Finite element simulations of non-Markovian random fields on Riemannian manifolds

Mike PEREIRA[1,2], Nicolas DESASSIS[1]

[1]Geostatistics team, MINES ParisTech, PSL Research University
[2]ESTIMAGES / ShoWhere

SPDE-INLA Workshop
RESSTE
November 8th, 2018

P **polynomial** taking strictly positive value on $\mathbb{R}_+$.

If Z is a continuous Markov random field field,

- its spectral density (Fourier transform of the covariance function) of the form (Rozanov, 1977):

$$g(\omega) = 1/P(\|\omega\|^2)$$

Continuous Markov random fields

P **polynomial** taking strictly positive value on $\mathbb{R}_+$.

If Z is a continuous Markov random field field,

- its spectral density (Fourier transform of the covariance function) of the form (Rozanov, 1977):

$$g(\omega) = 1/P(\|\omega\|^2)$$

- it can be seen as a solution of a stochastic partial derivative equations of the form (Rozanov, 1977, Simpson et al., 2012):

$$P(-\Delta)^{1/2} Z = \mathcal{W}$$

  where $\mathcal{W}$ is a Gaussian white noise, and $P(-\Delta)^{1/2}$ is the differential operator defined as :

$$P(-\Delta)^{1/2}[.] = \mathscr{F}^{-1}\left[\omega \mapsto \sqrt{P(\|\omega\|^2)}\mathscr{F}[.](\omega)\right]$$

  where $\mathscr{F}$ denotes the Fourier transform operator.

Simulate Markovian fields by solving numerically the SPDE:

$$P(-\Delta)^{1/2} Z = \mathcal{W} \quad (1)$$

using finite element method :

$$\boxed{Z(s) = \sum z_i \psi_i(s)}, \quad s \in \mathcal{D}$$

where $\{\psi_i\}$ are basis functions on a triangulated domain $\mathcal{D}$ (bounded polygonal or manifold), and $\{z_i\}$ are Gaussian weights.

Representation of Markov random fields

Simulate Markovian fields by solving numerically the SPDE:

$$P(-\Delta)^{1/2}Z = \mathcal{W} \tag{1}$$

using finite element method :

$$\boxed{Z(s) = \sum z_i \psi_i(s)}, \quad s \in \mathcal{D}$$

where $\{\psi_i\}$ are basis functions on a triangulated domain $\mathcal{D}$ (bounded polygonal or manifold), and $\{z_i\}$ are Gaussian weights.

Proposition : Markov random fields (Lindgren et al., 2011)

The precision matrix of the weights $\{z_i\}$ of the finite element (FE) representation of the stationary solutions of (1) is:

$$\boxed{\boldsymbol{Q}_z = \boldsymbol{C}^{1/2} P(\boldsymbol{S}) \boldsymbol{C}^{1/2}}$$

where:

$$\boldsymbol{C} = \mathrm{Diag}(\langle \psi_i, 1 \rangle), \quad \boldsymbol{G} = [\langle \nabla \psi_i, \nabla \psi_j \rangle], \quad \boldsymbol{S} = \boldsymbol{C}^{-1/2} \boldsymbol{G} \boldsymbol{C}^{-1/2}$$

$\Rightarrow$ For spectral densities of the form:

$$g(\|\omega\|) = \frac{1}{P(\|\omega\|^2)}$$

Proposition : Generalized random field (Lang and Potthoff, 2011)

A second-order stationary, isotropic Gaussian random field Z with spectral density $g : \mathbb{R}_+ \mapsto \mathbb{R}_+$ on $\mathcal{D} \subset \mathbb{R}^d$ can be expressed as:

$$Z = \mathcal{L}_{\sqrt{g}} \mathcal{W} \tag{2}$$

where $\mathcal{L}_{\sqrt{g}}[.] := \mathscr{F}^{-1}\left[\omega \mapsto \sqrt{g(\|\omega\|^2)}\mathscr{F}[.](\omega)\right]$ and $\mathcal{W}$ is a Gaussian white noise on $\mathcal{D}$.

