P_1^2(x) + x P_2^2(x). \tag{25}$$

Indeed, all the positive polynomials over $\mathbb{R}_+$ can be written according to Eq. (25). A fixed positive value ε is added to the right-hand side of Eq. (25) to ensure strict positivity of P. In this study, ε is set to 10^{-3}. The components of $\boldsymbol{\theta}$ are the coefficients of P_1 and P_2. The likelihood is maximized with the COBYLA algorithm (Powell, 1994), using several initial guesses and selecting the best output. To compute the log-determinant through Hutchinson estimators, the same Gaussian vectors are used for all the iterations. First only one vector is used in order to find an acceptable solution. Then the algorithm is relaunched and 10 random vectors are used until convergence.

The true coefficients of P (resulting from those of P_1 and P_2) are given in Table 1 along with the estimated coefficients. We also give in this table the initial guess that yielded the estimated coefficients. The initial value for τ^2 was set to 1 and its estimation is 0.00944. At first glance, the estimation results does not seem very accurate. However, the associated covariance functions, computed by Fast-Fourier-Transform (FFT) and displayed on Figure 6 lead to a more positive conclusion. Indeed, the estimated covariance is close to the true one, especially near the origin. Nevertheless, the likelihood seems to have several modes and the results are sensitive to the choice of the initial values. τ^2 is generally well estimated but the estimated covariance of the underlying GMRF is not always so close to the true one. Despite this optimization problem, further discussed in the next section, the matrix-free approach allows to approximate the likelihood in order to estimate the shape of the covariance, even with a moderate number of random vectors for the Hutchinson estimator.

6 Discussion and Conclusion

We have proposed a generic approach for defining GRFs on compact Riemannian manifolds. It combines two ingredients of modern geostatistics, as pioneered in Lindgren et al. (2011): the definition of GRFs through the expansions in the eigenfunctions of the Laplace–Beltrami operator on the Riemannian manifold and the finite element approximation of these GRFs. This approach is quite general. Not limited to spheres, it can be applied to construct valid GRFs on

any smooth compact manifold – and in particular to any compact surface. Moreover, since the GRF is characterized by its spectral density, it is not limited to Whittle–Matérn random fields. The proposed Riemannian metric offers a straightforward interpretation of local anisotropies. Our approach is thus also perfectly suited to the analysis of data in Euclidean domains with nonstationary correlation ranges and nonstationary anisotropies.

For this quite general class of GRFs, we have provided efficient algorithms that do not require to build and store possibly very large matrices. Instead, our "matrix-free" approach is grounded on algorithms only requiring efficient routines for computing the product between very sparse matrices and vectors, as shown in Algorithm 2. This ensures the scalability of this method, thus paving the way to efficient nonstationary geostatistics for large datasets. We have shown on synthetic examples that our approach is able to handle grids with millions of nodes (up to 10^7) in only few minutes.

The main bottleneck is the computation of the log-determinant of the matrix function $h(\boldsymbol{B})$ in Proposition 4.1. When possible, the Cholesky decomposition of $\boldsymbol{B}$ is best. This is the case for matrices whose size is in the range of 10^4 or smaller. The "matrix-free" approach described above scales very well with the size of $\boldsymbol{B}$. It has been successfully applied in the context of seismic filtering (Pereira, 2019, Pereira et al., 2020) and it is now part of industrial codes able to filter very large noisy seismic datasets (in the range 10^6 to 10^7 in few minutes). Even though being scalable, our algorithms could be accelerated in some situations. Algorithm 5 requires a Monte-Carlo loop relating to the Hutchinson estimator Eq. (24) and iterated products as described in Algorithm 2 which can be long if the degree of the polynomial is high. Further research is needed in order to better assess the precision of the approximation (24) in view of optimizing the computation of the log-determinant.

Besides, the inference of parameters through likelihood maximization poses some challenges due to the existence in general of local maxima (Williams and Rasmussen, 2006). In our case, we performed the minimization with several initial guesses and chose the one yielding the best result. In order to robustify the estimation, one could move to algorithms better equipped to handle local maxima, e.g. particle swarm optimizers and other evolutionary optimizers. Inspired by the recent advances in Machine Learning and Deep Learning, another line of research is how to maximize the log-likelihood using (stochastic) gradient algorithms (Simon, 2013).

We believe that the "matrix-free" approach is thus a very promising tool for the analysis of environmental dataset that will be tested in future work, including in a spatio-temporal setting. Other directions for future research include 3D extensions and the modeling of nonstationary anisotropies on spheres and on manifolds in general.

