



HAL
open science

How to initialize the Circulant Embedding method to speed up the generation of stationary Gaussian Random Fields?

Géraldine Pichot, Simon Legrand, Michel Kern, Nathanael
Tepakbong-Tematio

► **To cite this version:**

Géraldine Pichot, Simon Legrand, Michel Kern, Nathanael Tepakbong-Tematio. How to initialize the Circulant Embedding method to speed up the generation of stationary Gaussian Random Fields?. 2021. hal-03190252v2

HAL Id: hal-03190252

<https://inria.hal.science/hal-03190252v2>

Preprint submitted on 24 Sep 2021 (v2), last revised 15 Nov 2022 (v6)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithms to speed up the generation of stationary Gaussian Random Fields with the Circulant Embedding method

Simon Legrand¹, Géraldine Pichot ^{*,1,2}, and Nathanael Tepakbong-Tematio³

¹Inria, 2 rue Simone Iff, 75589 Paris, France

²Université Paris-Est, CERMICS (ENPC), 6 et 8 av. Blaise Pascal, 77455 Marne-la-Vallée Cedex 2, France

³ISAE-SUPAERO, 10, avenue Édouard-Belin, BP 54032, 31055 Toulouse Cedex 4, France

Abstract

The Circulant Embedding Method (CEM) is a well known technique to generate stationary Gaussian Random Fields (GRF). The main idea is to embed the covariance matrix in a larger nested block circulant matrix, whose factorization can be rapidly computed thanks to the fast FFT algorithm. The CEM requires that the extended matrix is at least positive semidefinite which is proved to be the case if the enclosing domain is sufficiently large. In this paper, we study the Matérn family of covariances and we propose algorithms for this family of covariances based on fitting functions to compute good estimates of the required enclosing domain size for the CEM algorithm to work. The fitting functions are inspired by the theoretical work from [Graham *et al*, SIAM Journal on Numerical Analysis, 2018] and the function parameters estimations are done with numerical simulations. Several numerical tests are performed to show the efficiency of the proposed algorithms.

1 Introduction

Stationary Gaussian Random Fields (GRF) are classically used to model physical properties in environment applications. For example, in hydrogeology, stationary GRF are used to model the permeability of a porous domain [7, 2]. Here we focus on the generation of stationary GRF on regular grids. Note when these fields are used in general finite elements framework, the sampling grid does not need to be identical to the finite elements grid thanks to quadrature rules [7]. If the grid has M points, the general algorithm to generate a realization of the stationary random vector \mathbf{Y} of M normal variables, with zero mean and the $M \times M$ covariance matrix \mathbf{R} , is based on a factorization of the covariance matrix \mathbf{R} as shown by Algorithm 1.

Using Algorithm 1, \mathbf{Y} has \mathbf{R} as covariance matrix:

$$\mathbb{E}[\mathbf{Y}\mathbf{Y}^T] = \mathbb{E}[(\mathbf{B}\boldsymbol{\theta})(\mathbf{B}\boldsymbol{\theta})^T] = \mathbf{B}\mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T]\mathbf{B}^T = \mathbf{B}\mathbf{B}^T = \mathbf{R}.$$

*Corresponding author: Geraldine.Pichot@inria.fr

Algorithm 1 General algorithm

- 1: Factorize $\mathbf{R} = \mathbf{B}\mathbf{B}^T$ with \mathbf{B} of size $M \times M$;
 - 2: Generate a vector $\boldsymbol{\theta}$ of size M of realizations of standard normal random variables with zero mean and uncorrelated: $\mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^T] = Id$,
 - 3: One realization is obtained by computing $\mathbf{Y} = \mathbf{B}\boldsymbol{\theta}$.
-

Algorithm 1 requires the factorization of the covariance matrix \mathbf{R} which can be done according to different methods, a review of those methods can be found in [10]. The direct one is based on the Cholesky factorization of matrix \mathbf{R} . However it demands $O(M^3)$ operations to factorize a matrix of size M , which becomes prohibitive for large M . Among the other techniques, let us cite the Karhunen-Loève (K.-L.) expansion [12]. The K.-L. expansion is infinite, so in practice the expansion is truncated to obtain an approximation of a GRF. The number of terms to keep depends on the parameters. Another method based on H-matrix has recently been developed in [5]. It has the advantage to also handle non-stationary covariance functions. It gives approximations of random fields at given points, also possibly distributed on irregular grids. Another method is the Circulant Embedding Method (CEM) which produces random vectors with exactly the required correlation structure on regular grids [4, 3, 8]. The principle of CEM is to embed the matrix \mathbf{R} of size M in a bigger $s \times s$ nested block circulant matrix \mathbf{R}^{ext} , whose factorization can be rapidly computed thanks to the fast FFT algorithm in $O(s \log(s))$ operations. The matrix \mathbf{R} is positive semidefinite and also a nested block symmetric Toeplitz matrix under an appropriate ordering of the indices. However the extension \mathbf{R}^{ext} might not be positive semidefinite unless the extended domain is large enough [8].

In this work, we are interested in the CEM and especially in efficiently computing the size of the enlarged domain to ensure that the extended matrix \mathbf{R}^{ext} is at least positive semidefinite. The following section 2 presents the notations used in this paper together with the classical CEM algorithm and the theoretical results from [8]. Section 3 gives the new algorithms we propose to speed up the computations of the classical CEM algorithm. The main idea is to cook up fitting functions, inspired by the theoretical work from [8], in order to compute a good estimate of the required size of the enlarged domain. Section 4 explains how the parameters of the fitting functions are estimated from numerical simulations. Section 5 emphasizes the quality of the estimates obtained with our new algorithms on several isotropic and anisotropic covariance test cases.

All the simulations in this paper are done with the parallel C++17 `ParaCirce` library that implements both the classical and the new CEM algorithms presented in this paper. `ParaCirce` performs the discrete Fourier transforms with the `FFTW3` library [6] and uses `Rngstream` [9] for the generation of pseudo-random numbers. `ParaCirce` handles 64-bit and 80-bit floating point precision with template arguments. The simulations are done in 80-bit floating point precision in order to avoid, as much as possible, the stagnation of the minimum eigenvalue as explained in Appendix A. `ParaCirce` parallelization is based on the MPI standard.

2 The CEM method

2.1 Notations

Consider regular grids of size $L_1 \times \dots \times L_d$ (d is the dimension) with given number of points $m_{0,i}$ in each direction i , $i = 1, \dots, d$. The objective is to generate a stationary random vector \mathbf{Y} on this uniform grid with

$$M = (m_{0,1} + 1) \times \dots \times (m_{0,d} + 1)$$

points. The grid spacing is denoted $h_{0,i} = \frac{L_i}{m_{0,i}}$ along each direction i and the grid points are $\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_M$.

Let us consider the discrete representation of \mathbf{Y} as $[Y_i]_{i=1}^M$. The covariance matrix \mathbf{R} is

$$\mathbf{R} = [\rho(\mathbf{x}_i - \mathbf{x}_j)]_{i,j=1}^M = \mathbb{E}[\mathbf{Y}_i \mathbf{Y}_j]_{i,j=1}^M$$

with $\rho : \mathbb{R}^d \rightarrow \mathbb{R}$ the covariance function.

2.2 Matérn family of covariances

A common family of covariance functions are the Matérn family of covariances. The standard anisotropic Matérn family of covariances functions, with the correlation lengths $\boldsymbol{\lambda} = (\lambda_i)_{i=1, \dots, d}$, writes in 2D, for $\mathbf{x} = (x, y)$,

$$\rho_{2D}(\mathbf{x}, \nu, \boldsymbol{\lambda}) = \kappa\left(\frac{\sqrt{\lambda_2^2 x^2 + \lambda_1^2 y^2}}{\lambda_1 \lambda_2}, \nu\right), \quad (1)$$

and in 3D, for $\mathbf{x} = (x, y, z)$,

$$\rho_{3D}(\mathbf{x}, \nu, \boldsymbol{\lambda}) = \kappa\left(\frac{\sqrt{(\lambda_2 \lambda_3 x)^2 + (\lambda_1 \lambda_3 y)^2 + (\lambda_2 \lambda_1 z)^2}}{\lambda_1 \lambda_2 \lambda_3}, \nu\right), \quad (2)$$

where

$$\kappa(r, \nu) = \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} r)^\nu K_\nu(\sqrt{2\nu} r), \quad (3)$$

with Γ the gamma function and K_ν is the modified Bessel function of the second kind, λ_i is the correlation length in direction i , $i = 1, \dots, d$, and $\nu > 0$ is a given smoothness parameter. The case $\nu = 1/2$ corresponds to the (non separable) exponential covariance and $\nu = \infty$ to the Gaussian covariance.