Proposition : Generalized random field (Lang and Potthoff, 2011)

A second-order stationary, isotropic Gaussian random field Z with spectral density $g : \mathbb{R}_+ \mapsto \mathbb{R}_+$ on $\mathcal{D} \subset \mathbb{R}^d$ can be expressed as:

$$Z = \mathcal{L}_{\sqrt{g}}\mathcal{W} \tag{2}$$

where $\mathcal{L}_{\sqrt{g}}[.] := \mathscr{F}^{-1}\left[\omega \mapsto \sqrt{g(\|\omega\|^2)}\mathscr{F}[.](\omega)\right]$ and $\mathcal{W}$ is a Gaussian white noise on $\mathcal{D}$.

By representing Z using a finite element approach, we showed:

Proposition : Covariance matrix of the FE weights (Pereira and Desassis, 2018b)

The covariance matrix of the weights $\{z_i\}$ of the FE representation of (2) is:

$$\boxed{\boldsymbol{\Sigma}_z = \boldsymbol{C}^{-1/2} g(\boldsymbol{S}) \boldsymbol{C}^{-1/2}}$$

where: $\boldsymbol{C} = \text{Diag}(\langle \psi_i, 1 \rangle)$, $\boldsymbol{G} = [\langle \nabla\psi_i, \nabla\psi_j \rangle]$, $\boldsymbol{S} = \boldsymbol{C}^{-1/2}\boldsymbol{G}\boldsymbol{C}^{-1/2}$,

$$\boldsymbol{S} = \boldsymbol{V} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \boldsymbol{V}^T, \quad g(\boldsymbol{S}) = \boldsymbol{V} \begin{pmatrix} g(\lambda_1) & & \\ & \ddots & \\ & & g(\lambda_n) \end{pmatrix} \boldsymbol{V}^T$$

Theoretical framework for the proof

Same framework as in (Bolin et al., 2017).

$L^2(\mathcal{D})$ = Hilbert space of square-integrable function on $\mathcal{D}$

- The negative Laplacian $-\Delta$ is a self-adjoint positive semi-definite operator on $L^2(\mathcal{D}) \Rightarrow$ Diagonalizable :
  - Countable eigenvalues : $0 \leq \mu_1 \leq \mu_2 \leq \dots \leq \mu_j \leq \dots, \quad j \in \mathbb{N}$
  - the eigenfunctions of $-\Delta$ form an orthonormal basis of $L^2(\mathcal{D})$

Theoretical framework for the proof

Same framework as in (Bolin et al., 2017).

$L^2(\mathcal{D})$ = Hilbert space of square-integrable function on $\mathcal{D}$

- The negative Laplacian $-\Delta$ is a self-adjoint positive semi-definite operator on $L^2(\mathcal{D}) \Rightarrow$ Diagonalizable :
  - Countable eigenvalues : $0 \leq \mu_1 \leq \mu_2 \leq \cdots \leq \mu_j \leq \ldots, \quad j \in \mathbb{N}$
  - the eigenfunctions of $-\Delta$ form an orthonormal basis of $L^2(\mathcal{D})$

- It can be showed that then, the Gaussian white noise $\mathcal{D}$ can be expressed as :

$$\mathcal{W} = \sum_{j\in\mathbb{N}} \xi_j e_j$$

  for a family of i.i.d. standard Gaussian weights $\{\xi_j\}_{j\in\mathbb{N}}$

- The generalized random field Z is given by :

$$Z = \mathcal{L}_{\sqrt{g}}\mathcal{W} = \sum_{j\in\mathbb{N}} \sqrt{g(\mu_j)}\xi_j e_j$$

Theoretical framework for the proof

Same framework as in (Bolin et al., 2017).

$L^2(\mathcal{D})$ = Hilbert space of square-integrable function on $\mathcal{D}$

- The negative Laplacian $-\Delta$ is a self-adjoint positive semi-definite operator on $L^2(\mathcal{D}) \Rightarrow$ Diagonalizable :
  - Countable eigenvalues : $0 \leq \mu_1 \leq \mu_2 \leq \dots \leq \mu_j \leq \dots, \quad j \in \mathbb{N}$
  - the eigenfunctions of $-\Delta$ form an orthonormal basis of $L^2(\mathcal{D})$
- It can be showed that then, the Gaussian white noise $\mathcal{D}$ can be expressed as :

$$\mathcal{W} = \sum_{j\in\mathbb{N}} \xi_j e_j$$

  for a family of i.i.d. standard Gaussian weights $\{\xi_j\}_{j\in\mathbb{N}}$
- The generalized random field Z is given by :

$$Z = \mathcal{L}_{\sqrt{g}}\mathcal{W} = \sum_{j\in\mathbb{N}} \sqrt{g(\mu_j)}\xi_j e_j$$

- The finite element representation of Z is defined as the projection of Z onto the linear span of the basis functions $\text{span}\{\psi_k : k \in [\![1, n]\!]\} \subset L^2(\mathcal{D})$.

