Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Sampling Gaussian stationary random fields: A stochastic realization approach[☆]

Bin Zhu[a,*], Jiahao Liu[a], Zhengshou Lai[b], Tao Qian[c]

[a] *School of Intelligent Systems Engineering, Sun Yat-sen University, Gongchang Road 66, 518107 Shenzhen, China*
[b] *School of Civil Engineering, Sun Yat-sen University, Daxue Road 2, 519082 Zhuhai, China*
[c] *Macau Centre for Mathematical Sciences, Macau University of Science and Technology, Macao Special Administrative Region of China*

ARTICLE INFO

Article history:
Received 13 September 2022
Received in revised form 20 June 2023
Accepted 4 August 2023
Available online xxxx

Keywords:
Stationary random field
Sample generation
Stochastic realization
ARMA model
Multiscale simulation

ABSTRACT

Generating large-scale samples of stationary random fields is of great importance in the fields such as geomaterial modeling and uncertainty quantification. Traditional methodologies based on covariance matrix decomposition have the difficulty of being computationally expensive, which is even more serious when the dimension of the random field is large. This paper proposes an efficient stochastic realization approach for sampling Gaussian stationary random fields from a systems and control point of view. Specifically, we take the exponential and squared exponential covariance functions as examples and make a decoupling assumption when there are multiple dimensions. Then a rational spectral density is constructed in each dimension using techniques from covariance extension, and the corresponding autoregressive moving-average (ARMA) model is obtained via spectral factorization. As a result, samples of the random field with a specific covariance function can be generated very efficiently in the space domain by implementing the ARMA recursion using a Gaussian white noise input. Such a procedure is computationally cheap due to the fact that the constructed ARMA model has a low order. Furthermore, the same method is integrated to multiscale simulations where interpolations of the generated samples are achieved when one zooms into finer scales. Both theoretical analysis and simulation results show that our approach performs favorably compared with covariance matrix decomposition methods.

1. Introduction

There is a vast number of applications of Gaussian random fields across several engineering and scientific disciplines including systems and control [1,2], signal processing [3,4], geotechnical engineering [5–9], image processing [10], biology, and meteorology [11,12]. In these applications, we often face the standard problem of sampling a random field which could be used e.g., for the numerical solution of a stochastic partial differential equation (PDE). In particular in geotechnical engineering, certain geomaterial properties are modeled as zero-mean stationary random fields indexed by spatial or temporal variables. Then the correlations between random variables in the field are described by the covariance function [13]. In order to carry out simulations in geomaterial modeling, a first step is to generate (possibly large-scale) samples of a stationary random field such that its covariances coincide with the values of a prescribed covariance function. A traditional method for this problem is called Covariance Matrix Decomposition (CMD) [14] which employs e.g., the Cholesky factorization (see also [15] for a modified version). Such a method in general costs $O(N^3)$ flops given an $N \times N$ matrix, which is computationally prohibitive when the covariance matrix has a large size. The latter case is typical for multidimensional random fields. For example, a 3-d random field with a (moderate) size $100 \times 100 \times 100$ results in a $10^6 \times 10^6$ covariance matrix after vectorization. Thus applications of CMD are seriously limited to small-scale and unidimensional cases (time series). However, many practical problems involve multidimensional random fields [16,17] and the ability to efficiently generate large-scale samples is also important.

In the literature, there are a number of methods to handle such a problem. Recent developments include [17] which aims to embed a finite multilevel Toeplitz covariance matrix into a larger positive definite multilevel circulant matrix, following earlier works on the embedding problem for (one-level) Toeplitz covariance matrices [18,19]. Then the fast Fourier transform (FFT) can be used to compute the spectral decomposition

[☆] Partial results of this paper were presented at the 23rd Chinese Conference on System Simulation Technology and its Application (CCSSTA 2022).

[*] Corresponding author.

E-mail addresses: zhub26@mail.sysu.edu.cn (B. Zhu), liujh226@mail2.sysu.edu.cn (J. Liu), laizhengsh@mail.sysu.edu.cn (Z. Lai), tqian@must.edu.mo (T. Qian).

https://doi.org/10.1016/j.isatra.2023.08.007
0019-0578/© 2023 ISA. Published by Elsevier Ltd. All rights reserved.

of the (multilevel) circulant matrix at a reduced cost. Another direction is to use the Karhunen–Loève expansion for the continuous covariance kernel and to truncate the expansion for practical computations [4,7,20]. A different strategy for the approximation of the covariance matrix involves the notion of H matrices [21] of which the square root can be computed at an almost linear cost. More recently, hierarchical sampling approaches based on the solution of stochastic PDEs have also been proposed in [11,12,22,23] with applications to multilevel Markov Chain Monte Carlo simulations.

Although the above works are mathematically interesting and deal with general covariance functions, difficulties can still arise when one wants to generate very large-scale samples with a limited hardware configuration (say, a PC), especially when the dimension is greater than or equal to three. In order to handle the latter issue, the paper [24] made a *decoupling assumption* on the multivariable covariance function and proposed a Stepwise CMD method which essentially computes the matrix square root along each dimension and thus reduces the computational cost compared to a full CMD. However, the space domain computation in that paper can be once again greatly reduced if one is able to recognize the frequency-domain structure of the covariance function, namely the *spectral density* of the random field. This is the main idea behind the current paper. More specifically, we draw inspiration from the systems and control literature on *stochastic realization* [25] and *rational covariance extension* in which one aims to describe an underlying random process with a linear dynamical system driven by white noise. The latter topic has undergone decades of development from scalar random processes to vector random fields, see [26–38] and the references therein. In particular, we have shown that the exponential covariance function corresponds exactly to an autoregressive (AR) model (a rational filter) of order one, which is easy to implement recursively and permits the sample generation at a *linear* cost. The decoupling assumption makes straightforward the generalization to multidimensional random fields, and the resulting multidimensional filter is simply a product of individual filters in each dimension. In this way, the filtering procedure is also decoupled as expected.

In addition, our approach is extremely suitable for multiscale simulations, see e.g., [5], where the generated samples of a random field are *interpolated* and fed into a numerical PDE solver in order to obtain a refined solution. The usual interpolation method samples the probability density of the fine-scale random variables whose values are to be determined, conditioned on the coarse-scale samples that have already been generated. The advantage of our approach is that, once a suitable refined noise input has been determined, only "boundary" samples that are necessary to initiate the *fine-scale* ARMA recursions need to be computed from the conditional probability density, and the rest fine-scale samples are generated in the same fashion as the coarse-scale sampling.

The outline of the paper is as follows. In Section 2 we state the problem of sampling a stationary random field with a given covariance function that can be decoupled in each dimension. In Section 3 we propose a stochastic realization approach to the sampling problem based on a set of moment equations, and in Section 4 we focus on the solution to the moment equations given an exponential or a squared exponential covariance function. In Section 5 we integrate our method to multiscale simulations and provide an explicit solution procedure for the bivariate exponential covariance function. A number of numerical simulations, including a comparison with the Stepwise CMD, are presented in Section 6, and in Section 7 we draw the conclusions.

2. Background on sampling random fields

Let $y(\mathbf{t},\omega)$ be a d-dimensional real random field over a probability space $(\Omega,\mathcal{F},P)$ where $\mathbf{t}=(t_1,\dots,t_d)\in\mathbb{R}^d$ can be interpreted as a space (or spatio-temporal) coordinate vector. For each fixed $\mathbf{t}\in\mathbb{R}^d$, assume that $y(\mathbf{t},\cdot)$ is a zero-mean real-valued random variable with a finite variance, that is,

$$\mathbb{E}y(\mathbf{t},\cdot)=0 \quad \text{and} \quad \mathbb{E}\,[y(\mathbf{t},\cdot)]^2<\infty$$

where $\mathbb{E}$ indicates mathematical expectation. It is customary to suppress the dependence on ω and write simply $y(\mathbf{t})$. Assume further that the random field under consideration is *stationary*, which means that the covariance function

$$\rho(\mathbf{t},\mathbf{s}):=\mathbb{E}[y(\mathbf{t})y(\mathbf{s})]$$

depends only on the difference $\boldsymbol{\tau}:=\mathbf{t}-\mathbf{s}$ between the arguments, so we can write $\rho(\boldsymbol{\tau})$ instead.[1] Notice the symmetry $\rho(-\boldsymbol{\tau})=\rho(\boldsymbol{\tau})$.

In applications of geotechnical engineering, see e.g., [39,40], it is often assumed that the covariance function has a decoupled form, namely

$$\rho(\boldsymbol{\tau})=\rho_1(\tau_1)\rho_2(\tau_2)\cdots\rho_d(\tau_d), \tag{1}$$

where each ρ_j is a covariance function in one variable, and $\tau_j\in\mathbb{R}$ is the jth component of $\boldsymbol{\tau}$. In the following, since we shall be concerned with the sampling problem of the random field $y(\mathbf{t})$, let us now define the sampled version of the random field as well as the covariance function. Take $\tau_j=x_jT_j$ where $T_j>0$ is sampling distance and $x_j\in\mathbb{Z}$. Define a random field on the integer grid $\mathbb{Z}^d$ via

$$y_s(\mathbf{x})=y(x_1T_1,\dots,x_dT_d) \tag{2}$$

where the subscript s means "sampled". Then it is easy to deduce that the covariance function of the discrete random field y_s is

$$\rho_s(\mathbf{k})=\rho(k_1T_1,\dots,k_dT_d)=\rho_1(k_1T_1)\cdots\rho_d(k_dT_d), \tag{3}$$

where $\mathbf{k}=(k_1,\dots,k_d)\in\mathbb{Z}^d$ denotes the difference between two discrete grid points. The sampling problem can then be phrased as follows.

Problem 1. Given a covariance function $\rho(\boldsymbol{\tau})$ of the form (1), a vector $\mathbf{T}=(T_1,\dots,T_d)$ of sampling distances, and a vector $\mathbf{N}=(N_1,\dots,N_d)$ of positive integers, generate samples of the random field $y_s(\mathbf{x})$ in (2) for $\mathbf{x}$ in the index set (a regular cuboid)

$$\mathbb{Z}^d_{\mathbf{N}}:=\{(x_1,\dots,x_d):0\le x_j\le N_j-1,\ j=1,\dots,d\} \tag{4}$$

such that its covariance function coincides with the sampled version $\rho_s(\mathbf{k})$ in (3).