Supplementary Material

The code used to perform the maximum likelihood estimation in Section 5.2 is available at https://github.com/mike-pereira/matrix-free-mle.

A Proofs

A.1 Proof of Proposition 3.1

Consider the vector $\boldsymbol{X}$ given by

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{Y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{np} \\ \boldsymbol{M}_{\mathrm{D}} & \tau \boldsymbol{I}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{\epsilon} \end{pmatrix}.$$

Note that $\boldsymbol{X}$ is a centered Gaussian vector with covariance matrix

$$\mathrm{Cov}[\boldsymbol{X}] = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{np} \\ \boldsymbol{M}_{\mathrm{D}} & \tau \boldsymbol{I}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{0}_{nn} \\ \boldsymbol{0}_{pp} & \boldsymbol{I}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{np} \\ \boldsymbol{M}_{\mathrm{D}} & \tau \boldsymbol{I}_p \end{pmatrix}^T = \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T \\ \boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma} & \boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T + \tau^2 \boldsymbol{I}_p \end{pmatrix}. \tag{26}$$

Since $\boldsymbol{X}$ is multivariate Gaussian it follows that the conditional distribution of $\boldsymbol{Z}$ given $\boldsymbol{Y}$ is a Gaussian vector with mean $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]$ and covariance matrix $\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}]$ (Tong, 2012, Theorem 3.3.4) given by

$$\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}] = \boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T(\boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T + \tau^2\boldsymbol{I}_p)^{-1}\boldsymbol{Y},$$
$$\mathrm{Cov}[\boldsymbol{Z}|\boldsymbol{Y}] = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T(\boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma}\boldsymbol{M}_{\mathrm{D}}^T + \tau^2\boldsymbol{I}_p)^{-1}\boldsymbol{M}_{\mathrm{D}}\boldsymbol{\Sigma}.$$

Then, Eqs. (15) and (16) follow from computing $\mathrm{Cov}[\boldsymbol{X}]^{-1}$ from Eq. (26).

Finally, recall that since we are dealing with Gaussian vectors, the vector of kriging predictors $\boldsymbol{Z}^*$ coincides with the conditional expectation of the vector $\boldsymbol{Z}_{\mathrm{T}} = (Z(x_{p+1}), \ldots, Z(x_{p+q}))^T$, given the observations $\boldsymbol{Y}$. Hence, by linearity of the expectation and definition of $\boldsymbol{M}_{\mathrm{T}}$, we have

$$\boldsymbol{Z}^* = \mathbb{E}[\boldsymbol{Z}_{\mathrm{T}}|\boldsymbol{Y}] = \mathbb{E}[\boldsymbol{M}_{\mathrm{T}}\boldsymbol{Z}|\boldsymbol{Y}] = \boldsymbol{M}_{\mathrm{T}}\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Y}]. \quad \square$$

A.2 Proof of Proposition 4.1

By definition of matrix functions, and denoting by $\lambda_1, \ldots, \lambda_n$ the eigenvalues of $\boldsymbol{B}$, we have

$$\mathrm{Trace}(\log h(\boldsymbol{B})) = \sum_{i=1}^{n} \log h(\lambda_i) = \log \prod_{i=1}^{n} h(\lambda_i) = \log |h(\boldsymbol{B})|.$$

The second equality is then a direct consequence of the well-known Hutchinson trace estimator (Hutchinson, 1989). $\square$

B Intervals of Eigenvalues

We here show how intervals containing all eigenvalues of the matrices $\boldsymbol{S}$ and $\boldsymbol{A}$ respectively defined in Eq. (4) and Eq. (20) can be computed. Since the matrix $\boldsymbol{S}$ is positive definite, an interval containing its eigenvalues is obtained by considering $[0, \lambda_{\max}(\boldsymbol{S})]$, where $\lambda_{\max}(\boldsymbol{S})$ is an upper-bound of the eigenvalues of $\boldsymbol{S}$. This upper-bound can be obtained by taking $\lambda_{\max}(\boldsymbol{S}) = \sqrt{\mathrm{Trace}(\boldsymbol{S}^T\boldsymbol{S})}$ or by relying on Gershgorin circle theorem (Gershgorin, 1931), which we now recall.

Proposition B.1. *The eigenvalues of a symmetric matrix $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ with entries B_{ij} are contained in the interval*

$$[\min_{1\leqslant i\leqslant n}(B_{ii} - R_i), \max_{1\leqslant i\leqslant n}(B_{ii} + R_i)],$$

where $R_i = \sum_{j\neq i} |B_{ij}|$, $1 \leqslant i \leqslant n$.

