Comput Geosci (2011) 15:673–689
DOI 10.1007/s10596-011-9235-5

ORIGINAL PAPER

Assessing the accuracy of sequential Gaussian simulation and cosimulation

Xavier Emery · María Peláez

Received: 15 October 2010 / Accepted: 31 March 2011 / Published online: 28 April 2011

Abstract Sequential Gaussian simulation is widespread in Earth Science applications to quantify the uncertainty about regionalized properties. Its practical implementation relies on the screen effect approximation in order to determine the successive conditional distributions by considering only the information available in the neighborhood of the target location. A methodology is presented to assess the accuracy of sequential Gaussian simulation, by calculating the theoretical moments (expectation and variance–covariance matrix) of the simulated random vector and comparing them with the moments of the underlying model. The methodology can be applied in both the conditional and non-conditional contexts, as well as for univariate or multivariate simulation. It is helpful to determine appropriate implementation parameters, in particular about the visiting sequence and the design of the moving neighborhood for selecting relevant conditioning information, prior to performing simulation.

Keywords Gaussian random fields · Moving neighborhood · Screen effect approximation · Visiting sequence · Collocated co kriging

X. Emery (✉)
Department of Mining Engineering /
Advanced Mining Technology Center, University of Chile,
Avenida Tupper 2069, Santiago, Chile
e-mail: xemery@ing.uchile.cl

M. Peláez
Department of Mathematics, Northern Catholic University,
Avenida Angamos 0610, Antofagasta, Chile
e-mail: mpelaez@ucn.cl

1 Introduction

The simulation of Gaussian random fields is widely used in ore body and reservoir modeling, hydrology, soil and environmental sciences for assessing the uncertainty in the unsampled values of regionalized properties, such as mineral grades, concentrations of contaminants, porosity, water saturation, permeability, electrical conductivity, rock density, hardness, metal recovery, or acid consumption [6, 9, 11, 12, 26, 43].

Gaussian simulation algorithms can be classified in two families:

1. Algorithms that simulate random fields whose finite-dimensional distributions are exactly multivariate Gaussian (except for round-off errors or for the use of pseudo-random numbers), e.g., LU decomposition of the covariance matrix [1, 10], sequential Gaussian [15, 24], convolution [5, 8], circulant-embedding [16], and discrete spectral simulation [7, 33].
2. Algorithms that simulate random fields whose finite-dimensional distributions are approximately multivariate Gaussian, on account of the central limit theorem, e.g., continuous spectral [38], turning bands [23, 31], or dilution [8, 30].

This work focuses on a specific algorithm of the first family, the sequential Gaussian, which is widely used in applications. To reduce computing costs and storage requirements when the target random field must be simulated at a large number of spatial locations, an approximate implementation is usually adopted, based on the selection of subsets of data to determine the successive conditional distributions. The selected data are the closest to the location targeted for simulation,

which are assumed to screen out the influence of the data located farther away [17, 24].

The screen effect approximation results in a loss of accuracy in the reproduction of the model statistics, which needs to be assessed [28]. Currently, most approaches for validating sequential simulation consist in comparing the experimental statistics of a finite set of realizations with the model statistics [18, 24, 32, 40] or with data statistics [29]. A drawback of these approaches is the difficulty in assessing whether or not the deviations between *experimental* and *model statistics* are significant. Indeed, even if simulation is perfectly accurate, the experimental statistics of a single realization are likely to fluctuate around the model statistics. To avoid drawing mistaken conclusions, statistical hypothesis testing can be made on a set of realizations, but this leaves open the questions of how many realizations should be generated and of what decision rule should be used to balance false positives (errors of the first kind) and false negatives (errors of the second kind): reducing one type of error generally results in increasing the other type of error [20].

In the following section, we propose a different approach consisting in determining the theoretical distribution of the simulated random field and contrasting this distribution with the underlying model distribution. The accuracy of the sequential simulation algorithm can then be measured by comparing *theoretical statistics*, without the need for generating a set of realizations, calculating experimental statistics, or performing hypothesis testing.

2 Theoretical distributions of random fields simulated with the sequential Gaussian algorithm

2.1 Non-conditional simulation

It is of interest to simulate a standard Gaussian random vector $\mathbf{Y} = (Y_1,\ldots Y_n)^T$ with given variance–covariance matrix $\mathbf{C_Y}$. This vector represents the variables of a Gaussian random field at a set of spatial locations targeted for simulation. In the following, $\mathbf{C_Y}$ is assumed to be positive definite, i.e., all its eigenvalues are positive. Otherwise, one should restrict the number of target locations so as to ensure positive definiteness (a zero eigenvalue occurs when one variable Y_i is a linear combination of the other variables Y_j with $j \neq i$, in which case the ith location can be removed).

The sequential Gaussian simulation algorithm consists in simulating each component of $\mathbf{Y}$ in turn and using it as conditioning information for the subsequent components [15, 24]. Denoting by $\tilde{\mathbf{Y}} = (\tilde{Y}_1,\ldots\tilde{Y}_n)^{\mathrm{T}}$ the simulated vector, one has:

$$\forall i = 1\ldots n,\ \tilde{Y}_i = \sum_{j=1}^{i-1} \lambda_{j,i}\tilde{Y}_j + \sigma_i U_i \tag{1}$$

where $\lambda_{j,i}$ is the simple kriging weight assigned to Y_j when predicting Y_i, σ_i the associated kriging standard deviation, and U_i a standard Gaussian random variable independent of $U_1,\ldots U_{i-1}$, $\tilde{Y}_1,\ldots \tilde{Y}_{i-1}$.

From Eq. 1, it is seen that any linear combination of the components of $\tilde{\mathbf{Y}}$ is a linear combination of the independent Gaussian random variables $U_1,\ldots U_n$ and, therefore, has a Gaussian distribution. Accordingly, $\tilde{\mathbf{Y}}$ is a Gaussian random vector and is fully characterized by its first- and second-order moments. Since $U_1,\ldots U_n$ have zero means, the mean vector of $\tilde{\mathbf{Y}}$ is zero, so that it remains to determine its variance–covariance matrix.