In the isotropic case, $\lambda_i = \lambda, \forall i = 1, \dots, d$ and the standard isotropic Matérn family of covariances writes

$$\rho(\mathbf{x}, \nu, \lambda) = \kappa\left(\frac{\|\mathbf{x}\|_2}{\lambda}, \nu\right). \quad (4)$$

Remark 2.1 If \mathbf{Y} is a second order stationary Gaussian field with mean 0 and covariance function ρ , $\mu + \sigma \mathbf{Y}$ is a second order stationary Gaussian field with mean μ and covariance function $\sigma^2 \rho$.

Examples of 2D fields with isotropic Matérn covariances are shown on Figure 1.

Examples of 2D fields with anisotropic Matérn covariances are shown on Figure 2.

Examples of 2D fields with isotropic and anisotropic Gaussian covariances are shown on Figure 3.

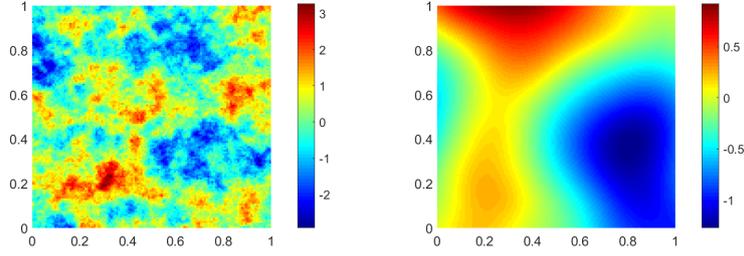


Figure 1: 2D case ($m_{0,1} = m_{0,2} = 128$, $L_1 = L_2 = 1$) - isotropic Matérn covariances (left) $\lambda = 0.125$, $\nu = 0.5$ (right) $\lambda = 0.5$, $\nu = 4$.

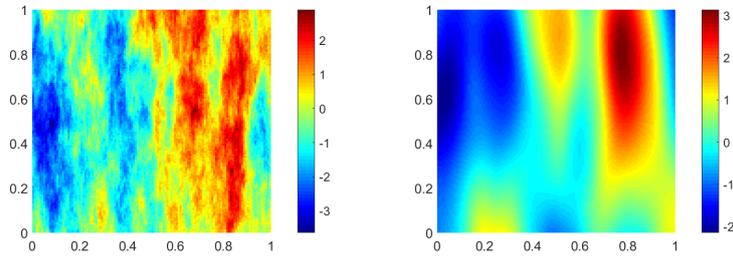


Figure 2: 2D case ($m_{0,1} = m_{0,2} = 128$, $L_1 = L_2 = 1$) - anisotropic Matérn covariances $\lambda_1 = 0.125$, $\lambda_2 = 0.5$ (left) $\nu = 0.5$ (right) $\nu = 4$

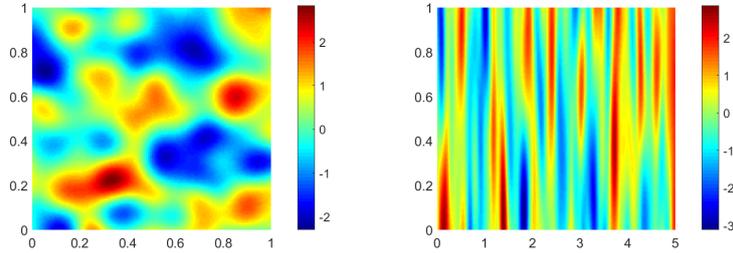


Figure 3: 2D case - Gaussian covariance (left) $m_{0,1} = m_{0,2} = 128$, $L_1 = L_2 = 1$, isotropic: $\lambda = 0.125$ (right) $m_{0,1} = 256$, $m_{0,2} = 128$, $L_1 = 5$, $L_2 = 1$, anisotropic: $\lambda_1 = 0.125$, $\lambda_2 = 0.5$.

2.3 Classical simulation algorithm

The principle of CEM is to embed the matrix \mathbf{R} in a bigger $s \times s$ symmetric nested block circulant matrix \mathbf{R}^{ext} with $s = (2m_1) \times \dots \times (2m_d)$, so that its factorization can be rapidly computed with the fast FFT algorithm. The circulant structure is obtained by a mirroring procedure of \mathbf{R} as explained in details for example in [7, 8]. In practice, with the CEM, we never need to build the full nested block circulant matrix \mathbf{R}^{ext} , we only need its first column, denoted \mathbf{r} . If one is interested by building the full matrix \mathbf{R}^{ext} , details are given for example page 77 in [3] or in [8]. The sizes m_i (or equivalently the enlarged domain of size $\ell_i := m_i \times h_{0,i}$), $i = 1, \dots, d$, must be chosen so as to guarantee that the matrix \mathbf{R}^{ext} is

at least positive semidefinite, otherwise the algorithm can not work due to the presence of negative eigenvalues. Here, we call the padding the extension between $m_{0,i}$ and m_i on which the covariance function is also evaluated [7, 11]. Note the padding may be different in each direction according to the different values of L_i , ν , λ_i and $m_{0,i}$, $i = 1, \dots, d$.

Therefore the CEM method relies on two steps:

- Step 1. the computation of the size of the extended domain to guarantee that \mathbf{R}^{ext} is at least positive semidefinite;
- Step 2. the sampling of instances of the random field with the discrete Fourier transform.

These two steps can be performed according to the following Algorithm 2 and Algorithm 3 modified from [8] according to our notations.

Algorithm 2 Step 1. Compute the size of the extended domain to guarantee that \mathbf{R}^{ext} is at least positive semidefinite.

Data: d , $m_{0,i}$ ($i = 1, \dots, d$), and covariance function ρ .

Result: Number of points m_i , $i = 1, \dots, d$, to guarantee that \mathbf{R}^{ext} is at least positive semidefinite and the vector of eigenvalues \mathbf{v} .

1. Set $m_i = m_{0,i}$, $i = 1, \dots, d$.
 2. Calculate \mathbf{r} , the first column of \mathbf{R}^{ext} .
 3. Calculate \mathbf{v} , the vector of eigenvalues of \mathbf{R}^{ext} , by d -dimensional FFT on \mathbf{r} .
 4. If smallest eigenvalue < 0 then increment m_i and go to Step 2.
-

Algorithm 3 Step 2. Sample two independent instances \mathbf{z}_1 and \mathbf{z}_2 of the random field.

Data: d , $m_{0,i}$, and m_i and \mathbf{v} obtained by Algorithm 1.

Result: Two instances \mathbf{z}_1 and \mathbf{z}_2 of the random field.

1. With $s = (2m_1) \times \dots \times (2m_d)$, sample two s -dimensional normal random vector \mathbf{y}_{re} and \mathbf{y}_{im} : $\mathbf{y} = \mathbf{y}_{re} + i\mathbf{y}_{im}$
 2. Update \mathbf{y}_{re} and \mathbf{y}_{im} by elementwise multiplication with \sqrt{v} .
 3. Set $\mathbf{w}_{re} + i\mathbf{w}_{im}$ to be the d -dimensional iFFT of \mathbf{y} .
 4. Obtain two independent instances \mathbf{z}_1 and \mathbf{z}_2 by extracting the appropriate $M = (m_{0,1} + 1) \times \dots \times (m_{0,d} + 1)$ entries of \mathbf{w}_{re} and \mathbf{w}_{im} respectively.
-

Remark 2.2 Algorithm 3 yields two independent realizations of the random field [4]. Lemma 5 in [7] and Algorithm 2 in [8] explain how to sample only one field.

2.4 Theoretical estimation of the required domain size

In Algorithm 2, the number of iterations at Step 4 (that is the number of times m is incremented) depends on the values of $m_{0,i}$, L_i , λ_i , ν and d . Each iteration costs the computation of \mathbf{r} , the first column of \mathbf{R}^{ext} , and a d -dimensional FFT.

In [8], the authors studied the isotropic Matérn family of covariances with $1/2 \leq \nu < \infty$ and the isotropic Gaussian covariances ($\nu = \infty$) on a unit cube domain ($L_i = 1$, $m_{0,i} = m_0$, $h_{0,i} = h_0 = 1/m_0$, $m_i = m$, $\forall i$). They proved that, under quite general conditions, if m (or equivalently $\ell := m h_0$) is chosen large enough, the extension \mathbf{R}^{ext} is necessarily positive definite. The two main Theorems from [8] are recalled below.