Generalization to Riemannian manifold I

Definition

A Riemannian manifold $\mathcal{M} = (\mathcal{D}, H)$ of dim. d is a composed by:

- a manifold $\mathcal{D}$, i.e. a domain that "behaves" locally like $\mathbb{R}^d$
- a metric H, i.e. a smooth application that associates to any $s \in \mathcal{D}$ an inner product on the tangent space of $\mathcal{D}$ at the point s.
  In particular, H can be seen as a family of positive definite matrices of size d indexed by the points of $\mathcal{D}$

The Laplacian (or Laplace-Beltrami operator) on $\mathcal{M}$ is defined by:

$$\Delta_{\mathcal{M}} f = \frac{1}{\sqrt{\det H}} \sum_{i=1}^{d} \partial_i \left[\sqrt{\det H} \sum_{j=1}^{d} [H^{-1}]_{ij} \partial_j f \right]$$

$$= \frac{1}{\sqrt{\det H}} \operatorname{div} \left(\sqrt{\det H} H^{-1} \nabla f \right)$$

$\Rightarrow$ It is a self-adjoint positive semi-definite operator on $L^2(\mathcal{M})$!

$\Rightarrow$ Generalize the previous result to fields on $\mathcal{M}$ (Pereira and Desassis, 2018b)

Generalization to Riemannian manifold II

Let $\mathcal{M} = (\mathcal{D}, H)$ be a Riemannian manifold and let $g : \mathbb{R}_+ \mapsto \mathbb{R}_+$. Let Z be the generalized random field defined by:

$$Z = \mathcal{L}_{\sqrt{g}}\mathcal{W} := \sum_{j\in\mathbb{N}} \sqrt{g(\mu_j)}\xi_j e_j \tag{3}$$

where:

- $\{(\mu_j, e_j) : j \in \mathbb{N}\}$ are eigenpairs of the negative Laplacian $-\Delta_{\mathcal{M}}$, forming an orthonormal basis of $L^2(\mathcal{M})$
- $\{\xi_j\}_{j\in\mathbb{N}}$ is a set of i.i.d. standard Gaussian weights

Proposition : Covariance matrix of the FE weights (Pereira and Desassis, 2018b)

The covariance matrix of the weights $\{z_i\}$ of the FE representation of (3) is:

$$\boxed{\Sigma_z = \boldsymbol{C}^{-1/2} g(\boldsymbol{S}) \boldsymbol{C}^{-1/2}}$$

where:

$$\boldsymbol{C} = \mathrm{Diag}(\langle\sqrt{\det H}\psi_i, 1\rangle), \quad \boldsymbol{G} = [\langle\nabla\psi_i, \sqrt{\det H}H^{-1}\nabla\psi_j\rangle]$$

$$\boldsymbol{S} = \boldsymbol{C}^{-1/2}\boldsymbol{G}\boldsymbol{C}^{-1/2}$$

General form of the covariance matrix of finite element representations of Gaussian fields :

$$\boxed{\Sigma_z = \boldsymbol{C}^{-1/2} g(\boldsymbol{S}) \boldsymbol{C}^{-1/2}} \quad (4)$$

where

- $\boldsymbol{C}$ is a diagonal matrix with strictly positive elements.
- $\boldsymbol{S}$ is a symmetric positive semi-definite matrix whose elements are inner products of gradients of the basis functions.

$\Rightarrow$ How to simulate weights with covariance matrix (4)?