Because $\mathbf{C_Y}$ is positive definite, the kriging standard deviation σ_i is positive. Equation 1 can, therefore, be rewritten in the following fashion:

$$\forall i = 1\ldots n,\ U_i = \frac{\tilde{Y}_i - \sum_{j=1}^{i-1} \lambda_{j,i}\tilde{Y}_j}{\sigma_i} \tag{2}$$

or, equivalently:

$$\mathbf{U} = \boldsymbol{\Lambda}\tilde{\mathbf{Y}} \tag{3}$$

with $\boldsymbol{\Lambda} = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & \cdots & 0 \\ -\frac{\lambda_{1,2}}{\sigma_2} & \frac{1}{\sigma_2} & 0 & \\ \vdots & & \ddots & \vdots \\ -\frac{\lambda_{1,n-1}}{\sigma_{n-1}} & & -\frac{\lambda_{n-2,n-1}}{\sigma_{n-1}} & \frac{1}{\sigma_{n-1}} \quad 0 \\ -\frac{\lambda_{1,n}}{\sigma_n} & & -\frac{\lambda_{n-2,n}}{\sigma_n} & -\frac{\lambda_{n-1,n}}{\sigma_n} \quad \frac{1}{\sigma_n} \end{pmatrix}$ (lower triangular $n \times n$ matrix).

The components of $\mathbf{U}$ being mutually independent with unit variances, one has:

$$\mathbf{I} = E\left(\mathbf{U}\mathbf{U}^T\right) = \boldsymbol{\Lambda} E\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\right)\boldsymbol{\Lambda}^T \tag{4}$$

where $\mathbf{I}$ stands for the identity $n \times n$ matrix. Since the simulated vector $\tilde{\mathbf{Y}}$ has a zero mean, its variance–covariance matrix is

$$\mathbf{C}_{\tilde{\mathbf{Y}}} = E\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\right) = \boldsymbol{\Lambda}^{-1}\left(\boldsymbol{\Lambda}^{-1}\right)^T \tag{5}$$

If the kriging weights and standard deviation in Eq. 1 are calculated using all the previously simulated variables $\{\tilde{Y}_1,\ldots \tilde{Y}_{i-1}\}$ as conditioning data (*unique neigh-*

borhood implementation), the variance–covariance matrix $\mathbf{C}_{\tilde{\mathbf{Y}}}$ coincides with the model matrix $\mathbf{C}_{\mathbf{Y}}$ [15]. As $\boldsymbol{\Lambda}$ is triangular, Eq. 5 then coincides with the Cholesky decomposition of $\mathbf{C}_{\mathbf{Y}}$, which establishes equivalence between the sequential Gaussian and LU simulation algorithms [1, 10].

However, as n increases, using all the previously simulated variables as conditioning data in Eq. 1 becomes computationally prohibitive, so that simplifications of the sequential algorithm are necessary. In practice, kriging is performed with a *moving neighborhood*, taking only part of the previously simulated variables (usually the most correlated with the target variable) as conditioning data [24]. Equations 1–5 remain valid by assuming that the data outside the moving neighborhood are assigned zero weights, but the actual variance–covariance matrix of the simulated vector $\mathbf{C}_{\tilde{\mathbf{Y}}}$ (Eq. 5) is likely to differ from the prior model $\mathbf{C}_{\mathbf{Y}}$ [18, 32, 40].

The above statements provide a simple means to assess the accuracy of sequential simulation:

1. Define a visiting sequence (i.e., a permutation of {1,... n}) for simulation and order the components of $\mathbf{Y}$ accordingly.
2. Define a moving neighborhood: number of data, search radii and angular sectors [15].
3. According to the visiting sequence and moving neighborhood, calculate the simple kriging weights and standard deviations and define matrix $\boldsymbol{\Lambda}$ (Eq. 3).
4. Calculate the variance–covariance matrix $\mathbf{C}_{\tilde{\mathbf{Y}}}$ of the simulated vector (Eq. 5) and compare it with the model variance–covariance matrix $\mathbf{C}_{\mathbf{Y}}$, e.g., by calculating a norm of the difference matrix $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$.

2.2 Conditional simulation

Suppose now that one has p preexisting conditioning data $\mathbf{Y}_0 = (Y_0, Y_{-1}, \ldots Y_{1-p})^T$. Equation 1 is modified into

$$\forall i = 1 \ldots n,\ \tilde{Y}_i = \sum_{j=1-p}^{0} \lambda_{j,i} Y_j + \sum_{j=1}^{i-1} \lambda_{j,i} \tilde{Y}_j + \sigma_i U_i \qquad (6)$$

with $\lambda_{j,i}$, the kriging weight assigned to Y_j when predicting Y_i, and σ_i the associated kriging standard deviation. This is equivalent to writing

$$\mathbf{U} = \boldsymbol{\Lambda}_0 \mathbf{Y}_0 + \boldsymbol{\Lambda} \tilde{\mathbf{Y}} \qquad (7)$$

with $\boldsymbol{\Lambda}_0 = \begin{pmatrix} -\frac{\lambda_{1-p,1}}{\sigma_1} & \cdots & -\frac{\lambda_{0,1}}{\sigma_1} \\ \vdots & & \vdots \\ -\frac{\lambda_{1-p,n-1}}{\sigma_{n-1}} & \cdots & -\frac{\lambda_{0,n-1}}{\sigma_{n-1}} \\ -\frac{\lambda_{1-p,n}}{\sigma_n} & \cdots & -\frac{\lambda_{0,n}}{\sigma_n} \end{pmatrix}$ ($n \times p$ matrix)

$\boldsymbol{\Lambda} = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{\lambda_{1,n-1}}{\sigma_{n-1}} & \cdots & \frac{1}{\sigma_{n-1}} & 0 \\ -\frac{\lambda_{1,n}}{\sigma_n} & \cdots & -\frac{\lambda_{n-1,n}}{\sigma_n} & \frac{1}{\sigma_n} \end{pmatrix}$ (lower triangular $n \times n$ matrix).

Because $\mathbf{U}$ is a standard Gaussian random vector and is independent of $\mathbf{Y}_0$, its expectation conditionally to $\mathbf{Y}_0$ is null:

$$\mathbf{0} = E(\mathbf{U}|\mathbf{Y}_0) = \boldsymbol{\Lambda}_0 \mathbf{Y}_0 + \boldsymbol{\Lambda} E\left(\tilde{\mathbf{Y}}|\mathbf{Y}_0\right) \qquad (8)$$

The expectation of the simulated vector $\tilde{\mathbf{Y}}$ conditionally to the data $\mathbf{Y}_0$ is, therefore:

$$E\left(\tilde{\mathbf{Y}}|\mathbf{Y}_0\right) = -\boldsymbol{\Lambda}^{-1} \boldsymbol{\Lambda}_0 \mathbf{Y}_0 \qquad (9)$$