The most straightforward approach for this problem is covariance matrix decomposition mentioned in the Introduction. More precisely, since the index set (4) has a finite cardinality, all the samples can be stacked into a long vector $\mathbf{y}$ of dimension $|\mathbf{N}|:=\prod_{j=1}^d N_j$. Then the covariance matrix $\boldsymbol{\Sigma}:=\mathbb{E}(\mathbf{y}\mathbf{y}^\top)$ could in principle be evaluated elementwise according to (3). Problem 1 would then be solved via simple linear algebra

$$\mathbf{y}=L\mathbf{w}, \tag{5}$$

where $\mathbf{w}\sim\mathcal{N}(\mathbf{0},I)$ is an i.i.d. standard normal random vector of dimension $|\mathbf{N}|$, and L is any matrix square root of $\boldsymbol{\Sigma}$ which can in particular, be taken as the Cholesky factor such that $LL^\top=\boldsymbol{\Sigma}$.

[1] This is also called second-order stationary or weak-sense stationary. In addition, if the covariance function $\rho(\boldsymbol{\tau})$ depends only on the length $\|\boldsymbol{\tau}\|$ but not on the specific direction $\boldsymbol{\tau}$, the random field is said to be "isotropic". See Remark 1 at the end of this section.

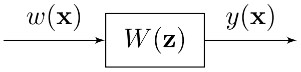


Fig. 1. A d-dimensional linear stochastic system with a white noise input.

However, it is well known that the matrix factorization, which is the major computational burden here, involves $O(|\mathbf{N}|^3)$ flops. Therefore, such a naive approach works only for random fields with a dimension $d = 1$ or 2 when the size of the samples $|\mathbf{N}|$ is not too large. For the generation of large-scale samples, one needs to exploit the inherent structure of the covariance function in order to facilitate fast computation. The latter point is indeed the theme of the next two sections where we will propose an efficient stochastic realization approach to Problem 1. More specifically, we consider two types of 1-d covariance functions of practical interest:

1. the exponential type
$$\rho(x) = \sigma^2 e^{-\alpha|x|}, \tag{6}$$
2. the squared exponential type
$$\rho(x) = \sigma^2 e^{-\alpha|x|^2}, \tag{7}$$

where σ^2 is the variance of the random field and $\alpha > 0$ is a parameter. The multidimensional covariance function is formed through the product in (1). Notice that our method works also for other types of covariance functions $\rho(x)$ provided that the decoupling assumption (1) holds. In this case, the spectral density of the random field can be well approximated in each dimension by a low-order rational model.

Remark 1. The exponential and the squared exponential covariance functions are special cases of the Matérn family of covariance functions, defined as

$$\rho(\boldsymbol{\tau}) = \kappa(\|\boldsymbol{\tau}\|/\lambda) \quad \text{where} \quad \kappa(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)}(\sqrt{2\nu}\, r)^{\nu} K_{\nu}(\sqrt{2\nu}\, r). \tag{8}$$

In the formulas above, $\boldsymbol{\tau} \in \mathbb{R}^d$, $\|\cdot\|$ is the Euclidean norm, λ the correlation length, σ^2 the variance, $\nu > 0$ a smoothness parameter, Γ the gamma function, and K_ν the modified Bessel function of the second kind. Notice that the case $\nu = 1/2$ corresponds to the exponential covariances, while $\nu = \infty$ corresponds to the Gaussian function of the form $\kappa(r) = \sigma^2 e^{-r^2/2}$, see e.g., [17, Example 2.7]. Notice that a random field with a covariance function $\rho(\boldsymbol{\tau})$ in (8) is called *isotropic* because the covariance function depends on $\|\boldsymbol{\tau}\|$ but not a specific direction. This represents a much stronger condition than stationarity [41]. Indeed, the covariance function of a stationary random field can take different values along different directions.

3. A stochastic realization approach

Let $\mathbf{z} = (z_1, \dots, z_d)$ be a vector of indeterminates. Consider a d-dimensional discrete-“time” linear stochastic system as depicted in Fig. 1, where

$$W(\mathbf{z}) = \sum_{\mathbf{k}\in\mathbb{Z}^d} \gamma(\mathbf{k})\mathbf{z}^{-\mathbf{k}} \tag{9}$$

is the transfer function, also called a shaping filter in the signal processing literature. Here the function $\gamma : \mathbb{Z}^d \to \mathbb{R}$ is called the *impulse response* of the system and $\mathbf{z}^{\mathbf{k}}$ is a shorthand notation for $z_1^{k_1}\cdots z_d^{k_d}$. Moreover, the symbol $\mathbf{z}^{-\mathbf{k}}$ can be interpreted as a $\mathbf{k}$-step delay operator. The system is excited by a normalized white noise $w(\mathbf{x})$ such that for any $\mathbf{x} \in \mathbb{Z}^d$,

$$\mathbb{E}[w(\mathbf{x})] = 0 \quad \text{and} \quad \mathbb{E}[w(\mathbf{x}+\mathbf{k})w(\mathbf{x})] = \delta_{\mathbf{k},\mathbf{0}} = \begin{cases} 1 & \text{if } \mathbf{k} = \mathbf{0}, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The output $y(\mathbf{x})$ is a zero-mean stationary random field. Symbolically, we write

$$y(\mathbf{x}) = W(\mathbf{z})w(\mathbf{x}) := \sum_{\mathbf{k}\in\mathbb{Z}^d} \gamma_{\mathbf{k}} w(\mathbf{x}-\mathbf{k}). \tag{11}$$

Notice that this is a standard model for stationary processes which goes back to the prediction theory of Wiener in the 1940s.

Let $\sigma(\mathbf{k}) := \mathbb{E}[y(\mathbf{x}+\mathbf{k})y(\mathbf{x})]$ be the covariance function of $y(\mathbf{x})$ and let $\mathbb{T} := [-\pi, \pi)$ denote the frequency interval. Then the spectral density of $y(\mathbf{x})$ is by definition [3,25] the multidimensional discrete-time Fourier transform (DTFT) of the covariance function:

$$\Phi(e^{i\boldsymbol{\theta}}) = \sum_{\mathbf{k}\in\mathbb{Z}^d} \sigma(\mathbf{k})\, e^{-i\langle \mathbf{k},\, \boldsymbol{\theta}\rangle}, \tag{12}$$

where the frequency vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d) \in \mathbb{T}^d$, $e^{i\boldsymbol{\theta}} := (e^{i\theta_1}, \dots, e^{i\theta_d})$ is a point on the d-torus (which is isomorphic to $\mathbb{T}^d$), and $\langle \mathbf{k},\ \boldsymbol{\theta}\rangle := k_1\theta_1 + \cdots + k_d\theta_d$ is the standard inner product in $\mathbb{R}^d$. It then follows from the spectral theory of stationary random fields [41] that

$$\Phi(e^{i\boldsymbol{\theta}}) = |W(e^{i\boldsymbol{\theta}})|^2 \tag{13}$$

where $W(e^{i\boldsymbol{\theta}}) = \sum_{\mathbf{k}\in\mathbb{Z}^d} \gamma(\mathbf{k})\, e^{-i\langle \mathbf{k},\, \boldsymbol{\theta}\rangle}$, so $\Phi(e^{i\boldsymbol{\theta}})$ takes *nonnegative* values. On the other hand, if the spectral density Φ satisfies certain analytic properties, then it admits a *spectral factor* W, see [25].

We are mostly interested in the case where $W(\mathbf{z})$ is a *rational* function, that is, it can be expressed as a ratio of two polynomials:

$$W(\mathbf{z}) = \frac{b(\mathbf{z})}{a(\mathbf{z})} = \frac{\sum_{\mathbf{k}\in\Lambda_{+,2}} b_{\mathbf{k}}\mathbf{z}^{-\mathbf{k}}}{\sum_{\mathbf{k}\in\Lambda_{+,1}} a_{\mathbf{k}}\mathbf{z}^{-\mathbf{k}}}, \tag{14}$$

where

$$\Lambda_{+,1} := \{(k_1, \dots, k_d) \in \mathbb{Z}^d : 0 \le k_j \le m_j,\ j = 1, \dots, d\},$$
$$\Lambda_{+,2} := \{(k_1, \dots, k_d) \in \mathbb{Z}^d : 0 \le k_j \le n_j,\ j = 1, \dots, d\}$$

are two index sets with positive integers m_j, n_j given for $j = 1, \dots, d$. Then the system (11) can equivalently be described in the time domain as an autoregressive moving-average (ARMA) model

$$\sum_{\mathbf{k}\in\Lambda_{+,1}} a_{\mathbf{k}}\, y(\mathbf{x}-\mathbf{k}) = \sum_{\mathbf{k}\in\Lambda_{+,2}} b_{\mathbf{k}}\, w(\mathbf{x}-\mathbf{k}). \tag{15}$$

Such a model is extremely useful in practice because rational functions (in fact, polynomials) can approximate any continuous function if the model order is sufficiently large. If the moving-average part of the model is trivial, i.e., $b(\mathbf{z}) \equiv b_{\mathbf{0}}$ is a constant, then (15) reduces to a simpler AR model.

In the above context, Problem 1 can be posed more concretely as follows:

Problem 2. Given a sampled covariance function $\rho_s(\mathbf{k})$ in (3), find a rational filter $W(\mathbf{z})$ of the form (14) such that when it is fed with a normalized white noise, the covariance function of the output $y(\mathbf{x})$ coincides with $\rho_s(\mathbf{k})$. Equivalently, we seek a rational spectral density satisfying the trigonometric moment equations

$$\int_{\mathbb{T}^d} e^{i\langle \mathbf{k},\,\boldsymbol{\theta}\rangle} \Phi(e^{i\boldsymbol{\theta}})\mathrm{d}\mu(\boldsymbol{\theta}) = \rho_{\mathrm{s}}(\mathbf{k}) \quad \forall \mathbf{k} \in \mathbb{Z}^d \tag{16}$$

where $\mathrm{d}\mu(\boldsymbol{\theta}) = \frac{1}{(2\pi)^d}\prod_{j=1}^d \mathrm{d}\theta_j$ is the normalized Lebesgue on $\mathbb{T}^d$.

Notice that the equivalence above is understood modulo the spectral factorization (13). Once the above problem is solved, samples of the random field $y_{\mathrm{s}}(\mathbf{x})$ for $\mathbf{x}$ indexed in (4) can be generated efficiently via the ARMA recursion (15) with a white noise input.