Proposition : Simulation of SPDE FEM solutions

Weights $\boldsymbol{z} = (z_1, \dots, z_n)^T$ with covariance matrix Σ_z given by (4) can be simulated through:

$$\boxed{\boldsymbol{z} = \boldsymbol{C}^{-1/2} \sqrt{g}(\boldsymbol{S}) \varepsilon}$$

where ε is a Gaussian vector with independent standard components and $\sqrt{g} : \mathbb{R}_+ \mapsto \mathbb{R}$ satisfies $\left(\sqrt{g}\right)^2 = g$.

Problem

How to compute $\sqrt{g}(\boldsymbol{S})\varepsilon$?

$$\boldsymbol{S} = \boldsymbol{V}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}\boldsymbol{V}^T \Rightarrow \sqrt{g}(\boldsymbol{S})\varepsilon = \underline{\underline{\boldsymbol{V}\begin{pmatrix}\sqrt{g}(\lambda_1) & & \\ & \ddots & \\ & & \sqrt{g}(\lambda_n)\end{pmatrix}\underline{\boldsymbol{V}^T\varepsilon}}}$$

$\Rightarrow$ Diagonalization + Storage : Expensive!!

Idea : use the polynomial case : for $P(X) = \sum a_k X^k$

Weight simulation

Problem

How to compute $\sqrt{g}(\boldsymbol{S})\varepsilon$?

$$\boldsymbol{S} = \boldsymbol{V}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}\boldsymbol{V}^T \Rightarrow \sqrt{g}(\boldsymbol{S})\varepsilon = \underline{\underline{\boldsymbol{V}\underline{\begin{pmatrix}\sqrt{g}(\lambda_1) & & \\ & \ddots & \\ & & \sqrt{g}(\lambda_n)\end{pmatrix}\underline{\boldsymbol{V}^T\varepsilon}}}}$$

$\Rightarrow$ Diagonalization + Storage : Expensive!!

Idea : use the polynomial case : for $P(X) = \sum a_k X^k$

$$P(\boldsymbol{S})\varepsilon = \boldsymbol{V}\begin{pmatrix}P(\lambda_1) & & \\ & \ddots & \\ & & P(\lambda_n)\end{pmatrix}\boldsymbol{V}^T\varepsilon = \sum a_k \boldsymbol{S}^k\varepsilon$$

$\rightarrow$ $P(\boldsymbol{S})\varepsilon$ is computable iteratively without having to diagonalize $\boldsymbol{S}$: only involves matrix-vector multiplications!

Problem

How to compute $\sqrt{g}(\boldsymbol{S})\varepsilon$?

$$\boldsymbol{S} = \boldsymbol{V}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}\boldsymbol{V}^T \Rightarrow \sqrt{g}(\boldsymbol{S})\varepsilon = \underline{\underline{\boldsymbol{V}\begin{pmatrix}\sqrt{g}(\lambda_1) & & \\ & \ddots & \\ & & \sqrt{g}(\lambda_n)\end{pmatrix}\underline{\boldsymbol{V}^T\varepsilon}}}$$

$\Rightarrow$ Diagonalization + Storage : Expensive!!

Idea : use the polynomial case : for $P(X) = \sum a_k X^k$

$$P(\boldsymbol{S})\varepsilon = \boldsymbol{V}\begin{pmatrix}P(\lambda_1) & & \\ & \ddots & \\ & & P(\lambda_n)\end{pmatrix}\boldsymbol{V}^T\varepsilon = \sum a_k \boldsymbol{S}^k\varepsilon$$

$\rightarrow$ $P(\boldsymbol{S})\varepsilon$ is computable iteratively without having to diagonalize $\boldsymbol{S}$: only involves matrix-vector multiplications!

Problem

How to compute $\sqrt{g}(\boldsymbol{S})\varepsilon$?

$$\boldsymbol{S} = \boldsymbol{V}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}\boldsymbol{V}^T \Rightarrow \sqrt{g}(\boldsymbol{S})\varepsilon = \underline{\underline{\boldsymbol{V}\underline{\begin{pmatrix}\sqrt{g}(\lambda_1) & & \\ & \ddots & \\ & & \sqrt{g}(\lambda_n)\end{pmatrix}}\underline{\boldsymbol{V}^T\varepsilon}}}$$

$\Rightarrow$ Diagonalization + Storage : Expensive!!