Furthermore, by taking into consideration the independence of the components of $\mathbf{U}$ and using Eqs. 7 and 9, one has:

$$\begin{aligned} \mathbf{I} &= E\left(\mathbf{U}\mathbf{U}^T|\mathbf{Y}_0\right) \\ &= E\Big(\boldsymbol{\Lambda}\mathbf{Y}_0\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T + \boldsymbol{\Lambda}\tilde{\mathbf{Y}}\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T + \boldsymbol{\Lambda}_0\mathbf{Y}_0\tilde{\mathbf{Y}}^T\boldsymbol{\Lambda}^T \\ &\qquad + \boldsymbol{\Lambda}\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\boldsymbol{\Lambda}^T|\mathbf{Y}_0\Big) \\ &= \boldsymbol{\Lambda}_0\mathbf{Y}_0\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T + \boldsymbol{\Lambda}E\left(\tilde{\mathbf{Y}}|\mathbf{Y}_0\right)\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T \\ &\quad + \boldsymbol{\Lambda}_0\mathbf{Y}_0E\left(\tilde{\mathbf{Y}}^T|\mathbf{Y}_0\right)\boldsymbol{\Lambda}^T + \boldsymbol{\Lambda}E\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T|\mathbf{Y}_0\right)\boldsymbol{\Lambda}^T \\ &= -\boldsymbol{\Lambda}_0\mathbf{Y}_0\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T + \boldsymbol{\Lambda}E\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T|\mathbf{Y}_0\right)\boldsymbol{\Lambda}^T \end{aligned} \qquad (10)$$

Consequently, the conditional variance–covariance matrix of $\tilde{\mathbf{Y}}$ is:

$$\begin{aligned} \mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0} &= E\left(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T|\mathbf{Y}_0\right) - E\left(\tilde{\mathbf{Y}}|\mathbf{Y}_0\right)E\left(\tilde{\mathbf{Y}}^T|\mathbf{Y}_0\right) \\ &= \boldsymbol{\Lambda}^{-1}\left(\mathbf{I} + \boldsymbol{\Lambda}_0\mathbf{Y}_0\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T\right)\left(\boldsymbol{\Lambda}^{-1}\right)^T \\ &\quad - \left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_0\,\mathbf{Y}_0\right)\left(\mathbf{Y}_0^T\boldsymbol{\Lambda}_0^T\left(\boldsymbol{\Lambda}^{-1}\right)^T\right) \\ &= \boldsymbol{\Lambda}^{-1}\left(\boldsymbol{\Lambda}^{-1}\right)^T \end{aligned} \qquad (11)$$

Note that, unlike the conditional expectation vector (Eq. 9), this matrix does not depend on the conditioning data values ($\mathbf{Y}_0$) or on the weights assigned to the

conditioning data ($\mathbf{\Lambda}_0$). Its calculation is formally the same as in the non-conditional case (Eq. 5).

Provided that the kriging weights and standard deviations used in Eq. 6 are calculated using *all* the data and previously simulated variables $\{Y_{1-p},...,Y_0, \tilde{Y}_1,... \tilde{Y}_{i-1}\}$ as conditioning data, then the conditional expectation (Eq. 9) and conditional variance–covariance matrix (Eq. 11) coincide with the kriging predictor of $\mathbf{Y}$ from $\mathbf{Y}_0$ and with the variance–covariance matrix of kriging errors, respectively. These results not only hold for simple kriging [3, 8] but also for ordinary, universal, or intrinsic kriging [19, 22]. What is more, the conditional expectation coincides with the kriging predictor of $\mathbf{Y}$ from $\mathbf{Y}_0$ under the weaker condition that the neighborhood used in sequential simulation contains all the components of the conditioning vector $\mathbf{Y}_0$ (Appendix 1).

Based on the previous statements, the following approach is proposed to assess the accuracy of sequential conditional simulation:

1. Determine the model conditional expectation $E(\mathbf{Y}|\mathbf{Y}_0)$ and conditional variance–covariance matrix $\mathbf{C}_{\mathbf{Y}|\mathbf{Y}0}$. This requires predicting $\mathbf{Y}$ from $\mathbf{Y}_0$ by kriging in a unique neighborhood. Formulae giving the kriging error covariances and variances can be found in [19].
2. Define a visiting sequence for simulation; order the components of $\mathbf{Y}$ accordingly.
3. Define a moving neighborhood: number of data, search radii and angular sectors.
4. In accordance with the visiting sequence and moving neighborhood, calculate the simple kriging weights and standard deviations and define matrices $\mathbf{\Lambda}$ and $\mathbf{\Lambda}_0$ (Eq. 7). Data not selected in the moving neighborhood are assigned zero weights.
5. Calculate the conditional expectation vector $\mathrm{E}(\tilde{\mathbf{Y}}|\mathbf{Y}_0)$ (Eq. 9) and the conditional variance–covariance matrix $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0}$ of the simulated vector (Eq. 11). Compare them with the model statistics, by calculating a vector norm of $E(\tilde{\mathbf{Y}}|\mathbf{Y}_0)$–$E(\mathbf{Y} \mid \mathbf{Y}_0)$ and a matrix norm of $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0} - \mathbf{C}_{\mathbf{Y}|\mathbf{Y}_0}$. These norms should be zero if simulation were perfectly accurate.

2.3 Non-conditional cosimulation

Here, it is of interest to cosimulate two jointly Gaussian random vectors $\mathbf{Y} = (Y_1,... Y_n)^T$ and $\mathbf{Z} = (Z_1,... Z_n)^T$ with zero means, variance–covariance matrices $\mathbf{C}_\mathbf{Y}$ and $\mathbf{C}_\mathbf{Z}$ and cross covariance matrix $\mathbf{C}_{\mathbf{YZ}} = E(\mathbf{YZ}^T)$.

Cosimulation can be performed hierarchically: having simulated $\mathbf{Z}$ first and obtained a vector $\tilde{\mathbf{Z}}$, $\mathbf{Y}$ is then simulated conditionally to $\tilde{\mathbf{Z}}$. Similarly to Eq. 7, one has:

$$\mathbf{U} = \mathbf{\Omega}\,\tilde{\mathbf{Z}} + \mathbf{\Lambda}\tilde{\mathbf{Y}} \tag{12}$$

with $\mathbf{U}$ a vector with n independent standard Gaussian components

$$\mathbf{\Omega} = \begin{pmatrix} -\frac{\omega_{1,1}}{\sigma_1} & \cdots & -\frac{\omega_{n,1}}{\sigma_1} \\ \vdots & & \vdots \\ -\frac{\omega_{1,n-1}}{\sigma_{n-1}} & \cdots & -\frac{\omega_{n,n-1}}{\sigma_{n-1}} \\ -\frac{\omega_{1,n}}{\sigma_n} & \cdots & -\frac{\omega_{n,n}}{\sigma_n} \end{pmatrix} \text{ and}$$

$$\mathbf{\Lambda} = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{\lambda_{1,n-1}}{\sigma_{n-1}} & \cdots & \frac{1}{\sigma_{n-1}} & 0 \\ -\frac{\lambda_{1,n}}{\sigma_n} & \cdots & -\frac{\lambda_{n-1,n}}{\sigma_n} & \frac{1}{\sigma_n} \end{pmatrix} (n \times n \text{ matrices})$$

where $\omega_{j,i}$ and $\lambda_{j,i}$ are the simple co kriging weights assigned to Z_j and Y_j when predicting Y_i, and σ_i is the standard deviation of the associated co kriging error.