Next, in view of the decoupling assumption (3), it follows easily from the multidimensional DTFT that the corresponding spectral density $\Phi(e^{i\boldsymbol{\theta}})$ also has a decoupled form

$$\Phi(e^{i\boldsymbol{\theta}}) = \Phi_1(e^{i\theta_1}) \dots \Phi_d(e^{i\theta_d}), \tag{17}$$

where the factor $\Phi_j(e^{i\theta_j})$ can be interpreted as the spectral density in the jth dimension for $j = 1, \dots, d$, i.e., it is the 1-d DTFT of the sampled covariance function $\rho_{\mathrm{s},j}(k_j) := \rho_j(k_jT_j)$. Therefore, the d-dimensional moment Eqs. (16) decouple into d sets of unidimensional moment equations

$$\int_{\mathbb{T}} e^{ik_j\theta_j}\Phi_j(e^{i\theta_j})\frac{\mathrm{d}\theta_j}{2\pi} = \rho_{\mathrm{s},j}(k_j) \quad \forall k_j \in \mathbb{Z},\ j = 1, \dots, d, \tag{18}$$

whose solutions have been extensively studied in the literature. After each Φ_j has been constructed from the covariance function $\rho_{\mathrm{s},j}$, we can perform the spectral factorization $\Phi_j(z_j) = W_j(z_j)W_j(z_j^{-1})$ to obtain the transfer function $W_j(z_j)$. Notice that since Φ_j is constrained to be rational, the spectral factorization reduces to that for positive trigonometric polynomials, for which there are a number of algorithms [42]. Hence, the above procedure leads to a d-dimensional ARMA model

$$y(\mathbf{x}) = \underbrace{W_1(z_1) \dots W_d(z_d)}_{=:W(\mathbf{z})} w(\mathbf{x}) \tag{19}$$

again in a decoupled form. The main steps of our approach for sampling stationary random fields are summarized as follows.

(1) Given a sampled covariance function (3), solve the decoupled moment Eqs. (18) for a rational spectral density of the form (17).
(2) Do spectral factorization to obtain the linear filter in (19).
(3) Feed the filter with a Gaussian i.i.d. white noise and collect the output random field.

At the end of this section, let us discuss the stationarity and Gaussianity of the output random field $y(\mathbf{x})$ in (19). Since the input noise is white, it is well known in the theory of linear stochastic systems [25] that the output process/field is stationary, and this property does not require Gaussianity of the input. The Gaussianity of the output does follow from that of the input since the input–output relation can be viewed as an infinite-dimensional linear mapping where the convergence is understood in the mean square sense. It is then a textbook result that mean square convergence implies convergence in distribution (to a multivariate normal distribution), see e.g., [43].

4. Solution to the decoupled moment equations

In this section, we focus on the nontrivial Step 1 above, i.e., solution to the decoupled moment Eqs. (18), with two types of covariance functions mentioned before, i.e., the exponential and squared exponential covariance functions. It is worth remarking that the case of the exponential covariance function admits an *exact* rational spectrum and hence a shaping filter in a *closed form*. Although the squared exponential covariance function does not lead to an analytic solution, it can be well *approximated* in the frequency domain by a rational spectral density in the sense that only low-order covariances with significant values are matched in (18).

4.1. Solution for the exponential covariance function: an $ar(1)$ model

Under the decoupling assumption for the d-dimensional covariance function, we only need to solve Problem 2 for each sampled covariance function $\rho_{\mathrm{s},j}(k_j) = \rho_j(k_jT_j)$ of one variable, as discussed previously. Hence, we suppress the subscript j. Consider first the exponential covariance function in (6), i.e., $\rho(x) = \sigma^2e^{-\alpha|x|}$ with $x \in \mathbb{R}$. We can for simplicity take $\sigma^2 = 1$ since it is only a multiplicative constant. Let $T > 0$ be the sampling distance, and we have

$$\rho_{\mathrm{s}}(k) := \rho(kT) = e^{-\alpha T|k|}, \quad k \in \mathbb{Z}. \tag{20}$$

Let $r = e^{-\alpha T}$. Using the DTFT pair

$$x(k) = r^ku(k) \overset{\mathcal{F}}{\mapsto} X(e^{i\theta}) = \frac{1}{1-re^{-i\theta}} \tag{21}$$

with $0 < |r| < 1$ and $u(k)$ the discrete-time unit step function, the spectral density corresponding to $\rho_{\mathrm{s}}(k)$ is

$$\Phi(e^{i\theta}) = Z(e^{i\theta}) + Z(e^{i\theta})^* = 2\mathrm{Re}\{Z(e^{i\theta})\}, \tag{22}$$

where $*$ denotes complex conjugate and

$$Z(e^{i\theta}) = \frac{1}{1-e^{-\alpha T}e^{-i\theta}} - \frac{1}{2} \tag{23}$$

is the so-called *positive real* part of $\Phi(e^{i\theta})$. Notice that $Z(e^{i\theta})$ admits an *analytic extension* as a rational function

$$Z(z) = \frac{1}{1-e^{-\alpha T}z^{-1}} - \frac{1}{2}, \quad z \in \mathbb{C}, \tag{24}$$

so that $\Phi(z)$ in (22) is rational as well. Then after some straightforward calculations, we obtain a transfer function

$$W(z) = \frac{(1-e^{-2\alpha T})^{\frac{1}{2}}}{1-e^{-\alpha T}z^{-1}} \tag{25}$$

which is a *stable and minimum-phase* spectral factor of $\Phi(z)$, namely

$$\Phi(z) = W(z)W(z^{-1}) \tag{26}$$

where the identity is valid in a neighborhood of the unit circle. We see that $W(z)$ corresponds to an AR model of order one that depends on the parameter $\alpha_{\mathrm{s}} := \alpha T$. In consequence, samples of the discrete random field $y_{\mathrm{s}}(\mathbf{x})$ can be generated by filtering a white noise through d cascaded AR(1) models, one for each dimension. Obviously, the algorithm achieves a very low computational cost because each AR component has only order one. Compared with [24], our approach has a more concise frequency-domain interpretation and a neater time-domain implementation that avoids factorization of large matrices. Moreover, the model can be reused for computing multiple realizations of the random field because the filter $W(z)$ remains unchanged with a given covariance function when the sampling distance is fixed. Although the Cholesky factor in the Stepwise CMD method [24] can also be reused, our method requires much less storage since only the filter coefficients need to be stored. In addition, our model can be used to compute samples of an *arbitrary* size. On the contrary, the CMD has to be redone from the start if the sample size changes. A comparison of the computational complexity between different algorithms will be given at the end of this section.

Table 1
The computational complexity of different methods in the 3-d case. In the particular example, we have $\mathbf{N} = (100, 100, 100)$, $\mathbf{C} = (512, 512, 512)$, $\mathbf{M} = 1.1\mathbf{N}$, and an exponential covariance function.

Methods	Major computational complexity	Example (flops)
CMD	$O(N_1^3 N_2^3 N_3^3)$	1.00×10^{18}
Stepwise CMD [24]	$O(N_1 N_2 N_3 (N_1 + N_2 + N_3))$	3.00×10^{8}
Circulant Embedding [17]	$O(C_1 C_2 C_3 \log_2(C_1 C_2 C_3))$	3.62×10^{9}
Stochastic Realization	$O(N_1 N_2 N_3)$	1.06×10^{7}

4.2. Solution for the squared exponential covariance function: An ARMA model

Unlike the case with an exponential covariance function, the problem with a general covariance function may not have an exact analytic solution. In this subsection, we discuss the case with an squared exponential covariance function in (7). The corresponding sampled version is

$$\rho_s(k) := \rho(kT) = \sigma^2 e^{-\alpha T^2 |k|^2}, \quad k \in \mathbb{Z}. \tag{27}$$

It is well known that the (continuous-time) Fourier transform of a Gaussian function is another Gaussian function which is certainly nonrational. We speculate that the same happens for the discrete samples of a Gaussian function, i.e., the corresponding spectral density is *not* a rational function. Therefore, an ARMA representation for such a covariance function must by nature be approximate.

The Gaussian function (7) decays fast as $|x|$ increases. So a natural idea is to construct a rational spectral density

$$\Phi(e^{i\theta}) = \frac{P(e^{i\theta})}{Q(e^{i\theta})} \tag{28}$$

that matches a *finite* number of low-order (dominant) covariances, where P and Q are positive symmetric trigonometric polynomials. There are a number of solution techniques for this *rational covariance extension* problem, see e.g., [27,44–49]. In the paper, we adopt the following *generalized maximum entropy* formulation [1,27]:

$$\begin{aligned} &\max_{\Phi>0} \int_{\mathbb{T}} P(e^{i\theta}) \log \Phi(e^{i\theta}) \frac{d\theta}{2\pi} \\ &\text{s.t.} \quad \sigma_k = \int_{\mathbb{T}} e^{ik\theta} \Phi(e^{i\theta}) \frac{d\theta}{2\pi} \quad \forall k \in \Lambda, \end{aligned} \tag{29}$$

where,

- P is a *known* positive symmetric polynomial which is constructed from a *given* factor $b(z) := \sum_{k=0}^{n} b_k z^{-k}$, i.e., $P(z) = b(z)b(z^{-1})$;
- the index set is defined as $\Lambda := \{-m, \dots, -1, 0, 1, \dots, m\}$ such that m is a user-specified positive integer,
- σ_k's are the covariance data evaluated from the covariance function, namely $\sigma_k = \rho_s(k)$.

More precisely, we choose m to be the smallest positive integer such that $\rho_s(k)$ is practically zero for all $k > m$. In other words, the approximation procedure takes into account the covariances with significant values and discards the rest. The optimization problem (29) is convex and has a unique solution $\Phi = P/\hat{Q}$, where $\hat{Q}$ is the optimal solution of the dual problem

$$\min_{Q>0} \langle \boldsymbol{\sigma}, \mathbf{q} \rangle - \int_{\mathbb{T}} P(e^{i\theta}) \log Q(e^{i\theta}) \frac{d\theta}{2\pi}, \tag{30}$$

where $\mathbf{q} := \{q_k\}_{k\in\Lambda}$ are Lagrange multipliers, $\langle \boldsymbol{\sigma}, \mathbf{q} \rangle := \sum_{k\in\Lambda} \sigma_k q_k$ denotes the inner product, and $Q(e^{i\theta}) := \sum_{k=-m}^{m} q_k e^{-ik\theta}$ is a symmetric trigonometric polynomial. For technical details we refer readers to [1,27]. The reason for choosing this formulation is that we want to obtain a rational $\Phi(e^{i\theta})$ of the form (28) which is directly connected to the ARMA model via spectral factorization. More precisely, the polynomial $a(z) = \sum_{k=0}^{m} a_k z^{-k}$ corresponding to the AR coefficients can be determined via the Bauer method [42] for factoring $\hat{Q}(z) = \sum_{k=-m}^{m} \hat{q}_k z^{-k}$ where $\hat{q}_k$'s are the optimal Lagrange multipliers.