Regarding the matrix $\boldsymbol{A} = \tau^2\boldsymbol{Q} + \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}}$, lower and upper bounds of its eigenvalues are given in the next proposition.

Proposition B.2. *Let $\lambda_{\min}(\boldsymbol{A})$ (resp. $\lambda_{\max}(\boldsymbol{A})$) denote some lower (resp. upper) bound of the eigenvalues of the matrix $\boldsymbol{A}$. Then,*

$$\lambda_{\min}(\boldsymbol{A}) = \tau^2\left(\min_{1\leqslant i\leqslant n}[\sqrt{\boldsymbol{C}}]_{ii}^2\right)\left(\inf_{\lambda\in[0,\lambda_{\max}(\boldsymbol{S})]}P(\lambda)\right),$$
$$\lambda_{\max}(\boldsymbol{A}) = \tau^2\left(\max_{1\leqslant i\leqslant n}[\sqrt{\boldsymbol{C}}]_{ii}^2\right)\left(\sup_{\lambda\in[0,\lambda_{\max}(\boldsymbol{S})]}P(\lambda)\right) + \left(\max_{1\leqslant i\leqslant n}\sum_{k=1}^{p}[\boldsymbol{M}_{\mathrm{D}}]_{ki}\right).$$

Proof. To ease the notation, let $\boldsymbol{B} = \boldsymbol{M}_{\mathrm{D}}^T\boldsymbol{M}_{\mathrm{D}}$. Note that we can take

$$\lambda_{\min}(\boldsymbol{A}) = \tau^2\lambda_{\min}(\boldsymbol{Q}) + \lambda_{\min}(\boldsymbol{B}), \quad \lambda_{\max}(\boldsymbol{A}) = \tau^2\lambda_{\max}(\boldsymbol{Q}) + \lambda_{\max}(\boldsymbol{B}),$$

where $\lambda_{\min}(\cdot)$ (resp. $\lambda_{\max}(\cdot)$) denotes some lower (resp. upper) bound of the eigenvalues of a matrix. On the one hand, since the matrix $\sqrt{\boldsymbol{C}}$ is diagonal, we can take $\lambda_{\min}(\boldsymbol{Q}) = \min_{1\leqslant i\leqslant n}[\sqrt{\boldsymbol{C}}]_{ii}^2\lambda_{\min}(P(\boldsymbol{S}))$ and $\lambda_{\max}(\boldsymbol{Q}) = \max_{1\leqslant i\leqslant n}[\sqrt{\boldsymbol{C}}]_{ii}^2\lambda_{\max}(P(\boldsymbol{S}))$. Recalling then the definition of matrix functions, it is clear that the eigenvalues of $P(\boldsymbol{S})$ are lower (resp. upper) bounded by the infimum (resp. supremum) of P over an interval containing the eigenvalues of $\boldsymbol{S}$, e.g. $[0, \lambda_{\max}(\boldsymbol{S})]$.

Finally, noting that $\boldsymbol{B}$ is positive semi-definite, we can take $\lambda_{\min}(\boldsymbol{B}) = 0$. Then, Proposition B.1 and the non-negativity of the entries of $\boldsymbol{B}$ allow us to get $\lambda_{\max}(\boldsymbol{B}) = \max_{1\leqslant i\leqslant n}(B_{ii} + R_i)$ where

$$B_{ii} + R_i = B_{ii} + \sum_{j\neq i}B_{ij} = \sum_{j=1}^{n}B_{ij} = \sum_{j=1}^{n}\sum_{k=1}^{p}[\boldsymbol{M}_{\mathrm{D}}]_{ki}[\boldsymbol{M}_{\mathrm{D}}]_{kj} = \sum_{k=1}^{p}[\boldsymbol{M}_{\mathrm{D}}]_{ki}\sum_{j=1}^{n}[\boldsymbol{M}_{\mathrm{D}}]_{kj}.$$

Noting then that the rows of $\boldsymbol{M}_{\mathrm{D}}$ sum to 1 (since they correspond to linear interpolation weights), we have $B_{ii} + R_i = \sum_{k=1}^{p}[\boldsymbol{M}_{\mathrm{D}}]_{ki}$, which ends the proof. □