2.4.1 Theorem for the isotropic Matérn family of covariances with $1/2 \leq \nu < \infty$

Theorem 2.9 in [8] provides the conditions for the isotropic Matérn family of covariances ($1/2 \leq \nu < \infty$):

Theorem 2.3 (From [8], Theorem 2.9) *According to [8], for a domain $[0; L]^d$ with $L = 1$, consider the isotropic Matérn covariance family, with smoothness parameter ν satisfying $\frac{1}{2} \leq \nu < \infty$ and the correlation length $\lambda \leq 1$. Suppose $h_0/\lambda \leq e^{-1}$. Then there exists positive constants c_1 and $c_2 \geq 2\sqrt{2}$ which may depend on the dimension d but are independent of the other parameters ℓ, h_0, λ, ν such that \mathbf{R}^{ext} is positive definite if:*

$$\ell/\lambda \geq c_1 + c_2 \nu^{0.5} \log(\max(\lambda/h_0, \nu^{0.5})) \quad (5)$$

The extension of Theorem 2.3 to $\nu \in (0; 1/2)$ remains an open question. In practical applications, $\nu \geq 1/2$ [8].

2.4.2 Theorem for the isotropic Gaussian covariances ($\nu = \infty$)

Theorem 2.11 in [8] provides the conditions on the required domain size for the Gaussian covariance ($\nu = \infty$):

Theorem 2.4 (From [8], Theorem 2.11) *For a domain $[0; L]^d$ with $L = 1$ and the Gaussian covariance (i.e., the isotropic Matérn kernel with $\nu = \infty$), there exists a constant B (depending only on spatial dimension d) such that positive definiteness of \mathbf{R}^{ext} is guaranteed when*

$$\ell \geq 1 + \lambda \max\{\sqrt{2} \frac{\lambda}{h_0}, B\} \quad (6)$$

The constant B depends on the dimension d and we found it to be $B_{2D} = 5.57$ in 2D and $B_{3D} = 7.515$ in 3D.

Remark 2.5 *Thanks to these theorems, the authors of [8] prove that, provided the domain is sufficiently enlarged, Algorithm 2 theoretically always terminates. In practice, for some test cases, stagnation of the smallest eigenvalue is observed (see Appendix A). Increasing floating point precision of the code helps in solving this numerical issue.*

3 New algorithms

The novelty of our work is to use the two Theorems 2.3 and 2.4 to cook up new algorithms to speed up Algorithm 2. The main idea is not to start from $m_{0,i}$ as in Step 1. of Algorithm 2 but rather to start at a good first estimate of the enlarged domain obtained with these two theorems and numerical simulations.

3.1 Fitting functions

Theorem 2.3 encourages us to define the function \mathcal{M} for the Matérn family of covariances with $1/2 \leq \nu < \infty$, as:

$$\mathcal{M}(\nu, v) = c_1^{est} + c_2^{est} \nu^{0.5} \log_{10}(\max(v, \nu^{0.5})), \quad (7)$$

with c_1^{est} and c_2^{est} to be determined with numerical simulations.

We will consider the function $\mathcal{M}(\nu, \lambda/h_0)$ in the isotropic covariances cases and $\mathcal{M}(\nu, \lambda_i/h_{0,i})$ in the anisotropic cases.

Theorem 2.4 together with our numerical simulations encourages us to define the function \mathcal{G} for the Gaussian covariances, as:

$$\mathcal{G}(u, v) = a(u, v) u, \quad (8)$$

with a slope a that depends on the parameters u and v .

We will consider the function $\mathcal{G}(\lambda, h_0)$ in the isotropic covariances case and $\mathcal{G}(\lambda_i, h_{0,i})$ in the anisotropic cases. According to our numerical experiments, the slope a will depend on the input parameters λ_i and $h_{0,i}$ (See section 4).

3.2 Estimation of the enlarged domain

For the Matérn family of covariances with $1/2 \leq \nu < \infty$, let us denote $\ell_{\mathcal{M},i}$ the estimated enlarged domain size in the direction i computed with the functions \mathcal{M} :

$$\ell_{\mathcal{M},i} := \mathcal{M}(\nu, \lambda_i/h_{0,i}) \lambda_i. \quad (9)$$

For the case $\lambda_i = \lambda$, $m_{0,i} = m_0$, $L_i = L$, $\forall i = 1, \dots, d$, the estimated enlarged domain does not depend on the direction i and we simply use the notation $\ell_{\mathcal{M}}$.

For the Gaussian covariances, let us denote $\ell_{\mathcal{G},i}$ the estimated enlarged domain size in the direction i computed with the functions \mathcal{G} :

$$\ell_{\mathcal{G},i} := \mathcal{G}(\lambda_i, h_{0,i}). \quad (10)$$

For the case $\lambda_i = \lambda$, $m_{0,i} = m_0$, $L_i = L$, $\forall i = 1, \dots, d$, the estimated enlarged domain does not depend on the direction i and we simply use the notation $\ell_{\mathcal{G}}$.

3.3 Estimation of m_i

For the Matérn family of covariances, we propose to compute an estimate of m_i , denoted $m_i^{\mathcal{M},est}$, $i = 1, \dots, d$, as:

$$m_i^{\mathcal{M},est} := \max(m_{0,i}, \lceil \ell_{\mathcal{M},i}/h_{0,i} \rceil) \quad (11)$$

with $\lceil x \rceil$ the notation for the ceiling part of x and $\ell_{\mathcal{M},i}$ defined by (9).

From $m_i^{\mathcal{M},est}$, we define the value of the enlarged domain in the direction i for a Matérn covariance case with $1/2 \leq \nu < \infty$ as: $\ell_{i,est} := m_i^{\mathcal{M},est} \times h_{0,i}$. For the case $\lambda_i = \lambda$, $m_{0,i} = m_0$, $L_i = L$, $\forall i = 1, \dots, d$, the estimated enlarged domain does not depend on the direction i and

we simply use the notations $m^{\mathcal{M},est} := \max(m_0, \lceil \ell_{\mathcal{M}}/h_0 \rceil)$ and $\ell_{est} := m^{\mathcal{M},est} \times h_0$.

For the Gaussian covariances, we propose to compute an estimate of m_i , denoted $m_i^{\mathcal{G},est}$, $i = 1, \dots, d$, as:

$$m_i^{\mathcal{G},est} := \max(m_{0,i}, \lceil \ell_{\mathcal{G},i}/h_{0,i} \rceil) \quad (12)$$

with $\lceil x \rceil$ the notation for the ceiling part of x and $\ell_{\mathcal{G},i}$ defined by (10).

From $m_i^{\mathcal{G},est}$, we define the value of the enlarged domain in the direction i for a Gaussian covariance case as: $\ell_{est} := m_i^{\mathcal{G},est} h_{0,i}$. Again, for the case $\lambda_i = \lambda$, $m_{0,i} = m_0$, $L_i = L$, $\forall i = 1, \dots, d$, the estimated enlarged domain does not depend on the direction i and we simply use the notations $m^{\mathcal{G},est} := \max(m_0, \lceil \ell_{\mathcal{G}}/h_0 \rceil)$ and $\ell_{est} := m^{\mathcal{G},est} \times h_0$.

Now we have all the necessary materials to design the optimized algorithms to compute the required domain size for the CEM algorithm.

3.4 New optimized algorithms

The estimate (11), $m_i^{\mathcal{M},est}$ of m_i , implemented in the new Algorithm 4 below, intends to save computational time by comparison with a start at $m_{0,i}$ as in Step 1 of Algorithm 2.

Algorithm 4 New algorithm for the Matérn family of covariance ($\frac{1}{2} \leq \nu < \infty$) to compute the size of the extended domain to guarantee that \mathbf{R}^{ext} is positive definite (replace Algorithm 2).

Data: d , $m_{0,i}$ ($i = 1, \dots, d$) and a Matérn covariance function ρ with $1/2 \leq \nu < \infty$.

Result: Number of points m_i ($i = 1, \dots, d$) to guarantee that \mathbf{R}^{ext} is positive definite and the vector of eigenvalues \mathbf{v} .

1. **[novelty]** Set $m_i = m_i^{\mathcal{M},est}$, $i = 1, \dots, d$, with $m_i^{\mathcal{M},est}$ given by (11) to get a good first guess of the domain size.
 2. Calculate \mathbf{r} , the first column of \mathbf{R}^{ext} .
 3. Calculate \mathbf{v} , the vector of eigenvalues of \mathbf{R}^{ext} , by d -dimensional FFT on \mathbf{r} .
 4. If smallest eigenvalue < 0 then increment m_i in each direction ($i = 1, \dots, d$) and go to Step 2.
-

The estimate (12), $m_i^{\mathcal{G},est}$ of m_i , intends to give a good order of magnitude of the required domain size and to save computational time by comparison with a start at $m_{0,i}$ as in Step 1 of Algorithm 2. This yields the following new Algorithm 5:

Remark 3.1 Note here we have made the two following assumptions:

1. Theorems 2.3 and 2.4 consider the cases of isotropic covariances and a unit cube domain of size $L=1$. To consider more general domains $L_1 \times \dots \times L_d$ (d is the dimension) and the cases of anisotropic covariances, we have made a (quite natural) assumption that Theorems 2.3 and 2.4 can be extended in these more general cases only by replacing λ by λ_i and h_0 by the scaling $L_i/m_{0,i}$.