Based on Eq. 12 and on the independence of $\mathbf{U}$ and $\tilde{\mathbf{Z}}$, one has:

$$\begin{aligned} E\left(\mathbf{\Lambda}\tilde{\mathbf{Y}}\,\tilde{\mathbf{Y}}^T\mathbf{\Lambda}^T\right) &= E\left(\mathbf{U}\mathbf{U}^T - \mathbf{\Omega}\,\tilde{\mathbf{Z}}\mathbf{U}^T - \mathbf{U}\tilde{\mathbf{Z}}^T\mathbf{\Omega}^T \right. \\ &\qquad \left. + \mathbf{\Omega}\,\tilde{\mathbf{Z}}\,\tilde{\mathbf{Z}}^T\mathbf{\Omega}^T\right) \\ &= \mathbf{I} - \mathbf{\Omega}\,\underbrace{E\left(\tilde{\mathbf{Z}}\right)}_{\mathbf{0}}\underbrace{E\left(\mathbf{U}^T\right)}_{\mathbf{0}} \\ &\qquad - \underbrace{E(\mathbf{U})}_{\mathbf{0}}\underbrace{E\left(\tilde{\mathbf{Z}}^T\right)}_{\mathbf{0}}\mathbf{\Omega}^T + \mathbf{\Omega}E\left(\tilde{\mathbf{Z}}\,\tilde{\mathbf{Z}}^T\right)\mathbf{\Omega}^T \end{aligned} \tag{13}$$

and

$$\begin{aligned} E\left(\mathbf{\Lambda}\tilde{\mathbf{Y}}\,\tilde{\mathbf{Z}}^T\right) &= E\left(\mathbf{U}\tilde{\mathbf{Z}}^T - \mathbf{\Omega}\,\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^T\right) \\ &= \underbrace{E\left(\mathbf{U}\right)}_{\mathbf{0}}\underbrace{E\left(\tilde{\mathbf{Z}}^T\right)}_{\mathbf{0}} - \mathbf{\Omega}E\left(\tilde{\mathbf{Z}}\,\tilde{\mathbf{Z}}^T\right) \end{aligned} \tag{14}$$

Let $\mathbf{C}_{\tilde{\mathbf{Y}}} = E\left(\tilde{\mathbf{Y}}\,\tilde{\mathbf{Y}}^T\right)$, $\mathbf{C}_{\tilde{\mathbf{Z}}} = E\left(\tilde{\mathbf{Z}}\,\tilde{\mathbf{Z}}^T\right)$ and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}} = E\left(\tilde{\mathbf{Y}}\,\tilde{\mathbf{Z}}^T\right)$ be the covariance matrices of the simulated vectors. Equations 13 and 14 give the following identities:

$$\mathbf{C}_{\tilde{\mathbf{Y}}} = \mathbf{\Lambda}^{-1}\left(\mathbf{\Lambda}^{-1}\right)^T + \mathbf{\Lambda}^{-1}\mathbf{\Omega}\mathbf{C}_{\tilde{\mathbf{Z}}}\,\mathbf{\Omega}^T\left(\mathbf{\Lambda}^{-1}\right)^T \tag{15}$$

$$\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}} = -\mathbf{\Lambda}^{-1}\,\mathbf{\Omega}\mathbf{C}_{\tilde{\mathbf{Z}}} \tag{16}$$

As in Section 2.1, we assumed that no component of $\mathbf{Y}$ and $\mathbf{Z}$ is a linear combination of other components (the standard deviations of co kriging errors are positive). If this is not the case, redundant components should be removed: the previous Eqs. 12–16 remain valid, although vectors $\mathbf{Y}$ and $\mathbf{Z}$ and matrices $\mathbf{\Omega}$ and $\mathbf{\Lambda}$ may have different sizes, depending on the number of removed components.

If the kriging and co kriging weights and standard deviations used to simulate $\mathbf{Z}$ and $\mathbf{Y}$ are calculated using all the previously simulated variables as conditioning data, then $\mathbf{C}_{\tilde{\mathbf{Y}}}$, $\mathbf{C}_{\tilde{\mathbf{Z}}}$, and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}}$ coincide with the model matrices $\mathbf{C}_{\mathbf{Y}}$, $\mathbf{C}_{\mathbf{Z}}$, and $\mathbf{C}_{\mathbf{YZ}}$. In contrast, if fewer variables are used, the covariance matrices of the simulated vectors are likely to be different from the model matrices; a few particular cases are presented in Appendix 2.

Consequently, the accuracy of sequential cosimulation can be assessed in the following fashion:

1. Calculate the variance–covariance matrix $\mathbf{C}_{\tilde{\mathbf{Z}}}$ of the simulated vector $\tilde{\mathbf{Z}}$ and compare it with the model matrix $\mathbf{C}_{\mathbf{Z}}$ (Section 2.1).
2. Define a visiting sequence for simulating $\mathbf{Y}$. Order the components of $\mathbf{Y}$ accordingly.
3. Define a moving neighborhood: number of data from $\mathbf{Y}$ and $\mathbf{Z}$, search radii and angular sectors.
4. In accordance with the visiting sequence and moving neighborhood, calculate the simple co kriging weights and standard deviations and define matrices $\mathbf{\Omega}$ and $\mathbf{\Lambda}$ (Eq. 12).
5. Calculate $\mathbf{C}_{\tilde{\mathbf{Y}}}$ and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}}$ (Eqs. 15 and 16) and compare these matrices with the model matrices $\mathbf{C}_{\mathbf{Y}}$ and $\mathbf{C}_{\mathbf{YZ}}$. Such a comparison can be done by calculating matrix norms of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$ and of $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}} - \mathbf{C}_{\mathbf{YZ}}$: cosimulation is all the more accurate when these norms are close to zero.