In this way, once the rational spectral density $\Phi(z)$ and the filter

$$W(z) = \frac{b(z)}{a(z)} = \frac{\sum_{k=0}^{n} b_k z^{-k}}{\sum_{k=0}^{m} a_k z^{-k}} \tag{31}$$

are constructed in each dimension, samples of the random field $y_s(\mathbf{x})$ can be generated in the same fashion as described in the previous subsection, now via the cascaded ARMA recursions. Each ARMA recursion has a fixed computational cost (though larger than that of the AR(1) model) related to the model order (m, n) which is chosen small.

4.3. Computational complexity analysis

In this subsection, we give the computational complexity of our approach to Problem 2. For practical applications, we are primarily interested in the 3-d case. Assume that we are asked to generate samples of a random field with a size $\mathbf{N} = (N_1, N_2, N_3) \in \mathbb{Z}^3$, see (4), and each ARMA model in the cascade has order (m_j, n_j) for $j = 1, 2, 3$. It is then common practice to generate samples of a slightly larger size $\mathbf{M} = (M_1, M_2, M_3)$ which can be taken as $(1+\beta)\mathbf{N}$, say with $\beta = 0.1$, in order to reduce the "transient" effect of filtering caused by an artificial boundary condition. For the exponential covariance function, the computational cost of the algorithm is proportional,[2] to the product $M_1 M_2 M_3$, so the complexity is $O(N_1 N_2 N_3)$ in which we have absorbed the constant $1 + \beta$ into the capital O notation. For the squared exponential covariance function which corresponds to an ARMA model, the algorithm still mainly runs in $O(N_1 N_2 N_3)$ flops because the computational cost of solving a small-size convex optimization problem (30) is far lower than that of implementing the ARMA recursion.

In Table 1 we state the computational costs of different methods: traditional CMD by Cholesky decomposition, Stepwise CMD, Circulant Embedding, and our stochastic realization approach, where an instance with specific numbers is also shown for clarity. It can be seen that our method has the lowest computational complexity which is *linear* in the number of samples.

Remark 2 (*Comparison with PDE-based sampling approaches*)**.** We have not compared with PDE-based sampling approaches as reported in [11,12,22,23] because they aim to generate samples of

[2] In order to generate one sample in the one-dimensional case using the rational filter (25) one simply rewrites the filtering equation as $y(t) = ay(t-1) + bw(t)$ where a and b are the filter coefficients, and computes $y(t)$ using $y(t-1)$ (either previously generated or known as an initial/boundary condition) and the input $w(t)$ (i.i.d. Gaussian noise). Hence the cost for one sample is 2 floating-point multiplications. The case for the ARMA filter (31) is similar, and so is the multidimensional version as long as the decoupling condition is met so that the filtering operation can be carried out along each dimension as illustrated in (19).

random fields with general Matérn-type covariance functions in (8), while we work under the decoupling assumption (1). The general sampling problem via the solution of stochastic PDEs is sophisticated and substantial computational power (e.g., a supercomputer) is needed for the numerical procedure. In contrast, in the special case where the covariance function can be decoupled in each dimension, our stochastic realization approach is easy to implement on a PC and offers the state-of-the-art time complexity.

5. Application to multiscale simulations

In some applications, e.g., the analysis of certain geomaterial properties [5,50], we often face a challenging problem that *refinements* of the generated samples of the random field are needed across several scales, where the spatial variability, i.e., the covariance function is expected to be maintained. Here by "a fine-scale simulation", we mean obtaining samples of the random field on a denser grid by *interpolating* the already generated coarse-scale realization, instead of generating fine-scale samples from scratch. Our approach is particularly suitable for such a purpose as we shall describe shortly.

For simplicity, we assume that the sampling distance T (which is the scale parameter) in (20) and (27) is halved in each fine-scale simulation, i.e., $T_1 = \frac{1}{2}T$, $T_2 = \frac{1}{2}T_1 = \frac{1}{4}T$, where the subscript denotes the number of fine-scale simulations that we perform. We want to point out that the model parameters in (25) and (31) in general *change* across different simulation scales since they depend on T, although the model order one is maintained in the case of an exponential covariance function. The change of the AR coefficients in (31) is more implicit and comes from the fact that the covariance data σ_k's are changed in a fine-scale simulation. In any case, once the fine-scale model (25) or (31) is constructed, the interpolation of the samples of the random field can be carried out by the same filtering technique as in the generation of the coarse-scale samples, *when* suitable boundary conditions and the fine-scale noise input (which is not i.i.d. any more) are provided.

5.1. The boundary conditions

The determination of such boundary conditions is rather standard. Suppose that the coarse-scale samples of the random field are collected into a column vector $\mathbf{y}_1$ and the *boundary* random variables that are needed to initiate the fine-scale filtering are collected in $\mathbf{y}_2$. Here we want to remark on the special 1-d case with the exponential covariance function, where the coarse-scale samples are themselves boundary conditions for the fine-scale simulation due to the AR(1) model structure. Hence, all one needs to do is to generate the white noise samples on the new grid points and to implement the filtering. In general however, some boundary values of the fine-scale samples in $\mathbf{y}_2$ are needed in order to start the multidimensional filtering, as shown in Fig. 2 for the 2-d case where the two cascaded filters are both AR(1).

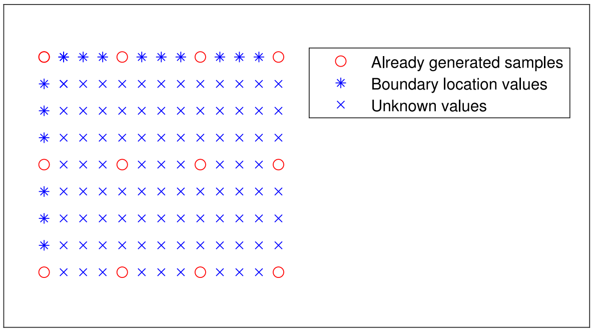


Fig. 2. A schematic figure for the boundary conditions in a multiscale simulation, where the number of values in boundary locations represents only a small fraction of the total number of unknown variables.

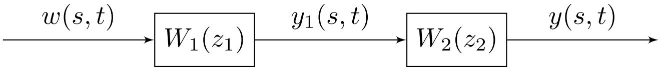


Fig. 3. The two cascaded linear stochastic system with given white noise input.

Let $\mathbf{y}$ be the joint vector such that

$$\mathbf{y} = \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}), \tag{32}$$

where we have introduced explicitly the Gaussianity assumption for the random field, and the matrix on the right side is the covariance matrix $\boldsymbol{\Sigma}$ of $\mathbf{y}$ which is evaluated using the given covariance function[3] $\rho(\boldsymbol{\tau})$ in (1) and partitioned in accordance with $\mathbf{y}_1$ and $\mathbf{y}_2$. Then the unknown random vector $\mathbf{y}_2$ conditioned on $\mathbf{y}_1 = \mathbf{a}$ still has a multivariate normal distribution $(\mathbf{y}_2|\mathbf{y}_1 = \mathbf{a}) \sim \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$ where

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\mathbf{a}, \tag{33}$$

and

$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}. \tag{34}$$

The matrix $\bar{\boldsymbol{\Sigma}}$ is known as the *Schur complement* of $\boldsymbol{\Sigma}_{11}$ in $\boldsymbol{\Sigma}$. It is positive definite because so is $\boldsymbol{\Sigma}$. Thus, the unknown boundary values $\mathbf{y}_2$ in a fine-scale simulation can be computed via:

$$\mathbf{y}_2 = R\mathbf{e} + \bar{\boldsymbol{\mu}}, \tag{35}$$

where the matrix R constitutes a *rank factorization* of $\bar{\boldsymbol{\Sigma}}$, namely $\bar{\boldsymbol{\Sigma}} = RR^\top$ which can be computed from the spectral decomposition plus truncation, and $\mathbf{e}$ is an i.i.d. standard normal vector.

5.2. The noise input

Obviously, samples on the fine-scale depend upon the white noise input which cannot be randomly generated any more because of the existence of the coarse-scale samples as interpolation conditions. Thus it is reasonable to *determine* the white noise before implementing ARMA recursion, and we report a particular solution for the case with exponential covariance functions, namely,

$$\rho_s(k_1, k_2) = \sigma^2 e^{-\alpha_1 T_1 |k_1| - \alpha_2 T_2 |k_2|}, \quad (k_1, k_2) \in \mathbb{Z}^2.$$

Here we focus on 2-d random fields which are of engineering interest in e.g., [5,50], also for the simplicity of presentation. In principle, a similar procedure should work for higher dimensional cases but the calculations will necessarily be more complicated.

Assume that we have computed the fine-scale samples in boundary locations, and we know the coarse-scale white noise and samples which have a size (N_1, N_2). We aim to interpolate the fine-scale input $w(s,t)$ and then samples $y(s,t)$ for $0 \le s \le 2(N_1 - 1)$, $0 \le t \le 2(N_2 - 1)$. In particular, the indices s and t of the coarse-scale samples are *even* numbers in the fine-scale realization. Referring to Fig. 3, the two cascaded AR(1) filters of the fine-scale random field are written as

$$W_1(z_1) = \frac{b}{1 - az_1^{-1}}, \quad W_2(z_2) = \frac{d}{1 - cz_2^{-1}}.$$

[3] Recall that the locations of samples in $\mathbf{y}$ are known.