Algorithm 5 New algorithm for the Gaussian covariance ($\nu = \infty$) to compute the size of the extended domain to guarantee that \mathbf{R}^{ext} is positive definite (replace Algorithm 2).

Data: $d, m_{0,i}$ ($i = 1, \dots, d$), and a Gaussian covariance function ρ .

Result: Number of points m_i ($i = 1, \dots, d$) to guarantee that \mathbf{R}^{ext} is positive definite and the vector of eigenvalues \mathbf{v} .

1. **[novelty]** Set $m_i = m_i^{\mathcal{G},est}$, $i = 1, \dots, d$, with $m_i^{\mathcal{G},est}$ given by (12) to get a good first guess of the domain size.
 2. Calculate \mathbf{r} , the first column of \mathbf{R}^{ext} .
 3. Calculate \mathbf{v} , the vector of eigenvalues of \mathbf{R}^{ext} , by d -dimensional FFT on \mathbf{r} .
 4. If smallest eigenvalue < 0 then increment m_i in each direction ($i = 1, \dots, d$) and go to Step 2.
-

2. We propose to keep on applying the padding estimation in each direction even for the cases of non cubic domains and anisotropic covariances. We do not have any proof yet that it is the best choice. However it seems a natural choice and our numerical experiments in Section 5 show that this choice works very well in practice.

4 Estimation of the parameters of the fitting functions

To estimate the values of the parameters c_1^{est} and c_2^{est} of the function \mathcal{M} defined by (7) and the value the parameter a of the function \mathcal{G} defined by (8), let us perform an equivalent set of numerical experiments as the one done in [8] for isotropic Matérn covariances and Gaussian covariance. We consider a simple uniform domain with

$$M = (m_0 + 1)^d$$

grid points on the d -dimensional unit cube $[0; L]^d$ with $L = 1$, m_0 and d given. The grid spacing is denoted $h_0 = \frac{1}{m_0}$ along each direction. First we use Algorithm 2 and we iteratively increase the length of the domain by one space step (at Step 4. $m \leftarrow m+1$) and we note down the minimum value of the domain size ℓ_{min} to get eigenvalues above a threshold τ . In practice [8], τ is set to -10^{-13} as \mathbf{R}^{ext} has many very small eigenvalues and some of them are negative but very close to zero. Therefore, those eigenvalues which are less than this given threshold τ will be set to 0. However for the Gaussian 3D test cases, we choose $\tau = -5.10^{-13}$ to avoid some stagnation of the minimum eigenvalue as shown in Appendix A. Also, as the coefficients c_1^{est} , c_2^{est} and a may depend on the dimension d , we perform experiments for $d = 2$ and $d = 3$.

Remark 4.1 Other choice for the increment at Step 4 could have been made. Here we have chosen an increment of 1 to get the exact enlarged domain size ℓ_{min} from which all eigenvalues are above the threshold τ .

Remark 4.2 Notice the values of the parameters estimated here are dependent on the value of τ . The larger the absolute value of τ , the smaller ℓ_{min} . For each set of simulations, the value of τ is given.

4.1 2D Matérn family of covariances

Figure 4 gives the minimum length ℓ_{min} in 2d for different values of the isotropic Matérn smoothness parameter ν and for $\lambda = 0.125$, $\lambda = 0.25$ and $\lambda = 0.5$ respectively.

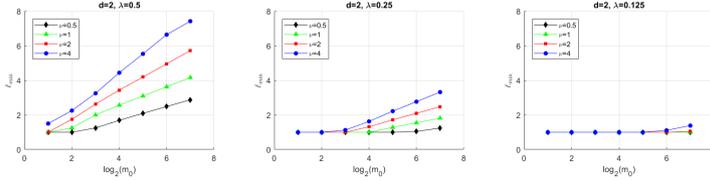


Figure 4: 2D case - isotropic Matérn covariances ($\tau = -10^{-13}$): minimum length ℓ_{min} obtained from numerical experiments versus $\log_2(m_0)$.

We obtain similar results as Fig. 1. (top) in [8]. The sole difference in the 2D graph is that we added the point $\lambda = 0.5$, $\nu = 4$, $m_0 = 128$ for which the threshold $\tau = -10^{-13}$ can only be reached with an extended 80-bit precision code.

Now, according to Theorem 2.3, we are interested in the quantities ℓ_{min}/λ versus

$$\zeta = \nu^{0.5} \log(\max(\lambda/h_0, \nu^{0.5})).$$

Figure 5 shows the quantities ℓ_{min}/λ versus ζ (computed with the data of Figure 4).

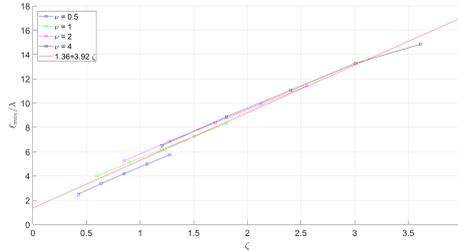


Figure 5: 2D case - isotropic Matérn covariances ($\tau = -10^{-13}$): ℓ_{min}/λ versus $\zeta = \nu^{0.5} \log(\max(\lambda/h_0, \nu^{0.5}))$.

The line that fits the all the data best (in a least-squares sense) has the equation $c_1^{est} + c_2^{est} \zeta$ with $c_1^{est} = 1.36$ and $c_2^{est} = 3.92$ (red line). Note that the estimated length in the case $\nu = 0.5$ will be slightly overestimated.

Figure 6 shows the 3D graph $\ell_{\mathcal{M}}/\lambda = \mathcal{M}(\nu, \lambda/h_0)$ with \mathcal{M} defined by (7) with $c_1^{est} = 1.36$ and $c_2^{est} = 3.92$. The surface is plotted in blue, together with red dots corresponding to the numerical experiments from Figure 4.

Some red dots are slightly above the blue map representing \mathcal{M} and in these cases, it means that the estimated value $\ell_{\mathcal{M}}$ of ℓ_{min} will be sufficient to guarantee the positiveness of \mathbf{R}^{ext} : $\ell_{min} < \ell_{\mathcal{M}}$. For the red dots slightly above the blue map, some more steps will be needed in Algorithm 2 to

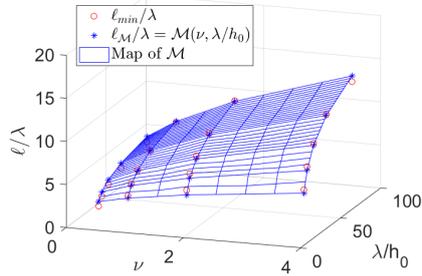


Figure 6: 2D case - isotropic Matérn covariances ($\tau = -10^{-13}$): ℓ/λ versus λ/h_0 and ν .

reach the value of ℓ_{min} (or equivalently m) to guarantee the positiveness of \mathbf{R}^{ext} (ℓ_{min} is slightly above ℓ_M).

4.2 3D Matérn family of covariances

Similarly, we do the same numerical experiments in 3D ($d = 3$). Figure 7 gives the minimum length in 3d for different values of the isotropic Matérn smoothness parameter ν and for $\lambda = 0.125$, $\lambda = 0.25$ and $\lambda = 0.5$ respectively. We obtain similar results as Fig. 1. (bottom) in [8].

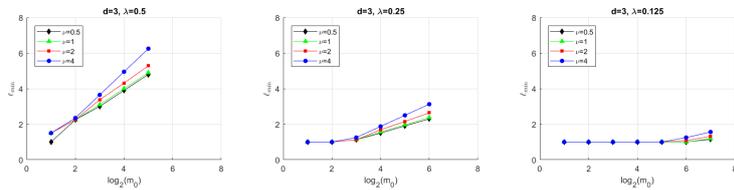


Figure 7: 3D case - isotropic Matérn covariances ($\tau = -10^{-13}$): minimum length ℓ_{min} obtained from numerical experiments versus $\log_2(m_0)$.

Now, according to Theorem 2.3, we are interested in the quantities ℓ_{min}/λ versus

$$\zeta = \nu^{0.5} \log(\max(\lambda/h_0, \nu^{0.5})).$$

Figure 8 shows the quantities ℓ_{min}/λ versus ζ (computed with the data of Figure 7).

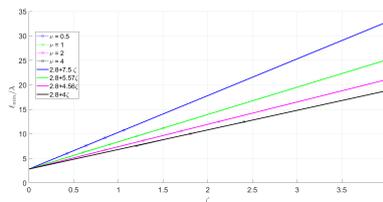


Figure 8: 3D case ($\tau = -10^{-13}$): ℓ/λ versus $\zeta = \nu^{0.5} \log(\max(\lambda/h_0, \nu^{0.5}))$.