2.4 Conditional cosimulation

Let us finally consider the problem of cosimulating two jointly Gaussian random vectors $\mathbf{Y}$ and $\mathbf{Z}$ conditionally to $\mathbf{Y}_0 = (Y_0, Y_{-1}, \ldots Y_{1-p})^T$ and $\mathbf{Z}_0 = (Z_0, Z_{-1}, \ldots Z_{1-p})^T$. Let us start with the simulation of $\mathbf{Z}$, by writing:

$$\forall i = 1 \ldots n,$$
$$\tilde{Z}_i = \sum_{j=1-p}^{0} \lambda_{j,i}^{Z} Z_j + \sum_{j=1}^{i-1} \lambda_{j,i}^{Z} \tilde{Z}_j + \sum_{j=1-p}^{0} \omega_{j,i}^{Z} Y_j + \sigma_i^Z U_i^Z \tag{17}$$

with $\lambda_{j,i}^{Z}$ and $\omega_{j,i}^{Z}$ the co kriging weights assigned to Z_j and Y_j when predicting Z_i, and σ_i^Z the associated co kriging standard deviation. This is equivalent to writing

$$\mathbf{U}_Z = \mathbf{\Lambda}_{0,Z}\, \mathbf{Z}_0 + \mathbf{\Lambda}_Z \tilde{\mathbf{Z}} + \mathbf{\Omega}_{0,Z}\, \mathbf{Y}_0 \tag{18}$$

with $\mathbf{U}_Z$ a standard Gaussian random vector with independent components,

$$\mathbf{\Lambda}_Z = \begin{pmatrix} -\frac{1}{\sigma_1^Z} & \cdots & 0 \\ \vdots & & \vdots \\ -\frac{\lambda_{1,n-1}^Z}{\sigma_{n-1}^Z} & \cdots & 0 \\ -\frac{\lambda_{1,n}^Z}{\sigma_n^Z} & \cdots & -\frac{1}{\sigma_n^Z} \end{pmatrix} \quad (n \times n \text{ matrix}),$$

$$\mathbf{\Lambda}_{0,Z} = \begin{pmatrix} -\frac{\lambda_{1-p,1}^Z}{\sigma_1^Z} & \cdots & -\frac{\lambda_{0,1}^Z}{\sigma_1^Z} \\ \vdots & & \vdots \\ -\frac{\lambda_{1-p,n-1}^Z}{\sigma_{n-1}^Z} & \cdots & -\frac{\lambda_{0,n-1}^Z}{\sigma_{n-1}^Z} \\ -\frac{\lambda_{1-p,n}^Z}{\sigma_n^Z} & \cdots & -\frac{\lambda_{0,n}^Z}{\sigma_n^Z} \end{pmatrix} \quad \text{and}$$

$$\mathbf{\Omega}_{0,Z} = \begin{pmatrix} -\frac{\omega_{1-p,1}^Z}{\sigma_1^Z} & \cdots & -\frac{\omega_{0,1}^Z}{\sigma_1^Z} \\ \vdots & & \vdots \\ -\frac{\omega_{1-p,n-1}^Z}{\sigma_{n-1}^Z} & \cdots & -\frac{\omega_{0,n-1}^Z}{\sigma_{n-1}^Z} \\ -\frac{\omega_{1-p,n}^Z}{\sigma_n^Z} & \cdots & -\frac{\omega_{0,n}^Z}{\sigma_n^Z} \end{pmatrix} \quad (n \times p \text{ matrices}).$$

Following with the simulation of $\mathbf{Y}$, one has:

$$\forall i = 1 \ldots n,$$
$$\tilde{Y}_i = \sum_{j=1-p}^{0} \lambda_{j,i}^{Y} Y_j + \sum_{j=1}^{i-1} \lambda_{j,i}^{Y} \tilde{Y}_j + \sum_{j=1-p}^{0} \omega_{j,i}^{Y} Z_j + \sum_{j=1}^{n} \omega_{j,i}^{Y} \tilde{Z}_j + \sigma_i^Y U_i^Y \tag{19}$$

with $\lambda_{j,i}^{Y}$ and $\omega_{j,i}^{Y}$ the co kriging weights assigned to Y_j and Z_j when predicting Y_i, and σ_i^Y the associated co kriging standard deviation. Equivalently:

$$\mathbf{U}_Y = \mathbf{\Lambda}_{0,Y} \mathbf{Y}_0 + \mathbf{\Lambda}_Y \tilde{\mathbf{Y}} + \mathbf{\Omega}_{0,Y} \mathbf{Z}_0 + \mathbf{\Omega}_Y \tilde{\mathbf{Z}} \tag{20}$$

with $\mathbf{U}_Y$, a standard Gaussian random vector with independent components,

$$\boldsymbol{\Lambda}_Y = \begin{pmatrix} -\frac{1}{\sigma_1^Y} & \cdots & 0 \\ \vdots & & \vdots \\ -\frac{\lambda_{1,n-1}^Y}{\sigma_{n-1}^Y} & \cdots & 0 \\ -\frac{\lambda_{1,n}^Y}{\sigma_n^Y} & \cdots & -\frac{1}{\sigma_n^Y} \end{pmatrix} \text{ and}$$

$$\boldsymbol{\Omega}_Y = \begin{pmatrix} -\frac{\omega_{1,1}^Y}{\sigma_1^Y} & \cdots & -\frac{\omega_{n,1}^Y}{\sigma_1^Y} \\ \vdots & & \vdots \\ -\frac{\omega_{1,n-1}^Y}{\sigma_{n-1}^Y} & \cdots & -\frac{\omega_{n,n-1}^Y}{\sigma_{n-1}^Y} \\ -\frac{\omega_{1,n}^Y}{\sigma_n^Y} & \cdots & -\frac{\omega_{n,n}^Y}{\sigma_n^Y} \end{pmatrix} (n \times n \text{ matrices}),$$

$$\boldsymbol{\Lambda}_{0,Y} = \begin{pmatrix} -\frac{\lambda_{1-p,1}^Y}{\sigma_1^Y} & \cdots & -\frac{\lambda_{0,1}^Y}{\sigma_1^Y} \\ \vdots & & \vdots \\ -\frac{\lambda_{1-p,n-1}^Y}{\sigma_{n-1}^Y} & \cdots & -\frac{\lambda_{0,n-1}^Y}{\sigma_{n-1}^Y} \\ -\frac{\lambda_{1-p,n}^Y}{\sigma_n^Y} & \cdots & -\frac{\lambda_{0,n}^Y}{\sigma_n^Y} \end{pmatrix} \text{ and}$$

$$\boldsymbol{\Omega}_{0,Y} = \begin{pmatrix} -\frac{\omega_{1-p,1}^Y}{\sigma_1^Y} & \cdots & -\frac{\omega_{0,1}^Y}{\sigma_1^Y} \\ \vdots & & \vdots \\ -\frac{\omega_{1-p,n-1}^Y}{\sigma_{n-1}^Y} & \cdots & -\frac{\omega_{0,n-1}^Y}{\sigma_{n-1}^Y} \\ -\frac{\omega_{1-p,n}^Y}{\sigma_n^Y} & \cdots & -\frac{\omega_{0,n}^Y}{\sigma_n^Y} \end{pmatrix} (n \times p \text{ matrices}).$$