Idea : use the polynomial case : for $P(X) = \sum a_k X^k$

$$P(\boldsymbol{S})\varepsilon = \boldsymbol{V}\begin{pmatrix}P(\lambda_1) & & \\ & \ddots & \\ & & P(\lambda_n)\end{pmatrix}\boldsymbol{V}^T\varepsilon = \sum a_k \boldsymbol{S}^k\varepsilon$$

$\rightarrow$ $P(\boldsymbol{S})\varepsilon$ is computable iteratively without having to diagonalize $\boldsymbol{S}$: only involves matrix-vector multiplications!

$\rightarrow$ Instead of computing $\sqrt{g}(\boldsymbol{S})\varepsilon$, compute $P(\boldsymbol{S})\varepsilon$ where P is an approximation of $\sqrt{g}$ over an interval containing $\{\lambda_1, \dots, \lambda_n\}$

$\Rightarrow P(\boldsymbol{S})\varepsilon \approx \sqrt{g}(\boldsymbol{S})\varepsilon$ because $P(\lambda_i) \approx \sqrt{g}(\lambda_i) \quad \forall i$

Chebyshev algorithm for weight simulation

Algorithm : Chebyshev simulation (Pereira and Desassis, 2018a)

Require: An order of approximation $K \in \mathbb{N}$.
Output: A vector $\boldsymbol{z} \approx \boldsymbol{C}^{-1/2}\sqrt{g}(\boldsymbol{S})\varepsilon$.

1. Compute an interval $[a, b]$ containing all the eigenvalues of $\boldsymbol{S}$
   - Ex: $[0, \sqrt{\text{Trace}(\boldsymbol{S}\boldsymbol{S}^T)}]$, Gershgorin circle theorem
2. Compute an approximation (denoted P) of $\sqrt{g}$ over $[a, b]$ by truncating its Chebyshev series at order K
   $\rightarrow$ Coefficients of the decomposition in Chebyshev basis obtained by FFT
3. Compute $\boldsymbol{u} = P(\boldsymbol{S})\varepsilon$ iteratively (only requires matrix to vector multiplications)
4. Return $\boldsymbol{z} = \boldsymbol{C}^{-1/2}\boldsymbol{u}$

- Computational complexity: $\mathcal{O}(Kn_{nz})$ operations, n_{nz} number of non-zero entries of $\boldsymbol{S}$
- Question : *How to choose the order of approximation K to get a "satisfying" output?*

Initial Goal

Simulate a zero-mean Gaussian vector with covariance matrix:

$$\Sigma = \boldsymbol{C}^{-1/2} g(\boldsymbol{S}) \boldsymbol{C}^{-1/2}$$

Output of the algorithm

A zero-mean Gaussian vector

$$\boldsymbol{z}_s = \boldsymbol{C}^{-1/2} P_K(\boldsymbol{S}) \varepsilon$$

with covariance matrix:

$$\Sigma_s = \boldsymbol{C}^{-1/2} P_K^2(\boldsymbol{S}) \boldsymbol{C}^{-1/2}$$

$\rightarrow$ *When can the simulated output "pass" as the targeted one?*

An approached output

Initial Goal

Simulate a zero-mean Gaussian vector with covariance matrix:

$$\Sigma = \boldsymbol{C}^{-1/2} g(\boldsymbol{S}) \boldsymbol{C}^{-1/2}$$

Output of the algorithm

A zero-mean Gaussian vector

$$\boldsymbol{z}_s = \boldsymbol{C}^{-1/2} P_K(\boldsymbol{S}) \varepsilon$$

with covariance matrix:

$$\Sigma_s = \boldsymbol{C}^{-1/2} P_K^2(\boldsymbol{S}) \boldsymbol{C}^{-1/2}$$

$\rightarrow$ *When can the simulated output "pass" as the targeted one?*

Idea: Use statistical tests on the output

- Let $\{\boldsymbol{z}_s^{(1)}, ..., \boldsymbol{z}_s^{(N)}\}$ be a N-sample of vectors simulated using the algorithm and $\boldsymbol{c} = (c_1, ..., c_n)^T \in \mathbb{R}^n$ be arbitrary coefficients.
- If the $\boldsymbol{z}_s^{(i)}$ have covariance matrix Σ, then
$$\mathcal{S}(\boldsymbol{c}) = \{\boldsymbol{c}^T \boldsymbol{z}_s^{(1)}, ..., \boldsymbol{c}^T \boldsymbol{z}_s^{(N)}\}$$
is a Gaussian sample with variance $\boldsymbol{c}^T \Sigma \boldsymbol{c}$.
  $\Rightarrow$ Use χ^2 test of variance on $\mathcal{S}(\boldsymbol{c})$ to check that.
- Given that the actual distribution of $\mathcal{S}(\boldsymbol{c})$ is known (Gaussian with variance $\boldsymbol{c}^T \Sigma_s \boldsymbol{c}$), **we can anticipate the results without actually realising any test**!

Criterion on the approximation precision

Proposition : Statistical and approximation errors (Pereira and Desassis, 2018a)

Let $R_{\text{reject}}(\boldsymbol{c})$ be the probability that a χ^2 test with significance α on the N-sample $\mathcal{S}(\boldsymbol{c})$ "fails" (i.e. null hypothesis rejected).

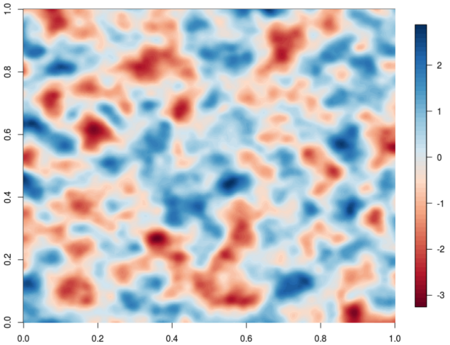
Then, $\forall \beta > 0$, $\exists \epsilon_\beta > 0$ such that :

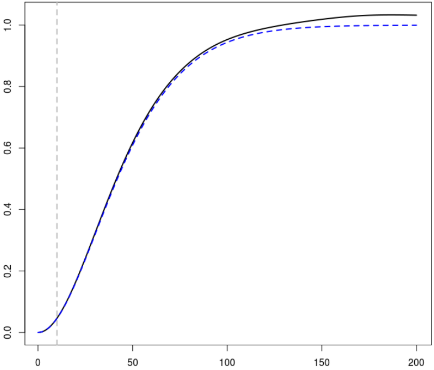
$$\underbrace{\max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} \left| \frac{g(\lambda) - P_K(\lambda)^2}{P_K(\lambda)^2} \right|}_{\text{Error of the polynomial approximation}} \leq \epsilon_\beta \Rightarrow \forall \boldsymbol{c}, R_{\text{reject}}(\boldsymbol{c}) \leq (1+\beta)\alpha$$

ϵ_β can be numerically computed and depends on α, N and $\beta > 0$.

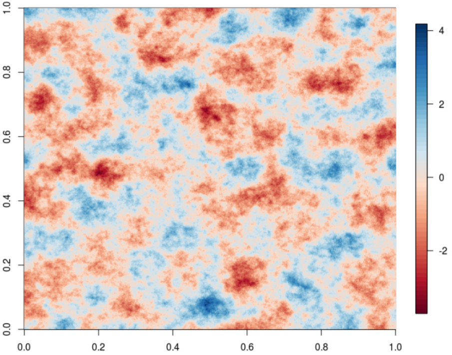
β	Sample Size N				
	50	100	1000	5000	10000
0,1%	6,40e-04	6,20e-04	4,80e-04	3,00e-04	2,40e-04
1%	5,44e-03	4,80e-03	2,36e-03	1,20e-03	8,60e-04
5%	1,89e-02	1,51e-02	5,94e-03	2,82e-03	2,02e-03
10%	3,00e-02	2,33e-02	8,64e-03	4,02e-03	2,88e-03
50%	7,66e-02	5,71e-02	1,98e-02	9,08e-03	6,46e-03
100%	1,10e-01	8,12e-02	2,80e-02	1,28e-02	9,10e-03

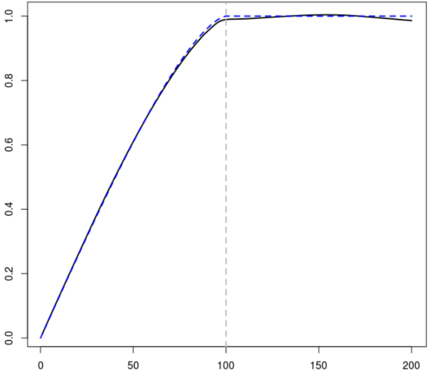
Table: Examples of $\epsilon_{N,\beta}$ values of for $\alpha = 0.05$