Contrary to the 2D case, here we observe more differences in the slopes of the curves with respect to ν . Table 1 gives the linear fits $\alpha \zeta + \beta$ of the data with an intercept equal to $\beta = 2.8$. The slope depends on ν . A good fit of the slope is a power fit of equation: $\alpha_{est} = 5.83\nu^{-0.301}$ as shown on Table 1 with a percentage of relative error between α and α_{est} below 5%.

ν	β	α	α_{est}	Relative error on α
0.5	2.8	7.5	7.18	4.17 %
1	2.8	5.57	5.83	4.69 %
2	2.8	4.56	4.73	3.71 %
4	2.8	4	3.84	3.90 %

Table 1: 3D case - isotropic Matérn covariances ($\tau = -10^{-13}$): Estimation of the slopes α with respect to ν

In 3D, we propose to choose $c_1^{est} = 2.8$ and $c_2^{est}(\nu) = 5.83\nu^{-0.301}$.

Figure 9 shows the 3D graph $\ell_{\mathcal{M}}/\lambda = \mathcal{M}(\nu, \lambda/h_0)$ with the estimates $c_1^{est} = 2.8$ and $c_2^{est}(\nu) = 5.83\nu^{-0.301}$ previously obtained. The surface is plotted in blue, together with red dots corresponding to the numerical experiments from Figure 7. As in 2D, some red dots are slightly above the blue map representing \mathcal{M} and in these cases, it means that the estimated value $\ell_{\mathcal{M}}$ of ℓ_{min} will be sufficient to guarantee the positiveness of \mathbf{R}^{ext} : $\ell < \ell_{\mathcal{M}}$. For the red dots slightly above the blue map, some more steps will be needed in Algorithm 2 to reach the value of ℓ_{min} (or equivalently m) to guarantee the positiveness of \mathbf{R}^{ext} (ℓ_{min} is slightly above $\ell_{\mathcal{M}}$).

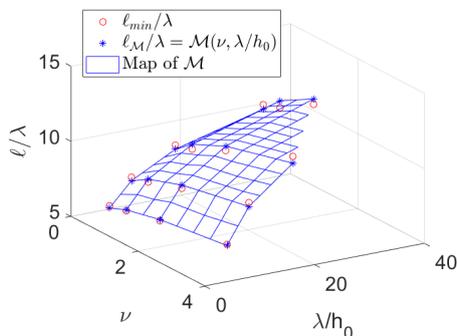


Figure 9: 3D case - isotropic Matérn covariances ($\tau = -10^{-13}$): ℓ/λ versus λ/h_0 and ν .

4.3 2D Gaussian covariances

Table 2 gives the required domain size ℓ_{min} in 2D to guarantee that \mathbf{R}^{ext} is positive for different choices of the parameters m_0 and λ . Table 3 gives the value of ℓ_{min} obtained from our numerical experiments for constant values of λ/h_0 . As predicted by Theorem 2.4, the larger λ , the larger the padding. Moreover, for a given λ , the larger m_0 (hence the larger the ratio λ/h_0 as $h_0 = 1/m_0$), the larger the padding.

$m_0 \backslash \lambda$	0.125	0.25	0.5	1
2	1	1	1	2.5
4	1	1	1.25	5.75
8	1	1	2.875	5.75
16	1	1.4375	2.875	5.8125
32	1	1.4375	2.90625	5.90625
64	1	1.453125	2.95312	6.01562
128	1	1.47656	3.00781	6.125

Table 2: 2D case - Gaussian covariance ($\tau = -10^{-13}$): Minimum length ℓ_{min} obtained from numerical experiments (Algorithm 2 with an increment of 1)

$\lambda/h_0 \backslash \lambda$	0.125	0.25	0.5	1
2	1	1	1.25	2.5
3	1	1.08333	2.16667	4.33333
4	1	1.4375	2.875	5.75
8	1	1.4375	2.875	5.75
16	1	1.453125	2.90625	5.8125
32	1	1.47656	2.95312	5.90625
64	1	1.50391	3.00781	6.01562

Table 3: 2D case - Gaussian covariance ($\tau = -10^{-13}$): Numerical experiments: Minimum length required for constant values of λ/h_0

Figure 10 displays the data of Table 3 ($\tau = -10^{-13}$).

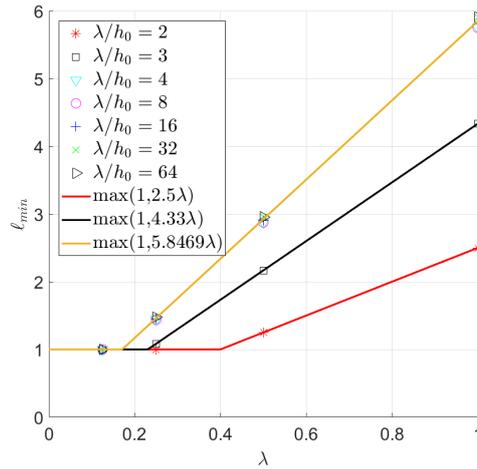


Figure 10: 2D case - Gaussian covariance: Numerical experiments: minimum length ℓ_{min} versus λ for constant values of λ/h_0

As predicted by Theorem 2.4, we clearly identify different regimes

depending on the value λ/h_0 . For $\lambda/h_0 \geq 4$ (cases 4, 8, 16, 32 and 64), a good estimate of the slope is 5.85 (in a least-squares sense, yellow curve). It is very close to the value $B_{2D} = 5.57$ proved in Theorem 2.4. For $\lambda/h_0 = 3$, a good estimate of the slope is 4.33 (black curve). For $\lambda/h_0 = 2$, a good estimate of the slope is 2.5 (red curve). The slope of the linear fits for the case $\lambda/h_0 < 4$ can be estimated from $\sqrt{2}\lambda/h_0$ as suggested by Theorem 2.4. Then for $\lambda/h_0 = 2$, we use a slope of 2.82 and for the case $\lambda/h_0 = 3$, we use a slope of 4.24.

4.4 3D Gaussian covariances

Here τ is chosen equal to -5.10^{-13} to avoid stagnation of the minimum eigenvalue as depicted in paragraph A.

Table 4 gives the required domain size ℓ_{min} in 3D to guarantee that \mathbf{R}^{ext} is positive definite for different choices of the parameters m_0 and λ . Table 5 gives the value of ℓ_{min} obtained from our numerical experiments for constant values of λ/h_0 . As predicted by Theorem 2.4, the larger λ , the larger the padding. Moreover, for a given λ , the larger m_0 (hence the larger the ratio λ/h_0 as $h_0 = 1/m_0$), the larger the padding.

$m_0 \backslash \lambda$	0.125	0.25	0.5	1
2	1	1	1	2.5
4	1	1	1.25	5.75
8	1	1	2.875	5.875
16	1	1.4375	2.9375	6
32	1	1.46875	3	6.125
64	1	1.5	3.0625	6.26562
128	1	1.53125	3.13281	-

Table 4: 3D case - Gaussian covariance ($\tau = -5.10^{-13}$): Minimum length ℓ_{min} obtained from numerical experiments

$\lambda/h_0 \backslash \lambda$	0.125	0.25	0.5	1
2	1	1	1.25	2.5
3	1	1.08333	2.16667	4.33333
4	1	1.4375	2.875	5.75
8	1	1.46875	2.9375	5.875
16	1	1.5	3	6
32	1	1.53125	3.0625	6.125
64	1	1.56641	3.13281	6.26562

Table 5: 3D case - Gaussian covariance: Numerical experiments: Minimum length required for constant values of λ/h_0

Figure 11 displays the data of Table 5.

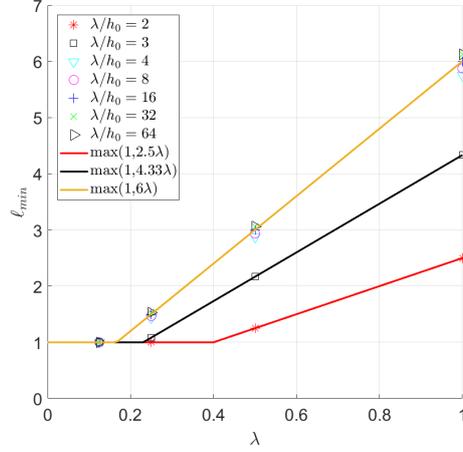


Figure 11: 3D case - Gaussian covariance: Numerical experiments: minimum length ℓ_{min} versus λ for constant values of λ/h_0

As predicted by Theorem 2.4, we clearly identify two regimes depending on the value λ/h_0 : the case $\lambda/h_0 \geq 4$ and the case $\lambda/h_0 < 4$. For $\lambda/h_0 \geq 4$, a good estimate of the slope is 6 (in a least-squares sense). It is a little bit lower than the value of $B_{3D} = 7.515$ in Theorem 2.4. For $\lambda/h_0 = 3$, a good estimate of the slope is 4.33 (as in 2D). For $\lambda/h_0 = 2$, a good estimate of the slope is 2.5 (as in 2D). As in 2D, the slope of the linear fits for the case $\lambda/h_0 < 4$ can be estimated from $\sqrt{2}\lambda/h_0$.