By reasoning as in Sections 2.2 and 2.3, the following identities relating the conditional covariance matrices of the simulated vectors are found:

$$\boldsymbol{\Lambda}_Z \mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} \boldsymbol{\Lambda}_Z^T = \mathbf{I} \tag{21}$$

$$\boldsymbol{\Lambda}_Y \mathbf{C}_{\tilde{\mathbf{Y}}|(\mathbf{Y}_0,\mathbf{Z}_0)} \boldsymbol{\Lambda}_Y^T = \mathbf{I} + \boldsymbol{\Omega}_Y \mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} \boldsymbol{\Omega}_Y^T \tag{22}$$

$$\boldsymbol{\Lambda}_Y \mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} = -\boldsymbol{\Omega}_Y \mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} \tag{23}$$

Provided that the co kriging weights and standard deviations are calculated using all the previously simulated variables as conditioning data, then $\mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, $\mathbf{C}_{\tilde{\mathbf{Y}}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)}$ coincide with the covariance matrices of the co kriging errors of $(\mathbf{Y},\mathbf{Z})$ from $(\mathbf{Y}_0, \mathbf{Z}_0)$. This result holds for simple, ordinary, universal, or intrinsic co kriging.

To assess the accuracy of sequential conditional cosimulation, the following approach is proposed:

1. Determine the model covariance matrices $\mathbf{C}_{\mathbf{Y}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, $\mathbf{C}_{\mathbf{Z}|(\mathbf{Y}_0,\mathbf{Z}_0)}$ and $\mathbf{C}_{\mathbf{YZ}|(\mathbf{Y}_0,\mathbf{Z}_0)}$. This requires co kriging $(\mathbf{Y},\mathbf{Z})$ from $(\mathbf{Y}_0,\mathbf{Z}_0)$ (co kriging in a unique neighborhood).
2. Define visiting sequences for cosimulating vectors $\mathbf{Z}$ and $\mathbf{Y}$.
3. Define moving neighborhoods for cosimulating vectors $\mathbf{Z}$ and $\mathbf{Y}$: number of data, search radii and angular sectors.
4. According to the visiting sequences and moving neighborhoods, calculate the co kriging weights and standard deviations and define matrices $\boldsymbol{\Lambda}_Z$, $\boldsymbol{\Lambda}_Y$, and $\boldsymbol{\Omega}_Y$ (Eqs. 18–20). Data not selected in the moving neighborhood are assigned zero weights.
5. Calculate $\mathbf{C}_{\tilde{\mathbf{Y}}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, $\mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)}$ and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)}$ (Eqs. 21–23).
6. Compare these matrices with the model matrices $\mathbf{C}_{\mathbf{Y}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, $\mathbf{C}_{\mathbf{Z}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, and $\mathbf{C}_{\mathbf{YZ}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, e.g., by calculating matrix norms of $\mathbf{C}_{\tilde{\mathbf{Y}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{Y}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, $\mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{Z}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{YZ}|(\mathbf{Y}_0,\mathbf{Z}_0)}$.

Note that the weights assigned to the conditioning data (matrices $\boldsymbol{\Lambda}_{0,Z}$, $\boldsymbol{\Lambda}_{0,Y}$, $\boldsymbol{\Omega}_{0,Z}$, and $\boldsymbol{\Omega}_{0,Y}$) and the conditioning data values $(\mathbf{Y}_0,\mathbf{Z}_0)$ have no influence on the conditional covariance matrices. However, they have an influence on the conditional means. Based on Eqs. 18 and 20, these are found to be:

$$E\left(\tilde{\mathbf{Z}}|\mathbf{Y}_0,\ \mathbf{Z}_0\right) = -\boldsymbol{\Lambda}_Z^{-1}\left(\boldsymbol{\Lambda}_{0,Z}\mathbf{Z}_0 + \boldsymbol{\Omega}_{0,Z}\mathbf{Y}_0\right) \tag{24}$$

$$E\left(\tilde{\mathbf{Y}}|\mathbf{Y}_0,\ \mathbf{Z}_0\right) = -\boldsymbol{\Lambda}_Y^{-1}\left(\boldsymbol{\Lambda}_{0,Y}\mathbf{Y}_0 + \boldsymbol{\Omega}_{0,Y}\mathbf{Z}_0 + \boldsymbol{\Omega}_Y E\left(\tilde{\mathbf{Z}}|\mathbf{Y}_0,\ \mathbf{Z}_0\right)\right) \tag{25}$$

As in the univariate case (Appendix 1), it can be shown that such conditional expectations coincide with the theoretically expected ones (i.e., the co kriging predictions in a unique neighborhood), provided that the moving neighborhoods used in the sequential simulation of $\mathbf{Z}$ and $\mathbf{Y}$ contain all the conditioning data ($\mathbf{Z}_0$ and $\mathbf{Y}_0$).

3 Numerical experiments

In this section, numerical experiments will be realized in order to assess the accuracy of sequential Gaussian simulation under given implementation settings (prior covariance model, visiting sequence, moving neighborhood, and type of kriging or co kriging used to determine the conditional distributions). Some alternatives will also be proposed in order to improve the reproduction of the model variance–covariance matrices.

3.1 Visiting sequences and neighborhood parameters

When using a moving neighborhood, the kriging or co kriging weights and variances used to determine the conditional distributions depend on the neighborhood parameters (search radii, use of angular sectors, number of data per sector...) and on the sequence according to which the spatial locations targeted for simulation are visited.

In the following, these locations will be the nodes of a regular 2D grid with 65 × 65 nodes and mesh 1 × 1. Four particular sequences will be considered:

1. A regular sequence in which the nodes are visited row-wise.
2. A regular sequence in which the nodes are visited from the center of the grid toward the edges, following an arithmetic spiral.
3. An irregular sequence, in which the ordering of the grid nodes is obtained by a random permutation of the 4,225 grid nodes.
4. An irregular sequence in which the corners of the grid are visited first; at each subsequent step, one visits the midpoints of the already visited points, until all the grid nodes are visited (Fig. 1). Such a "midpoint displacement" sequence is a particular case of *multiple-grid* sequence, in which one defines nested grids and starts visiting the coarsest grid then follows with the intermediate grids and ends with the finest grid [24, 40].