The input w, the intermediate output y_1, and the output y are related via

$$y_1(s,t) = W_1(z_1)w(s,t), \quad y(s,t) = W_2(z_2)y_1(s,t).$$

Apply twice the AR recursion, and we have

$$w(s-1,t) = \frac{1}{ab}y_1(s,t) - \frac{a}{b}y_1(s-2,t) - \frac{1}{a}w(s,t). \tag{36}$$

If we take $s = 2k$ and $t = 2\ell$ with $k = 1,\dots,N_1-1$ and $\ell = 1,\dots,N_2-1$, $w(2k,2\ell)$ is known as the coarse-scale noise input. Then $w(2k-1,2\ell)$ can be computed directly via (36) in which

$$y_1(2k,2\ell) = \frac{1}{d'}y(2k,2\ell) - \frac{c'}{d'}y(2k-2,2\ell),$$

where c' and d' correspond to the filter $W_2'(z_2)$ of the coarse-scale random field, and the involved samples of y are from the coarse-scale realization. For odd $t = 2\ell-1$, $w(2k,2\ell-1)$ is unknown and (36) can be rewritten as a linear equation

$$A\mathbf{w} = \mathbf{b} \tag{37}$$

with

$$A = \begin{bmatrix} a & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & a & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a & 1 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w(1,t) \\ w(2,t) \\ \vdots \\ w(2N_1-2,t) \end{bmatrix},$$

$$\mathbf{b} = \frac{1}{b}\begin{bmatrix} y_1(2,t) \\ y_1(4,t) \\ \vdots \\ y_1(2N_1-2,t) \end{bmatrix} - \frac{a^2}{b}\begin{bmatrix} y_1(0,t) \\ y_1(2,t) \\ \vdots \\ y_1(2N_1-4,t) \end{bmatrix},$$

where the coefficient matrix A has a size $(N_1-1)\times(2N_1-2)$, and $y_1(2k,2\ell-1)$ is computed via

$$y_1(2k,2\ell-1) = \frac{1}{cd}y(2k,2\ell) - \frac{c}{d}y(2k,2\ell-2) - \frac{1}{c}y_1(2k,2\ell),$$

where again the right-hand side requires only the coarse-scale samples. Eq. (37) has infinitely many solutions since A is of full row rank. In order to make the problem well-posed, we notice a *dual* linear equation by swapping $W_1(z_1)$ and $W_2(z_2)$ in the filtering which obviously does not change the final output y. The intermediate output is, however, different and we write it as

$$y_2(2k,2\ell) = \frac{1}{b'}y(2k,2\ell) - \frac{a'}{b'}y(2k,2\ell-2)$$

where a' and b' correspond to the filter $W_1'(z_1)$ of coarse-scale random field. We can now compute the components $w(2k,2\ell-1)$ in the vector $\mathbf{w}$ as

$$w(2k,2\ell-1) = \frac{1}{cd}y_2(2k,2\ell) - \frac{c}{d}y_2(2k,2\ell-2) - \frac{1}{c}w(2k,2\ell)$$

in the same fashion as computing $w(2k-1,2\ell)$ from (36). Substitute the result back into (37), and we get the rest components.

After the boundary values of the fine-scale random field are computed and the white noise input is obtained, the interpolation of the coarse-scale samples can be accomplished again at a linear cost via the AR recursion. We would like to point out that unlike traditional methods which generate *all* the fine-scale samples (which could be a lot) using the conditional distribution (32), see e.g., [50, Sec. 2.4], we only need $\mathbf{y}_2$ in boundary locations (whose number is much smaller, see Fig. 2). Therefore, our procedure involves multiplications and inversions of matrices of smaller sizes, which significantly improves computational efficiency.

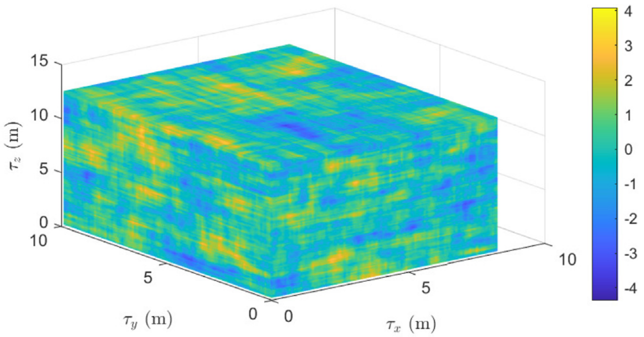


Fig. 4. A 3-d random field realization with an exponential covariance function. The size of the realization is 8.33 m ×10.00 m ×12.50 m in space with a total number of samples equal to $100^3 = 10^6$.

6. Numerical examples

In this section, we perform two sets of numerical simulations of our stochastic realization approach: one for sampling 3-d random fields and the other includes multiscale simulations in the 2-d case. All simulations are performed on a desktop computer with an Intel Core i7-10700K CPU and 16.0 GB of RAM.

6.1. Sampling 3-d Gaussian random fields

First, we apply the stochastic realization approach to Problem 2 with an exponential covariance function in three variables

$$\rho_s(x,y,z) = \sigma^2 e^{-\alpha_1 T_1|x| - \alpha_2 T_2|y| - \alpha_3 T_3|z|}, \quad (x,y,z)\in\mathbb{Z}^3. \tag{38}$$

In the following example, the size of the samples to be generated is $\mathbf{N} = (100, 100, 100)$ for the random field $y_s(x,y,z)$, the variance is $\sigma^2 = 1$, the parameter vector is $(\alpha_1,\alpha_2,\alpha_3) = (1,1,1)$, and the vector of sampling distances along x-, y-, and z-directions is $(T_1,T_2,T_3) = (1/12, 1/10, 1/8)$.

It has been discussed in Section 4.1 that the exponential covariance function corresponds to an exact rational spectral density, and the filter in each dimension can be directly computed via (25). Thus the required samples of the random field can be generated by implementing three cascaded AR recursions (19). The digital filtering can be carried out easily in Matlab via the `filter` command. For the 3-dimensional sampling, we just execute `filter` three times sequentially along all the dimensions. A realization of the random field is shown in Fig. 4 using the Matlab command `slice`, where the spatial distances are defined as $\tau_x = T_1|x|$, $\tau_y = T_2|y|$, and $\tau_z = T_3|z|$ along three directions. Next, in order to verify the performance of our method, we also plot the *sample covariances* of the realization $y_s(x,y,z)$ versus spatial distances along x-, y-, z-, and the diagonal directions in Fig. 5 with 100 repeated trials (a *Monte Carlo simulation*). The diagonal direction is along the line $x = y = z$ with $0 \le x \le N_1 - 1$.

In particular, the sample covariances of $y_s(x,y,z)$ are computed via the spatial average [35, Section 5]:

$$\hat{\sigma}_{\mathbf{k}} := \frac{1}{|\mathbf{N}|}\sum_{\mathbf{x}} y_s(\mathbf{x}+\mathbf{k})y_s(\mathbf{x}) \tag{39}$$

where $\mathbf{x} = (x,y,z)$ and $|\mathbf{N}| = N_1N_2N_3$. Since we have explicitly enforced covariance matching in Problem 2, it follows from the general covariance estimation theory [51] that the sample covariances of the output of the filter $W(\mathbf{z})$ must be close to the values of the given covariance function when the sample size is sufficiently large, and this point is well illustrated in Fig. 5. We

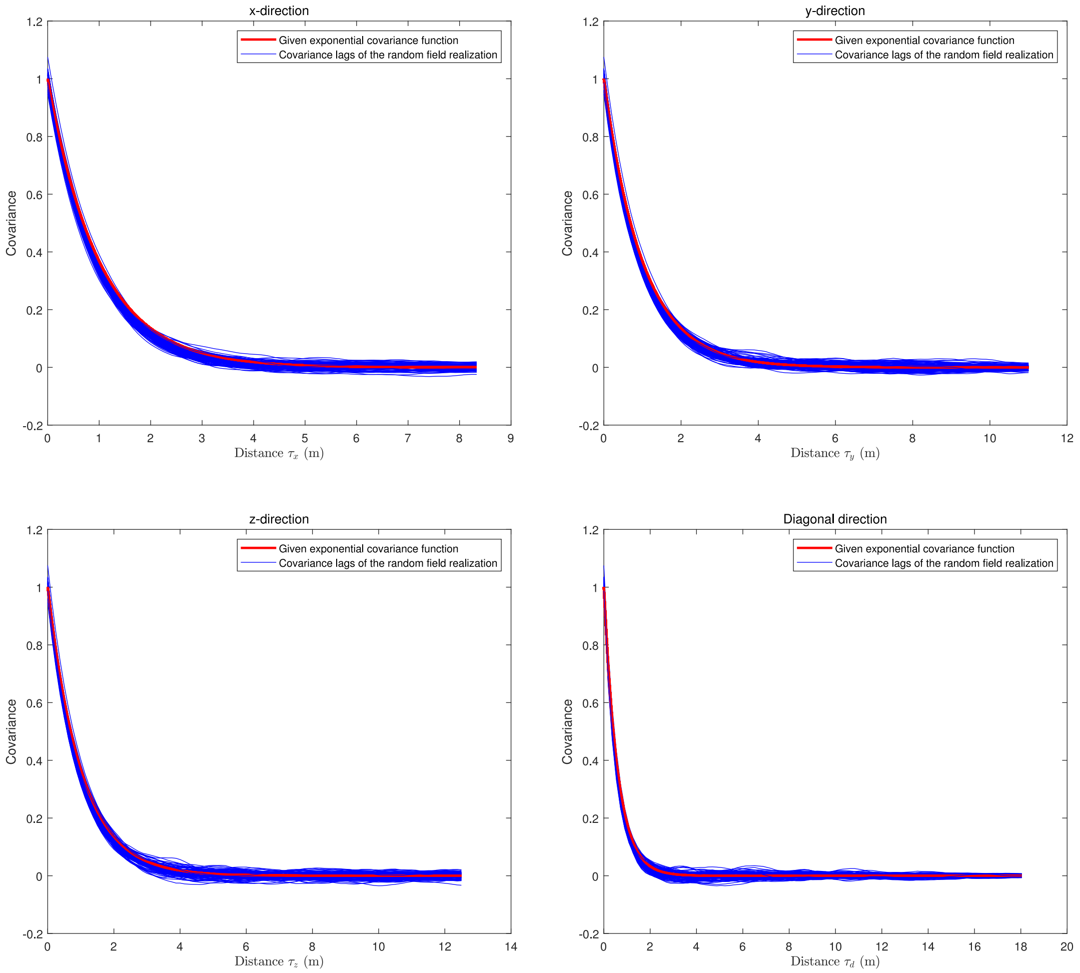


Fig. 5. The covariances versus distances along four directions of the 3-d random field realization which contains 100 repeated trials. The red line denotes the given exponential covariance function and the blue lines are the corresponding sample covariance lags. By x-direction, we mean the section $[k_1, 0, 0]$ of the sample covariance array with $k_1 = 0, \ldots, N_1 - 1$. The other directions are understood similarly. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

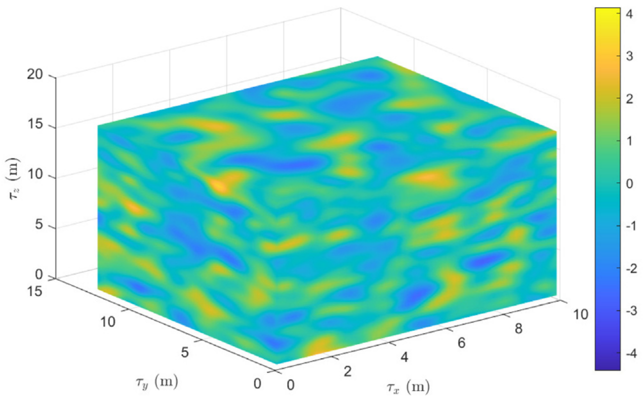


Fig. 6. A 3-d random field realization with a squared exponential covariance function. The size of the realization is 10 m × 12.50 m × 16.67 m in space with a total number of samples equal to $50^3 = 1.25 \times 10^5$.

remark that our result is visually much better than that reported in [24, Sec. 5.1].