4.5 Summary of the parameters values

To sum up, our numerical experiments suggest to take the following values of the parameters c_1^{est} and c_2^{est} of the function \mathcal{M} given in (7) and for the parameter a of the function \mathcal{G} given in (8):

- for the Matérn family of covariances ($1/2 \leq \nu < \infty$),
 - [2D case:] $c_1^{est} = 1.36$ and $c_2^{est} = 3.92$;
 - [3D case:] $c_1^{est} = 2.8$ and $c_2^{est}(\nu) = 5.83\nu^{-0.301}$.
- for the 2D isotropic Gaussian covariance, a good approximation of a is given by:
 - [$\lambda/h_0 < 4$]: $a = \sqrt{2}\lambda/h_0$;
 - [$\lambda/h_0 \geq 4$]: $a = 5.85$.
- for the 3D isotropic Gaussian covariance, a good approximation of a is given by:
 - [$\lambda/h_0 < 4$]: $a = \sqrt{2}\lambda/h_0$;
 - [$\lambda/h_0 \geq 4$]: $a = 6$.

Remark 4.3 These estimates of c_1^{est} , c_2^{est} and a could be improved, especially by adding more simulations with larger ratio λ/h_0 .

5 Numerical experiments

We use the values of c_1^{est} and c_2^{est} and a estimated in Section 4 for the function \mathcal{M} given in (7) and \mathcal{G} given in (8) respectively.

In these numerical experiments, we increase m by one at Step 4 ($m \leftarrow m + 1$), both in our optimized Algorithms 4 and 5 and also in the basic Algorithm 2. Then the exact value of ℓ_{min} can be derived from the number of iterations obtained with Algorithm 2.

In the following, we perform two kinds of validation. First we check the quality of the estimated length $\ell_{i,est} := m_i^{\mathcal{M},est} \times h_{0,i}$ with $m_i^{\mathcal{M},est}$ defined by (11). Second, we compute the number of iterations needed to reach the required enlarged domain size. By number of iterations, we refer to the number of times m is increased (step 4). Zero iteration means the domain size is large enough. Our objective is to show that our optimized Algorithms 4 and 5 decrease significantly the number of iterations.

5.1 Isotropic covariance

5.1.1 Isotropic Matérn family of covariance

In the isotropic case and for a cubic domain of size L , we use the notation, $\ell_{est} := m^{\mathcal{M},est} \times h_0$ with $m^{\mathcal{M},est} := \max(m_0, \lceil \ell_{\mathcal{M}}/h_0 \rceil)$, with $h_0 := L/m_0$, as the estimate does not depend on the direction i .

Table 6 and Table 7 give some examples of ℓ_{est} computed from $m^{\mathcal{M},est}$ of Algorithm 4 and the relative error with respect to ℓ_{min} numerically estimated with Algorithm 2. We also performed some experiments with $\nu = 5$.

L	ν	λ	m_0	λ/L	λ/h_0	ℓ_{est}	ℓ_{min}	$ \ell_{est} - \ell_{min} /\ell_{min}$	ℓ_{est}/L
0.5	0.5	0.125	5	0.25	1.25	0.5	0.5	0.00%	1
10	0.5	5	32	0.5	16	23.75	20.9375	13.43 %	2.38
0.5	1	0.25	14	0.5	7	1.1786	1.21429	2.94%	2.36
10	1	8	32	0.8	25.6	55.3125	55.3125	0.0 %	5.53
0.5	2	0.25	10	0.5	5	1.35	1.45	6.9%	2.7
0.5	2	0.25	14	0.5	7	1.5357	1.6428	6.52 %	3.07
10	2	10	32	1	32	97.1875	99.0625	1.89 %	9.72
10	4	12	32	1.2	38.4	165.625	165.625	0.0 %	16.56
1	5	0.25	64	0.25	16	2.98	3.02	1.18 %	2.98
1	5	0.25	128	0.25	32	3.64	3.46	5.2 %	3.64
1	5	0.5	32	0.5	16	5.96	6.03	1.18 %	5.97
1	5	0.5	64	0.5	32	7.28	6.92	5.17 %	7.28

Table 6: 2D case - isotropic Matérn covariances ($\tau = -10^{-13}$): various test cases, ℓ_{est} estimated from Algorithm 4 versus ℓ_{min} .

In 2D, we have a slight overestimation for $\nu = 0.5$ as expected. This is not the case in 3D as the fitting is sharper. The other estimations are very good. For small ratio λ/h_0 and $\lambda/L \ll 1$, almost (or even no) padding

L	ν	λ	m_0	λ/L	λ/h_0	ℓ_{est}	ℓ_{min}	$ \ell_{est} - \ell_{min} /\ell_{min}$	ℓ_{est}/L
0.5	0.5	0.125	2.25	5	1.25	0.5	0.5	0.00%	1
10	0.5	5	16	0.5	8	37.5	38.125	1.64%	3.75
0.5	1	0.25	14	0.5	7	1.9643	1.89286	3.77%	3.93
10	1	8	16	0.8	12.8	74.375	71.875	3.48%	7.44
0.5	2	0.25	10	0.5	5	1.9	1.85	2.7%	3.8
10	2	10	8	1	8	88.75	86.25	2.9%	8.88
10	4	12	8	1.2	9.6	125	127.5	1.96%	12.5
1	5	0.125	32	0.125	4	1	1	0 %	1
1	5	0.125	64	0.125	8	1.26	1.33	5.13 %	1.26
1	5	0.5	8	0.5	4	3.875	3.875	0 %	3.88
1	5	0.5	16	0.5	8	5	5.31	5.88 %	5

Table 7: 3D case - isotropic Matérn covariances ($\tau = -10^{-13}$): various test cases, ℓ_{est} estimated from Algorithm 4 versus ℓ_{min} .

is necessary.

A good estimate of ℓ_{est} yields a small number of iterations at Step 4 of the optimized Algorithm 4. Figure 12 and Figure 13 show the number of iterations in 2D and in 3D respectively ($L = 1$, $m_0 = 32$), for two ratios $\lambda/h_0 = 8$ ($\lambda = 0.25$) and $\lambda/h_0 = 32$ ($\lambda = 1$).

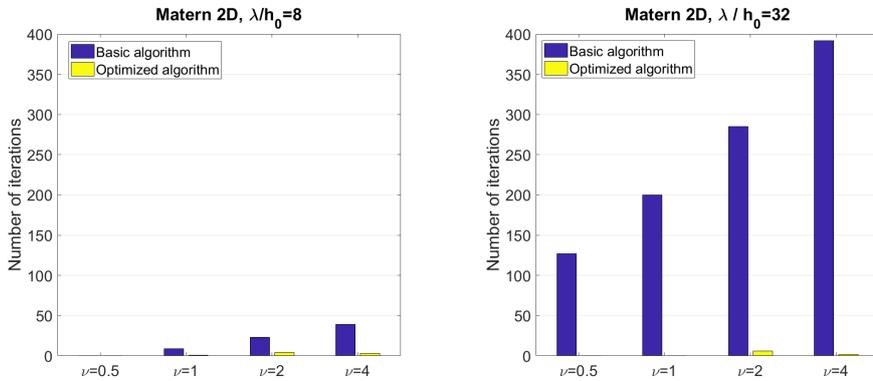


Figure 12: 2D isotropic Matérn family of covariance ($\tau = -10^{-13}$): Number of iterations at Step 4 of the basic Algorithm 2 and of the optimized Algorithm 4

Algorithm 4 allows to jump quickly to a value very close to the required ℓ_{min} as shown by the very small (or even zero) number of iterations, even for large ratio λ/h_0 . On the contrary, the number of iterations with Algorithm 2 is very large.