For brevity, these sequences will be named *row-wise*, *spiral*, *random*, and *midpoint*, respectively. Concerning the third sequence, any other random permutation of the grid nodes could have been chosen, which may have an impact on the reproduction of the model variance–covariance matrix. For this reason, 20 other random sequences have also been tested, but the results are not significantly different from those of the particular random sequence presented hereunder.

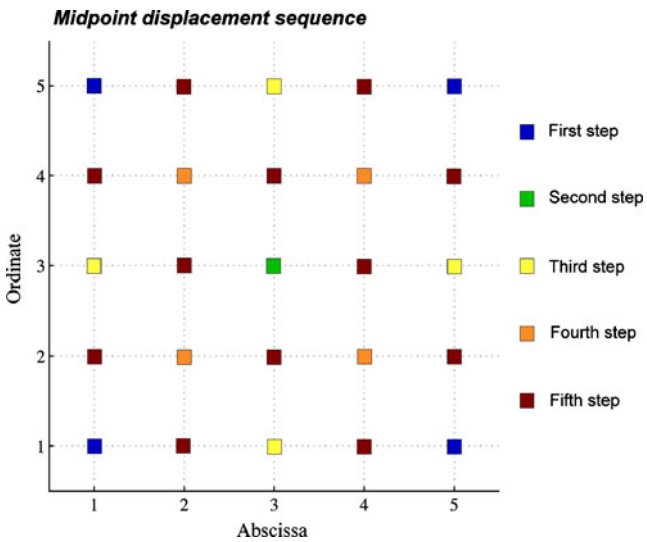

Fig. 1 Midpoint displacement sequence for a grid with 5 × 5 nodes

As for the search of conditioning data, we will consider four moving neighborhoods, respectively containing up to 10, 20, 50, and 100 previously simulated variables (the ones located at the grid nodes closest to the node targeted for simulation, without considering angular sectors or radius restrictions), in addition to all the original conditioning data in case of performing conditional simulation.

3.2 Criterion for assessing the accuracy of simulation

The accuracy of sequential simulation will be assessed by calculating the Frobenius norm of the difference between the model covariance matrix ($\mathbf{C}_Y$) and the covariance matrix of the simulated vector $\left(\mathbf{C}_{\tilde{\mathbf{Y}}}\right)$, standardized by the Frobenius norm of the model covariance matrix:

$$\eta = \frac{||\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}||}{||\mathbf{C}_{\mathbf{Y}}||} \tag{26}$$

where $||\mathbf{A}||$ is the square root of the sum of the squared entries of $\mathbf{A}$ (Frobenius norm).

In the sequel, η will be referred to as the *standardized Frobenius norm* of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$. It measures the accuracy in the reproduction of the model covariance matrix, in the form of a root-mean-square deviation relative to the root mean square of the entries of the model covariance matrix, i.e., in the form of a relative error. For conditional simulation, we will consider $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0}$ and $\mathbf{C}_{\mathbf{Y}|\mathbf{Y}_0}$ instead of $\mathbf{C}_{\tilde{\mathbf{Y}}}$ and $\mathbf{C}_{\mathbf{Y}}$. In multivariate cases, we will also consider cross covariance matrices in addition to the direct covariance matrices.

A zero value for η indicates a perfectly accurate simulation, for which the covariance matrix of the simulated vector $\mathbf{C}_{\tilde{\mathbf{Y}}}$ exactly matches the desired model matrix $\mathbf{C}_{\mathbf{Y}}$. Any non-zero value of η indicates an inaccurate reproduction of the model covariance matrix. This situation is likely to be met in practice, because of a deficient screening effect when using a moving neighborhood to determine the successive kriging weights and standard deviations. To decide whether or not the model matrix is well reproduced, the user can define a threshold value for η, e.g., $\eta_{\max} = 0.05$: the mismatch between $\mathbf{C}_{\tilde{\mathbf{Y}}}$ and $\mathbf{C}_{\mathbf{Y}}$ is deemed acceptable if the norm of the difference is smaller than 5% of the norm of $\mathbf{C}_{\mathbf{Y}}$.

3.3 Non-conditional simulation

In this sub-section, sequential Gaussian simulation is performed without any pre existing conditioning data. Eight stationary isotropic prior covariance models, each with sill 1, are considered:

1) Spherical with range 10
2) Spherical with range 40
3) Spherical with range 10 and nugget 30% of the total sill
4) Spherical with range 40 and nugget 30% of the total sill
5) Gaussian with practical range 10 and nugget 1% of the total sill
6) Gaussian with practical range 40 and nugget 1% of the total sill
7) Gaussian with practical range 10 and nugget 30% of the total sill
8) Gaussian with practical range 40 and nugget 30% of the total sill.

In each case, the standardized Frobenius norm of the difference matrix $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$ (Eq. 26) is indicated in Tables 1 and 2. Overall, it is seen that the accuracy of the simulation depends on the covariance model to be reproduced, on the moving neighborhood and on the visiting sequence. Detailed comments are provided in the next sub-sections.

Note that the pure Gaussian model (without nugget effect) has not been considered in this exercise, insofar as it leads to very poor results. This is explained by the sensitivity of the kriging weights obtained when using a moving neighborhood and by the numerical instabilities caused by this model when some data locations are close to each other [28, 34]. Other simulation algorithms that do not suffer from these problems, such as the continuous spectral or turning bands, should be preferred in this case [23, 28].

3.3.1 Impact of the visiting sequence

In almost all the cases, the midpoint displacement sequence yields the best reproduction of the target variance–covariance matrix, while the regular sequences (row-wise and spiral) have the poorest performances. These results corroborate the findings by various other authors who advocate against the use of regular visiting sequences [24, 40].