Furthermore, another simulation is performed for a squared exponential covariance function that has the form

$$\rho_s(x, y, z) = \sigma^2 e^{-\alpha_1 T_1^2 |x|^2 - \alpha_2 T_2^2 |y|^2 - \alpha_3 T_3^2 |z|^2}, \quad (x, y, z) \in \mathbb{Z}^3. \tag{40}$$

The parameters are reported as follows: $\sigma^2 = 1$, $(\alpha_1, \alpha_2, \alpha_3) = (1, 1, 1)$, and $(T_1, T_2, T_3) = (1/5, 1/4, 1/3)$. In this case, we set up an ARMA model to approximate the underlying spectral density. The numerator polynomial $b(\mathbf{z})$ in (14) is specified by the user. Here for simplicity, we take $b(z_1, z_2, z_3) = \prod_{j=1}^{3} b_j(z_j)$ with a quite arbitrary $b_j(z_j) = 1 - 0.2z_j^{-1}$ of order one and identical for $j = 1, 2, 3$. The orders of the denominator polynomials $a_j(z_j)$ are chosen to be $(m_1, m_2, m_3) = (8, 7, 7)$ which is a threshold for "dominant" covariances, i.e., large values of the covariance function. Then the approximate spectrum in each dimension is constructed via solving the optimization problem (29), where Newton's method is implemented, and the polynomial $a_j(z_j)$ is obtained from spectral factorization. The results are shown in Figs. 6 and 7 similar to the previous two figures. Since the above squared exponential covariance function decays fast as the spatial distance increases, we set the sample size $\mathbf{N} = (50, 50, 50)$ in order to show more details in the figures. It can be seen that the sample covariances are very close to the given squared exponential covariance function when the lag is small. The mismatch for large lags can be explained by the fact that we are using a relatively low-order ARMA spectrum to approximate the nonrational Gaussian function. In fact, we can also use a higher-order ARMA model for a better approximation with e.g., $(m_1, m_2, m_3) = (16, 14, 14)$ and the result is displayed in Fig. 8. However, there is a price to pay as more computational effort is needed when

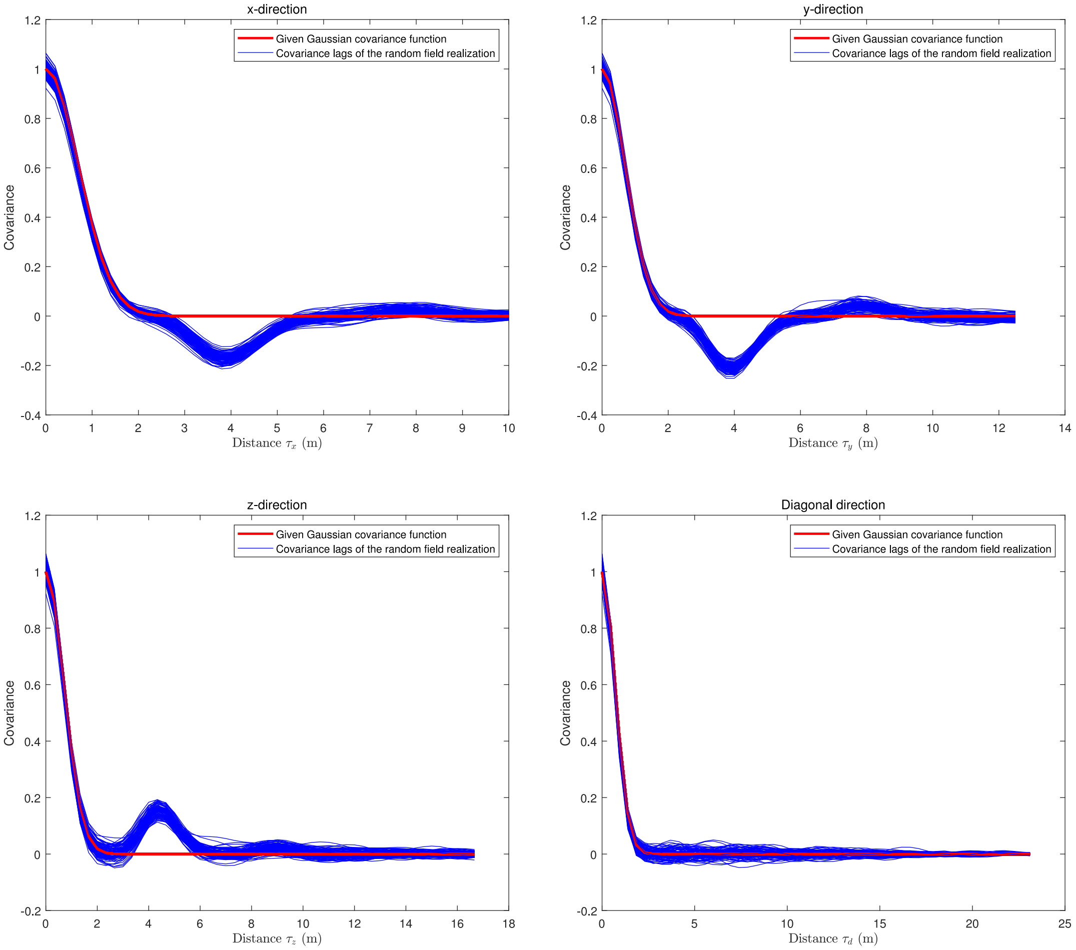


Fig. 7. The covariances versus distances along four directions of the 3-d random field realization which contains 100 repeated trials. The red line denotes the given squared exponential covariance function and the blue lines are the corresponding sample covariance lags. By x-direction, we mean the section $[k_1, 0, 0]$ of the sample covariance array with $k_1 = 0, \dots, N_1 - 1$. The other directions are understood similarly. The order of the approximate ARMA model is $\mathbf{m} = (8, 7, 7)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

generating the samples, that is, the constant hidden in the capital O notation for the operation count has (almost) doubled.

Remark 3. One can compare the generated samples (Figs. 4 and 6) from the exponential and squared exponential covariance functions, and find that the former realization is rough (seemingly discontinuous) while the latter is much smoother. This phenomenon can be understood via the *mean square differentiability* [13] of the underlying random fields on $\mathbb{R}^d$ with respective covariance functions. In particular, neither realization is inferior than the other as the smoothness is simply a property brought by the covariance function. Which one is favored depends on the specific application.

6.2. Comparison with the stepwise CMD

In this subsection, we perform several Monte Carlo simulations to compare the computational time between Stepwise CMD and our stochastic realization approach. The reason for such comparison with one single method is that according to [24], the Stepwise CMD outperforms other commonly used sampling methods including CMD, Circulant Embedding, and Karhunen–Loève. More specifically, we perform numerical simulations in 1-d and 3-d cases. The average computational time of 100 repeated trials versus the total number of nodes of the sampled random field is shown in Fig. 9 using the Matlab command `loglog`. In the 1-d case,[4] we have the exponential covariance function $\rho_s(x) = e^{-|x|/10}$ with $x \in \mathbb{Z}$, and the results are shown in the left panel of Fig. 9. The right panel corresponds to the 3-d case with the covariance function (38) and the same parameters described after the formula. In either case, the number of nodes (grid size) in the random field realization doubles for each point on the curve from left to right. More precisely for the 3-d case, we fix the grid size in two of the three dimensions equal to 100, and double the grid size in the third dimension every time. One can see that our stochastic realization approach requires significantly less computational time compared with the Stepwise CMD.

Remark 4. It is interesting to notice from Fig. 9 that in the 1-d case, the advantage of our stochastic realization approach against the Stepwise CMD is more appreciable, while in the 3-d case, such advantage is not so apparent until the grid size becomes very large. In view of Table 1, however, we expect our method to be better than the Stepwise CMD in terms of the computational time by at least one order of magnitude when the grid size $|\mathbf{N}| = 10^6$. Clearly we have not achieved it. The reason is that the CMD employs standard Linear Algebra packages which are specialized for matrix computation and are automatically *multithreaded* in Matlab.[5] In contrast, the filtering operation in our approach is by

[4] In the 1-d case, the Stepwise CMD is just CMD.

[5] See the online documentation https://www.mathworks.com/discovery/matlab-multicore.html.

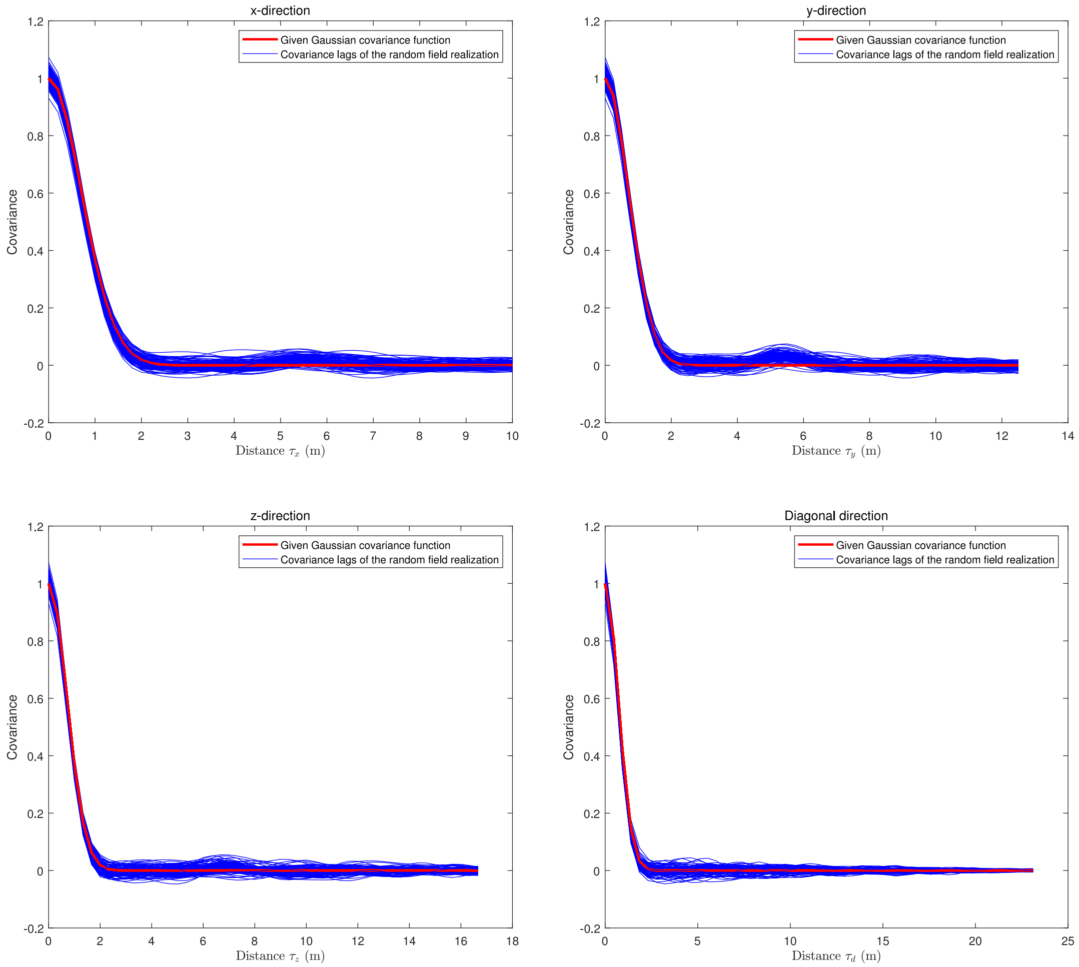
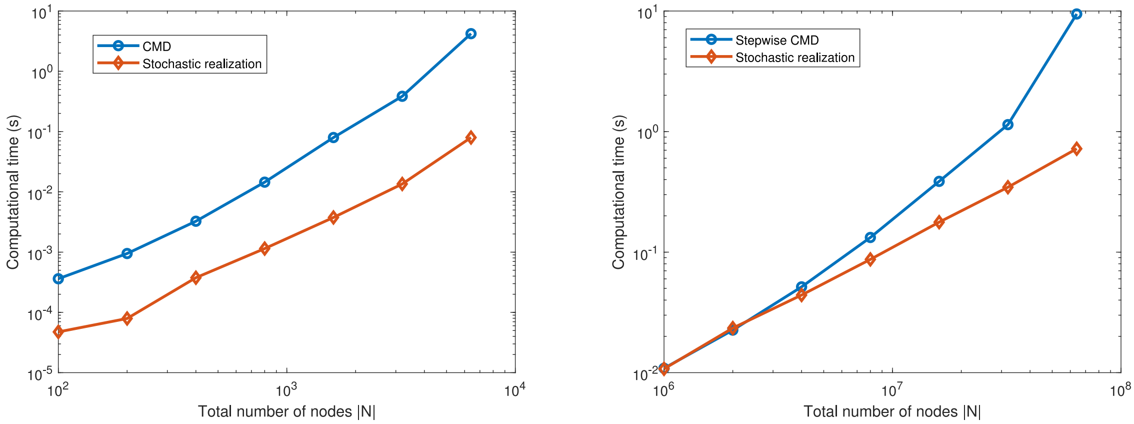