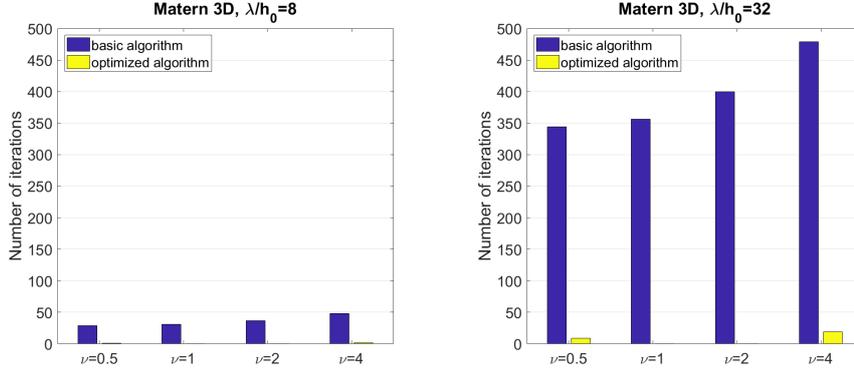


Figure 13: 3D isotropic Matérn family of covariance ($\tau = -10^{-13}$): Number of iterations at Step 4 of the basic Algorithm 2 and of the optimized Algorithm 4

5.1.2 Isotropic Gaussian covariance

Table 8 and Table 9 give some examples of ℓ_{est} computed at Step 1 of Algorithm 5 and the relative error with respect to ℓ_{min} numerically estimated with Algorithm 2. The estimates are very good. As expected, when a padding is required, the larger λ , the larger ℓ_{min} and for a given λ , the larger the ratio λ/h_0 , the larger the padding.

L	λ	m_0	λ/L	λ/h_0	ℓ_{est}	ℓ_{min}	$ \ell_{est} - \ell_{min} /\ell_{min}$	ℓ_{est}/L
0.5	0.125	0.25	5	1.25	0.5	0.5	0.00%	1
0.5	0.25	14	0.5	7	1.4643	1.42857	2.5%	2.93
1	0.0625	40	0.0625	2.5	1	1	0.0 %	1
1	0.125	28	0.125	3.5	1	1	0.0 %	1
1	0.25	14	0.25	3.5	1.2857	1.36	5.46 %	1.29
1	0.5	7	0.5	3.5	2.57	2.71	5.32 %	2.57
1	0.5	128	0.5	64	2.93	3.01	2.59 %	2.93
1	1	128	1	128	5.85	6.125	4.49 %	5.85
10	5	10	0.5	5	30	29	3.44%	3
10	5	32	0.5	16	29.375	29.0625	1.08%	2.94
10	8	32	0.8	25.6	46.875	47.1875	0.66%	4.69
10	10	32	1	32	58.75	59.0625	0.53%	5.88
10	12	32	1.2	38.4	70.3125	71.25	1.32%	7.03

Table 8: 2D case - isotropic Gaussian covariances ($\tau = -10^{-13}$): various test cases, ℓ_{est} estimated from Algorithm 5 versus ℓ_{min} .

A good estimate of ℓ_{est} yields a small number of iterations at Step 4 of the optimized Algorithm 5. Figures 14 shows the number of iterations for two ratios $\lambda/h_0 = 8$ and $\lambda/h_0 = 32$ ($m_0 = 32$, $L = 1$, $\lambda = 1$ and $\lambda = 0.25$ respectively). Algorithm 5 computes a good estimate ℓ_{est} as shown by the very small numbers of iterations to reach ℓ_{min} .

L	λ	m_0	λ/L	λ/h_0	ℓ_{est}	ℓ_{min}	$ \ell_{est} - \ell_{min} /\ell_{min}$	ℓ_{est}/L
0.5	0.125	5	0.25	1.25	0.5	0.5	0.00%	1
0.5	0.25	10	0.5	5	1.5	1.45	3.45%	3
0.5	0.25	14	0.5	7	1.5	1.46	2.44%	3
1	0.0625	40	0.0625	2.5	1	1	0.0 %	1
1	0.125	8	0.125	1	1	1	0.0 %	1
1	0.25	14	0.25	3.5	1.2857	1.36	5.46 %	1.29
1	0.5	7	0.5	3.5	2.5714	2.71	5.11 %	2.57
1	0.5	10	0.5	5	3	2.9	3.49 %	3
1	1	32	1	32	6	6.125	2.04 %	6
10	5	32	0.5	16	30	30	0.00%	3
10	8	32	0.8	25.6	48.125	48.4375	0.65%	4.8
10	10	32	1	32	60	61.25	2.04%	6
10	12	32	1.2	38.4	72.1875	74.0625	2.53%	7.2

Table 9: 3D case - isotropic Gaussian covariances ($\tau = -5.10^{-13}$): various test cases, ℓ_{est} estimated with Algorithm 5 versus ℓ_{min} .

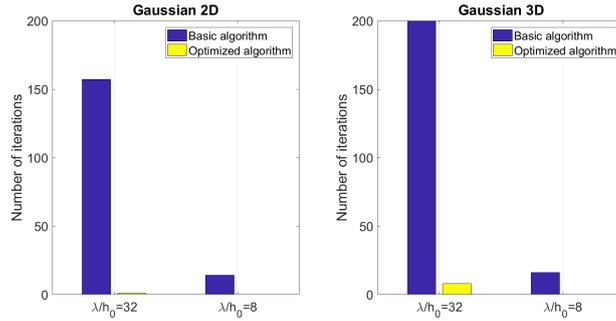


Figure 14: Isotropic Gaussian covariance ($\tau = -10^{-13}$): Number of iterations needed to compute ℓ_{min} using either the basic algorithm (Algorithm 2 with an increment of 1) or the optimized one (Algorithm 5)

5.2 Anisotropic covariances

Let us now consider test cases of anisotropic covariances with different ratio $\lambda_i/h_{0,i}$ in each direction. Tables 10 and 11 summarize the different parameters in 2D and in 3D respectively. Note λ_2 takes values in $\{0.5, 0.8, 1\}$.

Name	L_1	L_2	λ_1	λ_2	$m_{0,1}$	$m_{0,2}$	$\lambda_1/h_{0,1}$	$\lambda_2/h_{0,2}$
$A_{\lambda_2=0.5}^{2D}$	10	1	0.1	0.5	32	8	0.32	4
$A_{\lambda_2=0.8}^{2D}$	10	1	0.1	0.8	32	8	0.32	6.4
$A_{\lambda_2=1}^{2D}$	10	1	0.1	1	32	8	0.32	8

Table 10: Anisotropic test cases - 2D parameters.

Name	L_1	L_2	L_3	λ_1	λ_2	λ_3	$m_{0,1}$	$m_{0,2}$	$m_{0,3}$	$\lambda_1/h_{0,1}$	$\lambda_2/h_{0,2}$	$\lambda_3/h_{0,3}$
$A_{\lambda_2=0.5}^{3D}$	10	1	5	0.1	0.5	4	32	8	16	0.32	4	12.8
$A_{\lambda_2=0.8}^{3D}$	10	1	5	0.1	0.8	4	32	8	16	0.32	6.4	12.8
$A_{\lambda_2=1}^{3D}$	10	1	5	0.1	1	4	32	8	16	0.32	8	12.8

Table 11: Anisotropic test cases - 3D parameters.

5.2.1 Anisotropic Matérn family of covariances

We consider $\nu \in \{0.5, 1, 2, 4\}$. Tables 12 and 13 compare the number of iterations between Algorithm 2 and 4 in 2D and in 3D respectively. The estimated padding with Algorithm 4 is excellent as no additional iteration is necessary to reach a valid domain for the CEM.

Name	ν	Algorithm 2	Algorithm 4		
		Nbr iterations	$\ell_{1,est}$	$\ell_{2,est}$	Nbr iterations
$A_{\lambda_2=0.5}^{2D}$	0.5	0	10	1.625	0
$A_{\lambda_2=0.8}^{2D}$	0.5	0	10	3	0
$A_{\lambda_2=1}^{2D}$	0.5	0	10	3	0
$A_{\lambda_2=0.5}^{2D}$	1	0	10	1.875	0
$A_{\lambda_2=0.8}^{2D}$	1	7	10	3.625	0
$A_{\lambda_2=1}^{2D}$	1	13	10	5	0
$A_{\lambda_2=0.5}^{2D}$	2	8	10	2.375	0
$A_{\lambda_2=0.8}^{2D}$	2	24	10	4.75	0
$A_{\lambda_2=1}^{2D}$	2	34	10	6.375	0
$A_{\lambda_2=0.5}^{2D}$	4	15	10	3.125	0
$A_{\lambda_2=0.8}^{2D}$	4	38	10	6.25	0
$A_{\lambda_2=1}^{2D}$	4	55	10	8.5	0

Table 12: Anisotropic Matérn family of covariance ($\tau = 10^{-13}$) - Comparison of the number of iterations between Algorithm 2 and 4.

5.2.2 Anisotropic Gaussian covariances

Tables 14 and 15 compare the number of iterations between Algorithm 2 and 5 in 2D and in 3D respectively. Again, the estimated padding with Algorithm 5 is excellent as no additional iteration is necessary to reach a valid domain for the CEM.