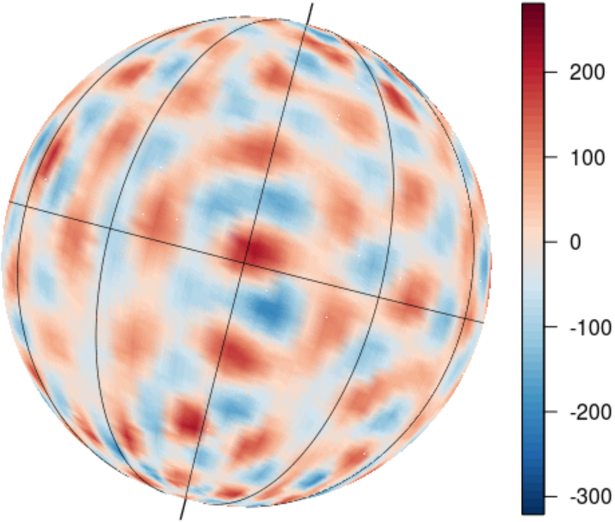
Simulation on 1000x1000 grid $Z = \mathcal{L}_{\sqrt{g}}\mathcal{W}$ with $g(\|\omega\|^2) = \left(\kappa^2 + \|\omega\|^2\right)^{-(\pi+1)}$



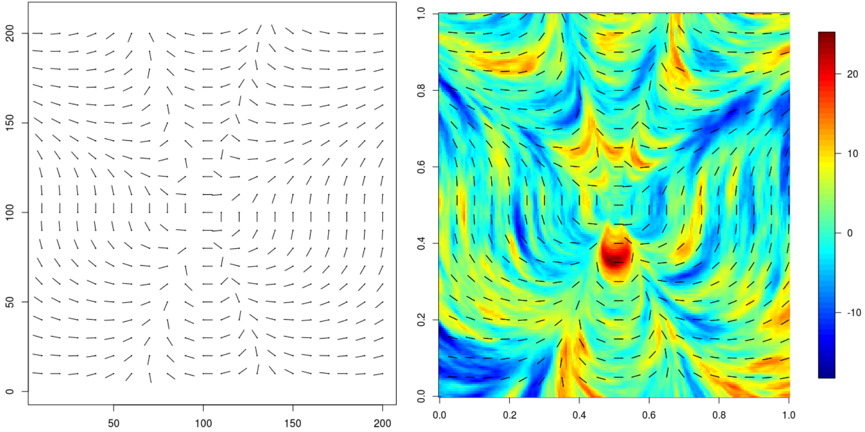


Simulation on 1000x1000 grid $Z = \mathcal{L}_{\sqrt{g}}\mathcal{W}$ with g is the spectral density o the Hyperspherical covariance.

Application : Simulation on the sphere



Simulation on a triangulated sphere of the field $Z = \mathcal{L}_g \mathcal{W}$ with $g(\|\omega\|^2) = \left(\kappa^4 + 2\kappa^2 \cos(2\pi\theta)\|\omega\|^2 + \|\omega\|^4\right)^{-2}$



Simulation on a Riemannian manifold of $Z = \mathcal{L}_g \mathcal{W}$ with $g(\|\omega\|^2) = \left(\kappa^2 + \|\omega\|^2\right)^{-1}$. The metric tensor is given by :

$$H^{-1}(s) = R(s)^T R(s), \quad R(s) = \begin{pmatrix} d_1(s) & 0 \\ 0 & d_2(s) \end{pmatrix} \begin{pmatrix} \cos(\theta(s)) & -\sin(\theta(s)) \\ \sin(\theta(s)) & \cos(\theta(s)) \end{pmatrix}$$

Conclusion

- Large class of Gaussian random fields : Characterized by their spectral density
- Large class of domains : manifolds and Riemannian manifolds
- Explicit expression of the covariance matrix of FE weights
- Efficient approximate algorithm for the computation of samples of weights : linear complexity
- Approximation error tolerance set to retrieve statistical properties

References

Bolin, D., Kirchner, K., and Kovács, M. (2017). Numerical solution of fractional elliptic stochastic pdes with spatial white noise. *arXiv preprint arXiv:1705.06565*.