C Integration on Riemannian Manifolds

We recall usual formulas related to the computation of integrals defined over Riemannian manifolds. We refer the reader to Jost (2008), Lee (2013) for further details. Let $(\mathcal{D}, g)$ be a compact Riemannian manifold of dimension d. Let (U, ϕ) denote a coordinate chart of $\mathcal{D}$, i.e. U is an open subset of $\mathcal{D}$ and ϕ is a homeomorphism mapping U to an open subset of $\mathbb{R}^d$. Recall that the integral of a function $f : \mathcal{D} \to \mathbb{R}$ over U is defined as the quantity

$$\int_U f dV_g = \int_{\phi(U)} f\circ\phi^{-1}(\boldsymbol{x})\sqrt{|g|(\phi^{-1}(\boldsymbol{x}))}d\boldsymbol{x},$$

where $|g|(\phi^{-1}(\boldsymbol{x}))$ is the determinant of the metric tensor of g at $\phi^{-1}(\boldsymbol{x}) \in U$, expressed in the coordinate chart (U, ϕ). The integral of f over $\mathcal{D}$ is then obtained by gluing together (using a partition of unity) local integrals over a collection of coordinate charts that cover $\mathcal{D}$.

In particular, assuming now that we have a triangulation of $\mathcal{D}$, the integral of f over $\mathcal{D}$ can be obtained by summing local integrals over each triangle T of the triangulation. In this case, the diffeomorphism ϕ associated to T is the map that sends T to the standard simplex of $\mathbb{R}^d$.

Let $L^2(\mathcal{D})$ denote the set of square-integrable functions of $(\mathcal{D}, g)$. $L^2(\mathcal{D})$ is a Hilbert space when equipped with the inner product $(\cdot,\cdot)$ defined by $(f_1, f_2) = \int_{\mathcal{D}} f_1 f_2 dV_g$. Note that for any differentiable functions f_1 and f_2 we denote by $(\nabla f_1, \nabla f_2)$ the integral over $\mathcal{D}$ of the function $h : p \mapsto g_p(\nabla f_1(p), \nabla f_2(p))$.

And in turn, given a coordinate chart (U, ϕ) of $\mathcal{D}$, the integral of h over U reduces to

$$\int_U h dV_g = \int_{\phi(U)} \nabla_{\mathbb{R}^d}(f_1 \circ \phi^{-1})(\boldsymbol{x})^T \boldsymbol{G}(\phi^{-1}(\boldsymbol{x}))^{-1} \nabla_{\mathbb{R}^d}(f_2 \circ \phi^{-1})(\boldsymbol{x}) \sqrt{|g|(\phi^{-1}(\boldsymbol{x}))} d\boldsymbol{x},$$

where $\boldsymbol{G}(\cdot)$ denotes the metric tensor at given point of $\mathcal{D}$ and expressed in the coordinate chart (U, ϕ), and $\nabla_{\mathbb{R}^d}$ denotes the usual gradient of functions of $\mathbb{R}^d$.

D Galerkin Approximation

Let $\psi_1, \ldots, \psi_n$ denote n linearly independent functions from $\mathcal{D}$ to $\mathbb{R}$ and let V_n denote their linear span. The Galerkin approximation $-\Delta_n$ of the Laplace–Beltrami operator $-\Delta$ is the endomorphism mapping any $f \in V_n$ to the element $-\Delta_n f \in V_n$ satisfying for any $u \in V_n$, $(-\Delta_n f, u) = (-\Delta f, u)$. This endomorphism is diagonalizable, and shares the same eigenvalues as the scaled stiffness matrix $\boldsymbol{S}$ defined in Eq. (4) (Lang and Pereira, 2021).

Competing Interests

The authors have no competing interests to declare.

Funding

The authors acknowledge the support of the Mines Paris / INRAE chair “Geolearning”.

References

Abdulah S, Ltaief H, Sun Y, Genton MG, Keyes DE (2018). ExaGeoStat: A high performance unified software for geostatistics on manycore systems. *IEEE Transactions on Parallel and Distributed Systems*, 29(12): 2771–2784.

Borovitskiy V, Azangulov I, Terenin A, Mostowsky P, Deisenroth M, Durrande N (2021). Matérn Gaussian processes on graphs. In: *International Conference on Artificial Intelligence and Statistics*, 2593–2601. PMLR.

Borovitskiy V, Terenin A, Mostowsky P, Deisenroth M (2020). Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 33: 12426–12437.

Carrizo Vergara R, Allard D, Desassis N (2022). A general framework for SPDE-based stationary random fields. *Bernoulli*, 28(1): 1–32.

Chilès JP, Delfiner P (2012). *Geostatistics: Modeling Spatial Uncertainty. 2nd Edition. Wiley Series In Probability and Statistics.*

Emery X, Porcu E (2019). Simulating isotropic vector-valued Gaussian random fields on the sphere through finite harmonics approximations. *Stochastic Environmental Research and Risk Assessment*, 33(8): 1659–1667.

Fouedjio F, Desassis N, Rivoirard J (2016). A generalized convolution model and estimation for non-stationary random functions. *Spatial Statistics*, 16: 35–52.