Name	ν	Algorithm 2	Algorithm 4			
		Nbr iterations	$\ell_{1,est}$	$\ell_{2,est}$	$\ell_{3,est}$	Nbr iterations
$A_{\lambda_2=0.5}^{3D}$	0.5	27	10	3	33.75	0
$A_{\lambda_2=0.8}^{3D}$	0.5	35	10	5.625	33.75	0
$A_{\lambda_2=1}^{3D}$	0.5	38	10	7.5	33.75	0
$A_{\lambda_2=0.5}^{3D}$	1	55	10	3.25	37.19	0
$A_{\lambda_2=0.8}^{3D}$	1	60	10	6.125	37.19	0
$A_{\lambda_2=1}^{3D}$	1	62	10	8.125	37.19	0
$A_{\lambda_2=0.5}^{3D}$	2	84	10	3.5	40.94	0
$A_{\lambda_2=0.8}^{3D}$	2	87	10	6.625	40.94	0
$A_{\lambda_2=1}^{3D}$	2	88	10	8.875	40.94	0
$A_{\lambda_2=0.5}^{3D}$	4	118	10	3.75	45.31	0
$A_{\lambda_2=0.8}^{3D}$	4	119	10	7.25	45.31	0
$A_{\lambda_2=1}^{3D}$	4	119	10	9.75	45.31	0

Table 13: Anisotropic Matérn family of covariance ($\tau = 10^{-13}$) - Comparison of the number of iterations between Algorithm 2 and 4.

Name	Algorithm 2	Algorithm 5		
	Nbr iterations	$\ell_{1,est}$	$\ell_{2,est}$	Nbr iterations
$A_{\lambda_2=0.5}^{2D}$	14	10	3	0
$A_{\lambda_2=0.8}^{2D}$	27	10	4.75	0
$A_{\lambda_2=1}^{2D}$	36	10	5.875	0

Table 14: Anisotropic Gaussian covariance ($\tau = -10^{-13}$) - Comparison of the number of iterations between Algorithm 2 and 5.

Name	Algorithm 2	Algorithm 5			
	Nbr iterations	$\ell_{1,est}$	$\ell_{2,est}$	$\ell_{3,est}$	Nbr iterations
$A_{\lambda_2=0.5}^{3D}$	55	10	3	24.0625	0
$A_{\lambda_2=0.8}^{3D}$	56	10	4.875	24.0625	0
$A_{\lambda_2=1}^{3D}$	56	10	6	24.0625	0

Table 15: Anisotropic Gaussian covariance ($\tau = -5.10^{-13}$) - Comparison of the number of iterations between Algorithm 2 and 5.

6 Conclusion

We have designed new algorithms (Algorithm 4 and Algorithm 5) that compute fairly good guesses of the required extended domain size for the CEM algorithm to work. These guesses allow to save computational resources by comparison with the standard CEM algorithm (Algorithm 2). They are inspired by the theoretical work from [8] and tuned with nu-

merical simulations. Several test cases have been conducted to check the efficiency of these new algorithms in terms of estimation of the enlarged domain size and number of iterations and the results are very good. These new algorithms are implemented in the `ParaCirce` library. Our current work is to study the performance of `ParaCirce`, especially its parallel efficiency, in order to provide all the necessary tools for large simulations of GRF with the CEM (typically in case of large ratio λ_i/h_{0_i}). In a recent paper [1], a better estimate for ℓ_{min} is proven thanks to a smooth periodization of the Matérn family of covariance functions ($0 \leq \nu < \infty$): the estimate does not blow up with λ/h_0 and depends only on ν (see equation (1.11) in [1]). A nice future work would be to apply our fitting procedure to that case. Finally, another interesting work would be to cook up a rule to automatically set the value of the threshold τ according to the input parameters and the floating point precision (in view of the stagnation of the minimum eigenvalue for some test cases as described in Appendix A).

A Influence of the floating point precision on the computations

The numerical simulations of this paper have been performed using a 80-bit precision code. The motivation comes from the observation that, depending on the input parameters and the chosen numerical precision, the minimum eigenvalue might reach a plateau and potentially never reaches the given threshold τ . Using an extended numerical precision code shifts the plateau, that is why we prefer to use the 80-bit version of `ParaCirce` to avoid, as much as possible, this situation. Examples for the isotropic Matérn covariance are shown on Figure 15 and 16 but the same can be observed from some numerical experiments with the Gaussian covariance in 3D.

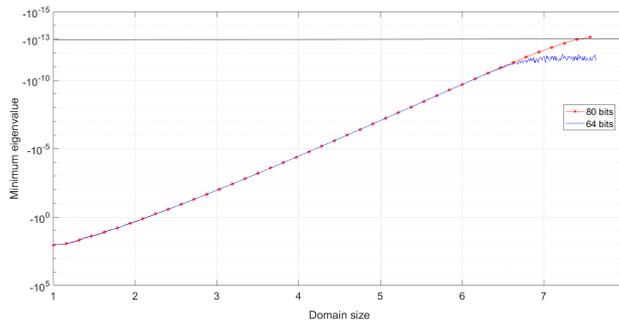


Figure 15: 2D case - isotropic Matérn covariance ($L = 1$, $\nu = 4$, $\lambda = 0.5$, $m_0 = 128$): Minimum eigenvalue versus the domain size ℓ

In the 2D example shown on Figure 15, the 80-bit precision is enough and Algorithm 2 stops in the extended precision case for $\tau = -10^{-13}$. However, in the 3D case shown on Figure 16, the threshold τ must be changed to a lower value for Algorithm 2 to stop.

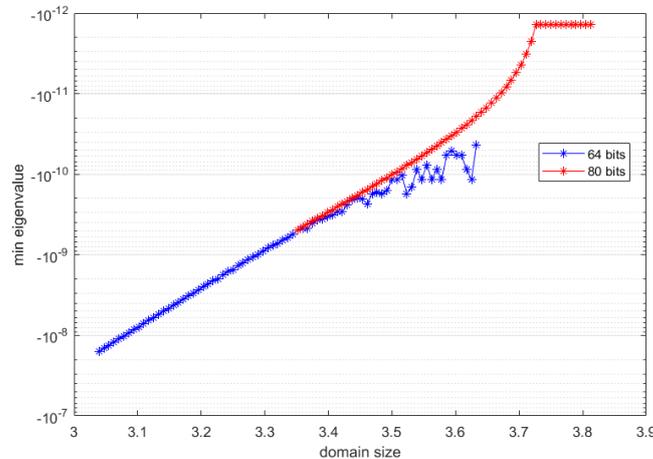


Figure 16: 3D case - isotropic Matérn covariance ($L = 1$, $\nu = 4$, $\lambda = 0.25$, $m_0 = 128$): Minimum eigenvalue versus the domain size ℓ

Acknowledgments. The authors warmly thank Jocelyne Erhel (Inria, France) and Ivan Graham (University of Bath, UK) for fruitful discussions related to the Circulant Embedding Method. The authors are grateful to the CLEPS infrastructure from the Inria of Paris for providing resources and support.

References

- [1] M. Bachmayr, I. G. Graham, V. K. Nguyen, and R. Scheichl. Unified analysis of periodization-based sampling methods for matérn covariances. *SIAM Journal on Numerical Analysis*, 58(5):2953–2980, 2020.
- [2] A. Beaudoin and J.-R. de Dreuzy. Numerical assessment of 3-d macrodispersion in heterogeneous porous media. *Water Resources Research*, 49(5):2489–2496, 2013.
- [3] X. H. Dang. *Identification de la variabilité spatiale des champs de contraintes dans les agrégats polycristallins et application à l’approche locale de la rupture*. Theses, Université Blaise Pascal - Clermont-Ferrand II, October 2012.
- [4] C. R. Dietrich and G. N. Newsam. A fast and exact method for multidimensional gaussian stochastic simulations. *Water Resources Research*, 29(8):2861–2869, 1993.
- [5] M. Feischl, F.Y. Kuo, and I.H. Sloan. Fast random field generation with h-matrices. *Numer. Math.*, 140:639–676, 2018.
- [6] M. Frigo and S. G. Johnson. The design and implementation of fftw3. *Proc. IEEE* 93, 2:216–231, 2005.
- [7] I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl, and I. H. Sloan. Quasi-monte carlo methods for elliptic pdes with random coefficients and applications. *Journal of Computational Physics*, 230(10):3668 – 3694, 2011.

- [8] I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl, and I. H. Sloan. Analysis of Circulant Embedding Methods for Sampling Stationary Random Fields. *SIAM Journal on Numerical Analysis*, 56(3):1871–1895, January 2018.
- [9] P. L’Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton. An objected-oriented random-number package with many long streams and substreams. *Operations Research*, 50(6):1073–1075, 2002.
- [10] Y. Liu, J. Li, S. Sun, and B. Yu. Advances in gaussian random field generation: a review. *Comput Geosci*, 23:1011–1047, 2019.
- [11] G. J. Lord, C. E. Powell, and T. Shardlow. *An Introduction to Computational Stochastic PDEs*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2014.
- [12] C. Schwab and R. A. Todor. Karhunen–loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100 – 122, 2006. Uncertainty Quantification in Simulation Science.