Lang, A. and Potthoff, J. (2011). Fast simulation of gaussian random fields. *Monte Carlo Methods and Applications*, 17(3):195–214.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields 670 and gaussian markov random fields: the spde approach (with discussion). *JR 671 Stat Soc, Series B*, 73:423–498.

Pereira, M. and Desassis, N. (2018a). Efficient simulation of Gaussian Markov random fields by Chebyshev polynomial approximation. *arXiv:1805.07423*.

Pereira, M. and Desassis, N. (2018b). Finite element approximation of non-Markovian random fields. *arXiv:1811.03004*.

Rozanov, J. A. (1977). Markov random fields and stochastic partial differential equations. *Mathematics of the USSR-Sbornik*, 32(4):515.

Simpson, D., Lindgren, F., and Rue, H. (2012). Think continuous: Markovian gaussian models in spatial statistics. *Spatial Statistics*, 1:16 – 29.

Thank you for your attention!
Questions?

Suppose that $g = g_{\boldsymbol{\theta}}$ depends on a vector of parameters $\boldsymbol{\theta}$.
The log-likelihood associated to $\boldsymbol{z}$ and $\boldsymbol{\theta}$ is given by :

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\theta}) = -\frac{1}{2}\left(N \log 2\pi + \log \det\left(g_{\boldsymbol{\theta}}(\boldsymbol{S})\right) + \boldsymbol{z}^T \boldsymbol{C}^{1/2} g_{\boldsymbol{\theta}}(\boldsymbol{S})^{-1} \boldsymbol{C}^{1/2} \boldsymbol{z}\right)$$

It can be expensive to compute/maximize...

Likelihood-based method for model inference II

- Inverse approximation

$$\boldsymbol{z}^T \boldsymbol{C}^{1/2} g_{\boldsymbol{\theta}}(\boldsymbol{S})^{-1} \boldsymbol{C}^{1/2} \boldsymbol{z} = \boldsymbol{z}^T \boldsymbol{C}^{1/2} \boldsymbol{V} \begin{pmatrix} 1/g_{\boldsymbol{\theta}}(\lambda_1) & & \\ & \ddots & \\ & & 1/g_{\boldsymbol{\theta}}(\lambda_n) \end{pmatrix} \boldsymbol{V}^T \boldsymbol{C}^{1/2} \boldsymbol{z}$$

$$= \boldsymbol{z}^T \boldsymbol{C}^{1/2} \frac{1}{g_{\boldsymbol{\theta}}}(\boldsymbol{S}) \boldsymbol{C}^{1/2} \boldsymbol{z}$$

$\Rightarrow$ Use polynomial approximation of $\frac{1}{g_{\boldsymbol{\theta}}}$

- Inverse approximation

$$z^T C^{1/2} g_{\theta}(S)^{-1} C^{1/2} z = z^T C^{1/2} V \begin{pmatrix} 1/g_{\theta}(\lambda_1) & & \\ & \ddots & \\ & & 1/g_{\theta}(\lambda_n) \end{pmatrix} V^T C^{1/2} z$$

$$= z^T C^{1/2} \frac{1}{g_{\theta}}(S) C^{1/2} z$$

$\Rightarrow$ Use polynomial approximation of $\frac{1}{g_{\theta}}$

- Determinant approximation

$$\log\det\left(g_{\theta}(S)\right) = \sum_{k=1}^{n} \log(g_{\theta}(\lambda_k)) \approx \sum_{m=0}^{M} \text{hist}(a_m) \log(g_{\theta}(a_m))$$

where

$$\text{hist}(a_m) := \text{Card}\left\{ i \in [\![0, N-1]\!] : \lambda_i \in]a_m - \frac{\tau}{2}, a_m + \frac{\tau}{2}] \right\}$$

$$= \mathbb{E}\left(||\mathbf{1}_{]a_m - \frac{\tau}{2}, a_m + \frac{\tau}{2}]}(\mathbf{S})\varepsilon||^2 \right)$$