Fouedjio F, Desassis N, Romary T (2015). Estimation of space deformation model for non-stationary random functions. *Spatial Statistics*, 13: 45–61.

Fuglstad GA, Lindgren F, Simpson D, Rue H (2015a). Exploring a new class of non-stationary spatial Gaussian random fields with varying local anisotropy. *Statistica Sinica*, 25: 115–133.

Fuglstad GA, Simpson D, Lindgren F, Rue H (2015b). Does non-stationary spatial data always require non-stationary random fields? *Spatial Statistics*, 14: 505–531.

Gershgorin S (1931). Über die Abgrenzung der Eigenwerte einer matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, 7: 749–754.

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

Han I, Malioutov D, Shin J (2015). Large-scale log-determinant computation through stochastic Chebyshev expansions. In: *International Conference on Machine Learning*, 908–917. PMLR.

Heaton MJ, Datta A, Finley AO, Furrer R, Guinness J, Guhaniyogi R, et al. (2019). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*, 24(3): 398–425.

Higdon D, Swall J, Kern J (1999). Non-stationary spatial modeling. *Bayesian Statistics*, 6(1): 761–768.

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

Huang H, Abdulah S, Sun Y, Ltaief H, Keyes DE, Genton MG (2021). Competition on spatial statistics for large datasets. *Journal of Agricultural, Biological and Environmental Statistics*, 26(4): 580–595.

Hutchinson MF (1989). A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3): 1059–1076.

Jeong J, Jun M, Genton MG (2017). Spherical process models for global spatial statistics. *Statistical Science*, 32(4): 501–513.

Jost J (2008). *Riemannian Geometry and Geometric Analysis*. Springer.

Lang A, Pereira M (2021). Galerkin–Chebyshev approximation of Gaussian random fields on compact riemannian manifolds. arXiv preprint: https://arxiv.org/abs/2107.02667.

Lang A, Schwab C (2015). Isotropic Gaussian random fields on the sphere: Regularity, fast simulation and stochastic partial differential equations. *The Annals of Applied Probability*, 25(6): 3047–3094.

Lantuéjoul C, Freulon X, Renard D (2019). Spectral simulation of isotropic Gaussian random fields on a sphere. *Mathematical Geosciences*, 51(8): 999–1020.

Lee JM (2013). *Introduction to Smooth Manifolds*. Springer.

Lindgren F, Rue H, Lindström J (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4): 423–498.

Marinucci D, Peccati G (2011). *Random Fields on the Sphere: Representation, Limit Theorems and Cosmological Applications. London Mathematical Society Lecture Note Series*. Cambridge University Press.

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

Paciorek CJ, Schervish MJ (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics*, 17(5): 483–506.

Pereira M (2019). Generalized random fields defined on Riemannian manfolds: Theory and practice, Ph.D. thesis, MINES ParisTech, PSL University.

Pereira M, Desassis N (2019). Efficient simulation of Gaussian Markov random fields by Chebyshev polynomial approximation. *Spatial Statistics*, 31: 100359.

Pereira M, Desassis N, Magneron C, Palmer N (2020). A matrix-free approach to geostatistical

filtering. arXiv preprint: https://arxiv.org/abs/2004.02799.

Perrin O, Senoussi R (2000). Reducing non-stationary random fields to stationarity and isotropy using a space deformation. *Statistics & Probability Letters*, 48(1): 23–32.

Porcu E, Furrer R, Nychka D (2021). 30 years of space–time covariance functions. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13(2): e1512.

Porcu E, Mateu J, Christakos G (2009). Quasi-arithmetic means of covariance functions with potential applications to space–time data. *Journal of Multivariate Analysis*, 100(8): 1830–1844.

Powell MJ (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In: *Advances in Optimization and Numerical Analysis*, 51–67. Springer.

Rayner NA, Auchmann R, Bessembinder J, Brönnimann S, Brugnara Y, Capponi F, et al. (2020). The EUSTACE project: Delivering global, daily information on surface air temperature. *Bulletin of the American Meteorological Society*, 101(11): E1924–E1947.

Rue H, Held L (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC.

Saad Y (2003). *Iterative Methods for Sparse Linear Systems*. SIAM.

Sampson PD, Guttorp P (1992). Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87(417): 108–119.

Simon D (2013). *Evolutionary Optimization Algorithms*. John Wiley & Sons.

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

Tong YL (2012). *The Multivariate Normal Distribution*. Springer Science & Business Media.

Williams CK, Rasmussen CE (2006). *Gaussian Processes for Machine Learning*. MIT press, Cambridge, MA.