Fig. 8. The covariances versus distances along four directions of the 3-d random field realization which contains 100 repeated trials. The red line denotes the given squared exponential covariance function and the blue lines are the corresponding sample covariance lags. By x-direction, we mean the section $[k_1, 0, 0]$ of the sample covariance array with $k_1 = 0, \dots, N_1 - 1$. The other directions are understood similarly. The order of the approximate ARMA model is now $\mathbf{m} = (16, 14, 14)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) 1-d case

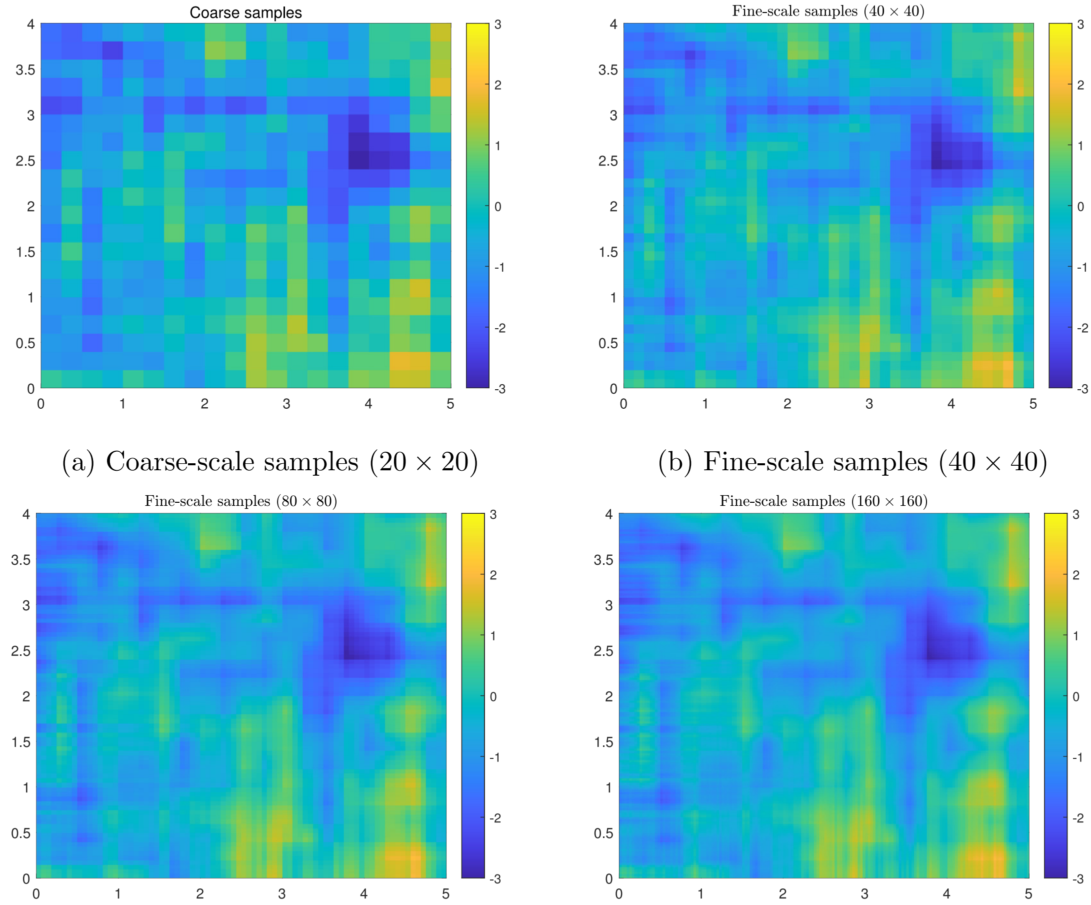
(b) 3-d case

Fig. 9. Stochastic Realization versus Stepwise CMD: the average computational time for one realization of the random field with an exponential covariance function versus the total number of nodes $|\mathbf{N}|$ in 1-d and 3-d cases. In both cases, the total number of nodes doubles for each point on the curve from left to right.

nature *serial* that is, samples of the random field are generated one after another. In consequence, the superiority of the stochastic realization approach displayed by Table 1 is mitigated by the algorithmic implementation when the grid size is relatively small.

6.3. Multiscale simulations in the 2-d case

In this subsection, we perform simulations to produce fine-scale samples of the random field given the coarse-scale realization using our stochastic realization approach. Following the



(a) Coarse-scale samples (20 × 20)

(b) Fine-scale samples (40 × 40)

(c) Fine-scale samples (80 × 80)

(d) Fine-scale samples (160 × 160)

Fig. 10. Multiscale simulations with an exponential covariance function. Subfig. (a) shows the coarse-scale samples (colored squares) which are already known, while Subfigs. (b), (c), and (d) are the fine-scale realizations, where the numbers in parentheses give the number of sampling points in each direction.

derivation in Section 5, we report an example with the following exponential covariance function

$$\rho_s(x, y) = \exp(-\frac{|x|}{5} - \frac{|y|}{4}), \quad (x, y) \in \mathbb{Z}^2.$$

Suppose that the coarse-scale samples of the random field have a size vector $(N_1, N_2) = (20, 20)$ which corresponds to 400 points. We take the sampling distances to be 1/5 and 1/4 along x- and y-directions, respectively, i.e., $\mathbf{T} = (T_1, T_2) = (1/5, 1/4)$, and the parameter vector $(\alpha_1, \alpha_2) = (1, 1)$. Then three fine-scale realizations of the random field are computed, where the sampling distances are reduced to 1/2, 1/4, and 1/8 of the original values, respectively. The simulations are carried out sequentially. More precisely, after the samples with the parameter $\mathbf{T}$ are generated, they are then treated as the coarse-scale realization, and the interpolation procedure is executed to produce fine-scale samples with the parameter $\mathbf{T}' = \frac{1}{2}\mathbf{T}$. Notice that we need to reconstruct the AR(1) filter under each scale, due to the fact that the filter parameters in (25) depend upon the product $\alpha_j T_j$. Then after the sample values in boundary locations are computed utilizing the conditional normal distribution and the white noise in unknown locations is obtained, the rest fine-scale samples can be generated by implementing the AR recursion. These operations are repeated for each finer scale. The result of such a multiscale simulation is shown in Fig. 10. It is evident that more details of the random field can be seen from the fine-scale samples than the coarse-scale realization. We also plot the sample covariances under each scale versus distances along x- and y-directions in Fig. 11. One can see that the sample covariances are again close to the given covariance function, which indicates that the spatial variability of the random field is well maintained across multiple scales in our simulations.

7. Conclusions

This paper proposes an efficient stochastic realization approach for sampling large-scale multidimensional Gaussian stationary random fields. The basic idea is to exploit the decoupling assumption on the covariance function, and to construct a rational model which approximates the spectrum of the underlying random field in terms of covariance matching. Moreover, our sampling approach features easy implementation and low computational complexity due to the simple structure of the approximate model. The work in the paper can be concluded as follows:

(1) Solutions to the sampling problem with the exponential and squared exponential covariance functions are given, respectively. The former corresponds to a rational spectral density that leads to an AR(1) filter, and the latter has a nonrational spectral density which can be approximated by an ARMA spectrum.

(2) The stochastic realization approach has been applied to multiscale simulations. Compared with traditional methods, only a few number of values in boundary locations are computed prior to the interpolation via the ARMA recursion, which achieves high efficiency.

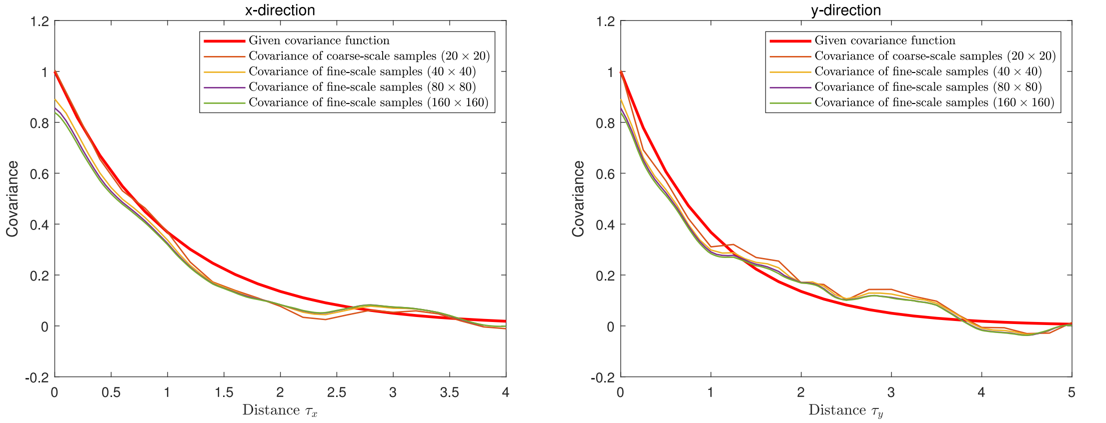


Fig. 11. Covariance lags of the random field versus distances. The red line denotes the given exponential covariance function, and the other four lines correspond to the sample covariances of under different scales. By x-direction, we mean the column of the sample covariance matrix indexed by $[k_1, 0]$ with $k_1 = 0, \ldots, N_1 - 1$. The y-direction is understood similarly. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(3) Several numerical simulations are performed and they show that our method exhibits good performances not only in sampling large-size random fields, but also in refining generated samples across multiple scales.

Finally, it is expected that our approach can be extended to the multivariate case (i.e., vector processes), which will be a future study.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under the grant number 62103453 and the "Hundred-Talent Program" of Sun Yat-sen University.

References

[1] Ringh A, Karlsson J, Lindquist A. Multidimensional rational covariance extension with applications to spectral estimation and image compression. SIAM J Control Optim 2016;54(4):1950–82.

[2] Zhu B, Zorzi M. A well-posed multidimensional rational covariance and generalized cepstral extension problem. SIAM J Control Optim 2023;61(3):1532–56.

[3] Stoica P, Moses R. Spectral analysis of signals. Upper Saddle River, NJ: Pearson Prentice Hall; 2005.

[4] Qian T, Zhang Y, Liu W, Qu W. Adaptive Fourier decomposition-type sparse representations versus the Karhunen–Loève expansion for decomposing stochastic processes. Math Methods Appl Sci 2023;46(13):14007–25.

[5] Chen Q, Wang C, Hsein Juang C. CPT-based evaluation of liquefaction potential accounting for soil spatial variability at multiple scales. J Geotech Geoenviron Eng 2016;142(2):04015077.

[6] Liu Y, Zhang W, Zhang L, Zhu Z, Hu J, Wei H. Probabilistic stability analyses of undrained slopes by 3D random fields and finite element methods. Geosci Front 2018;9(6):1657–64.

[7] Latz J, Eisenberger M, Ullmann E. Fast sampling of parameterised Gaussian random fields. Comput Methods Appl Mech Engrg 2019;348:978–1012.

[8] Yi JT, Huang LY, Li DQ, Liu Y. A large-deformation random finite-element study: failure mechanism and bearing capacity of spudcan in a spatially varying clayey seabed. Géotechnique 2020;70(5):392–405.

[9] Cheng P, Guo J, Yao K, Liu C, Liu X, Liu F. Uplift behavior of pipelines buried at various depths in spatially varying clayey seabed. Sustainability 2022;14(13):8139.

[10] Winkler G. Image analysis, random fields and Markov chain Monte Carlo methods: A mathematical introduction. Stochastic modelling and applied probability, second ed.. vol. 27, Springer Science & Business Media; 2003.

[11] Croci M, Giles MB, Rognes ME, Farrell PE. Efficient white noise sampling and coupling for multilevel Monte Carlo with nonnested meshes. SIAM/ASA J Uncertain Quantif 2018;6(4):1630–55.

[12] Khristenko U, Scarabosio L, Swierczynski P, Ullmann E, Wohlmuth B. Analysis of boundary effects on PDE-based sampling of Whittle–Matérn random fields. SIAM/ASA J Uncertain Quantif 2019;7(3):948–74.

[13] Williams CK, Rasmussen CE. Gaussian processes for machine learning. Adaptive computation and machine learning, vol. 2, Cambridge, MA: MIT Press; 2006.

[14] Fenton GA, Griffiths DV. Risk assessment in geotechnical engineering. John Wiley & Sons, New York; 2008.

[15] Liu Y, Lee F-H, Quek S-T, Beer M. Modified linear estimation method for generating multi-dimensional multi-variate Gaussian field in modelling material properties. Probab Eng Mech 2014;38:42–53.

[16] Vanmarcke E. Random fields: Analysis and synthesis. World Scientific; 2010.

[17] Graham IG, Kuo FY, Nuyens D, Scheichl R, Sloan IH. Analysis of circulant embedding methods for sampling stationary random fields. SIAM J Numer Anal 2018;56(3):1871–95.

[18] Dembo A, Mallows CL, Shepp LA. Embedding nonnegative definite Toeplitz matrices in nonnegative definite circulant matrices, with application to covariance estimation. IEEE Trans Inform Theory 1989;35(6):1206–12.

[19] Dietrich CR, Newsam GN. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. SIAM J Sci Comput 1997;18(4):1088–107.

[20] Zheng Z, Dai H. Simulation of multi-dimensional random fields by Karhunen–Loève expansion. Comput Methods Appl Mech Engrg 2017;324:221–47.

[21] Feischl M, Kuo FY, Sloan IH. Fast random field generation with H-matrices. Numer Math 2018;140(3):639–76.

[22] Osborn S, Vassilevski PS, Villa U. A multilevel, hierarchical sampling technique for spatially correlated random fields. SIAM J Sci Comput 2017;39(5):S543–62.

[23] Fairbanks HR, Villa U, Vassilevski PS. Multilevel hierarchical decomposition of finite element white noise with application to multilevel Markov chain Monte Carlo. SIAM J Sci Comput 2021;43(5):S293–316.

[24] Li D-Q, Xiao T, Zhang L-M, Cao Z-J. Stepwise covariance matrix decomposition for efficient simulation of multivariate large-scale three-dimensional random fields. Appl Math Model 2019;68:169–81.

[25] Lindquist A, Picci G. Linear stochastic systems: A geometric approach to modeling, estimation and identification. Series in contemporary mathematics, vol. 1, Springer-Verlag Berlin Heidelberg; 2015.

[26] Byrnes CI, Gusev SV, Lindquist A. A convex optimization approach to the rational covariance extension problem. SIAM J Control Optim 1998;37(1):211–29.

[27] Byrnes CI, Gusev SV, Lindquist A. From finite covariance windows to modeling filters: A convex optimization approach. SIAM Rev 2001;43(4):645–75.

[28] Enqvist P. A convex optimization approach to ARMA(n, m) model design from covariance and cepstral data. SIAM J Control Optim 2004;43(3):1011–36.

[29] Georgiou TT. Relative entropy and the multivariable multidimensional moment problem. IEEE Trans Inform Theory 2006;52(3):1052–66.

[30] Georgiou TT, Lindquist A. A convex optimization approach to ARMA modeling. IEEE Trans Automat Control 2008;53(5):1108–19.

[31] Ferrante A, Pavon M, Ramponi F. Hellinger versus Kullback–Leibler multivariable spectrum approximation. IEEE Trans Automat Control 2008;53(4):954–67.

[32] Lindquist A, Picci G. The circulant rational covariance extension problem: The complete solution. IEEE Trans Automat Control 2013;58(11):2848–61.

[33] Zhu B, Baggio G. On the existence of a solution to a spectral estimation problem *à la* Byrnes-Georgiou-Lindquist. IEEE Trans Automat Control 2019;64(2):820–5.

[34] Zhu B. On the well-posedness of a parametric spectral estimation problem and its numerical solution. IEEE Trans Automat Control 2020;65(3):1089–99.

[35] Zhu B, Ferrante A, Karlsson J, Zorzi M. M^2-spectral estimation: A relative entropy approach. Automatica 2021;125.

[36] Zhu B, Liu J. A fast robust numerical continuation solver to a two-dimensional spectral estimation problem. IET Control Theory Appl 2022;16(9):902–15.

[37] Liu J, Zhu B. An efficient stochastic realization approach for sampling Gaussian random fields with exponential covariance functions. In: Proceedings of the 23rd Chinese conference on system simulation technology and its application. 2022, p. 43–7.

[38] Zhu B, Zorzi M. On a weaker regularity condition for a multidimensional spectral estimation problem. IEEE Control Syst Lett 2023;7:1795–800.

[39] Firouzianbandpey S, Ibsen LB, Griffiths D, Vahdatirad M, Andersen LV, Sørensen JD. Effect of spatial correlation length on the interpretation of normalized CPT data using a Kriging approach. J Geotech Geoenviron Eng 2015;141(12):04015052.

[40] Ching J, Phoon K-K, Pan Y-K. On characterizing spatially variable soil Young's modulus using spatial average. Struct Saf 2017;66:106–17.

[41] Yaglom AM. Some classes of random fields in n-dimensional space, related to stationary random processes. Theory Probab Appl 1957;2(3):273–320.

[42] Sayed AH, Kailath T. A survey of spectral factorization methods. Numer Linear Algebra Appl 2001;8(6–7):467–96.

[43] Van der Vaart AW. Asymptotic statistics. Cambridge series in statistical and probabilistic mathematics, vol. 3, Cambridge University Press; 2000.

[44] Georgiou TT, Lindquist A. Kullback–Leibler approximation of spectral density functions. IEEE Trans Inform Theory 2003;49(11):2910–7.

[45] Ramponi F, Ferrante A, Pavon M. A globally convergent matricial algorithm for multivariate spectral estimation. IEEE Trans Automat Control 2009;54(10):2376–88.

[46] Ferrante A, Masiero C, Pavon M. Time and spectral domain relative entropy: A new approach to multivariate spectral estimation. IEEE Trans Automat Control 2012;57(10):2561–75.

[47] Zorzi M. A new family of high-resolution multivariate spectral estimators. IEEE Trans Automat Control 2014;59(4):892–904.

[48] Zorzi M. Rational approximations of spectral densities based on the Alpha divergence. Math Control Signals Systems 2014;26(2):259–78.

[49] Zhu B, Ferrante A, Karlsson J, Zorzi M. M^2-spectral estimation: A flexible approach ensuring rational solutions. SIAM J Control Optim 2021;59(4):2977–96.

[50] Chen Q, Seifried A, Andrade JE, Baker JW. Characterization of random fields and their impact on the mechanics of geosystems at multiple scales. Int J Numer Anal Methods Geomech 2012;36(2):140–65.

[51] Priestley MB. Spectral analysis and time series (two-volume set). Probability and mathematical statistics, Elsevier Academic Press; 1981, Reprinted in 2004.