3.3.2 Impact of the number of conditioning variables

In general, the target variance–covariance matrix is better reproduced if more variables are included in

Table 1 Standardized Frobenius norms of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$ for non-conditional simulation with spherical covariance model

Range	Nugget	Separate simulation of nugget effect?	Sequence	Moving neighborhood			
				10 nodes	20 nodes	50 nodes	100 nodes
10	0%	No	Row-wise	0.5098	0.3138	0.1783	0.1810
10	0%	No	Spiral	0.5144	0.2833	0.1706	0.1837
10	0%	No	Random	0.2357	0.1109	0.0467	0.0266
10	0%	No	Midpoint	0.2096	0.0599	0.0286	0.0175
10	30%	No	Row-wise	0.5138	0.5506	0.1820	0.1689
10	30%	No	Spiral	0.6073	0.5351	0.1706	0.1601
10	30%	No	Random	0.2318	0.2083	0.0734	0.0396
10	30%	No	Midpoint	0.1900	0.1616	0.0598	0.0343
10	30%	Yes	Row-wise	*0.4998*	*0.3077*	*0.1748*	0.1775
10	30%	Yes	Spiral	*0.5044*	*0.2778*	*0.1672*	0.1801
10	30%	Yes	Random	*0.2311*	*0.1087*	*0.0458*	*0.0261*
10	30%	Yes	Midpoint	0.2055	*0.0587*	*0.0281*	*0.0171*
40	0%	No	Row-wise	0.5225	0.4287	0.3380	0.2704
40	0%	No	Spiral	0.7312	0.5815	0.4262	0.2882
40	0%	No	Random	0.1761	0.0836	0.0208	0.0097
40	0%	No	Midpoint	0.1400	0.0342	0.0104	0.0046
40	30%	No	Row-wise	0.6424	0.4836	0.4943	0.3658
40	30%	No	Spiral	0.4736	0.6563	0.6824	0.4188
40	30%	No	Random	0.2221	0.1532	0.0962	0.0403
40	30%	No	Midpoint	0.2385	0.1334	0.0786	0.0342
40	30%	Yes	Row-wise	*0.5217*	*0.4280*	*0.3374*	*0.2700*
40	30%	Yes	Spiral	*0.7301*	*0.5807*	*0.4256*	*0.2878*
40	30%	Yes	Random	*0.1772*	*0.0712*	*0.0203*	*0.0095*
40	30%	Yes	Midpoint	*0.1398*	*0.0341*	*0.0104*	*0.0046*

Smallest norms are indicated in blue, largest norms in red. Italics indicate improvement of separate nugget effect simulation over direct simulation

Table 2 Standardized Frobenius norms of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$ for non-conditional simulation with Gaussian covariance model

Range	Nugget	Separate simulation of nugget effect?	Sequence	Moving neighborhood			
				10 nodes	20 nodes	50 nodes	100 nodes
10	1%	No	Row-wise	1.1373	3.2534	0.4956	0.2318
10	1%	No	Spiral	1.0768	1.1079	0.4017	0.2085
10	1%	No	Random	0.1610	0.0766	0.0253	0.0083
10	1%	No	Midpoint	0.1264	0.0437	0.0185	0.0057
10	30%	No	Row-wise	1.1539	1.8673	0.3741	0.1651
10	30%	No	Spiral	1.4059	1.2252	0.3076	0.1472
10	30%	No	Random	0.2887	0.2436	0.0650	0.0358
10	30%	No	Midpoint	0.2163	0.1958	0.0595	0.0316
10	30%	Yes	Row-wise	1.9290	2.6586	0.4654	0.2210
10	30%	Yes	Spiral	*1.2548*	*1.0119*	0.3599	0.2003
10	30%	Yes	Random	*0.1692*	*0.0769*	*0.0271*	*0.0088*
10	30%	Yes	Midpoint	*0.1331*	*0.0437*	*0.0195*	*0.0060*
40	1%	No	Row-wise	0.9870	2.2121	1.1857	0.7603
40	1%	No	Spiral	1.8846	1.6213	0.7173	0.5809
40	1%	No	Random	0.1179	0.0547	0.0201	0.0067
40	1%	No	Midpoint	0.0861	0.0433	0.0193	0.0065
40	30%	No	Row-wise	0.6757	0.5333	0.9348	1.2679
40	30%	No	Spiral	0.4984	0.9344	1.3695	0.7919
40	30%	No	Random	0.2277	0.1700	0.1457	0.0952
40	30%	No	Midpoint	0.2597	0.1554	0.1365	0.0937
40	30%	Yes	Row-wise	0.9434	1.7650	1.4765	*0.6809*
40	30%	Yes	Spiral	1.8340	1.7101	*0.6175*	*0.6282*
40	30%	Yes	Random	*0.1270*	*0.0629*	*0.0260*	*0.0089*
40	30%	Yes	Midpoint	*0.0923*	*0.0512*	*0.0246*	*0.0087*

Smallest norms are indicated in blue, largest norms in red. Italics indicate improvement of separate nugget effect simulation over direct simulation

the moving neighborhood. Significant improvements can be observed when passing from 20 to 50 variables and, to a lesser extent, from 50 to 100 variables. These numbers are greater than those habitually considered for kriging [8, 15, 25], suggesting that the design of the moving neighborhood for sequential simulation is more demanding than for kriging. This can be explained because the errors due to deficient screening effect are likely to propagate in sequential simulation, insofar as the simulated variables are used as conditioning data for simulating subsequent variables.

There are, however, a few exceptions, especially with the Gaussian covariance model, for which increasing the number of variables in the moving neighborhood deteriorates the reproduction of the model variance–covariance matrix. This rather counterintuitive result can be explained by recalling that the variance–covariance matrix of the simulated vector depends on the successive kriging weights and on the error variances and coincides with the model matrix if kriging is performed in a unique neighborhood (Section 2.1). When using a moving neighborhood, increasing the number of conditioning variables makes the error variances get closer to the ones that would be obtained in a unique neighborhood, but this may not be the case for the kriging weights due to edge effects, string effects, or to numerical instabilities for the Gaussian covariance model [13, 14, 34]. Accordingly, the norm of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$ may not necessarily be a monotonic function of the number of conditioning variables selected in the moving neighborhood, although it tends to zero as this number becomes very large.

3.3.3 Covariance structure of the simulated vector

Because the prior covariance model is stationary, the target variance–covariance matrix $\mathbf{C}_{\mathbf{Y}}$ has a block Toeplitz structure. However, this property does no longer hold with the variance–covariance matrix $\mathbf{C}_{\tilde{\mathbf{Y}}}$ of the simulated vector. In other words, the covariance (equivalently, the variogram) between the components of $\tilde{\mathbf{Y}}$ at two grid nodes depends on the positions of these two nodes within the grid, and not only on their separation vector. As a result, for a given separation vector, one obtains a set of simulated covariances (variograms) rather than a single covariance (variogram) value, which can be represented by a cloud of points or by a box-plot. An illustration is given in Fig. 2 for the spherical model with range 10 and no nugget effect, considering a neighborhood with 20 previously simulated variables. It is seen that the model is well reproduced at short-scale, but biases appear at larger

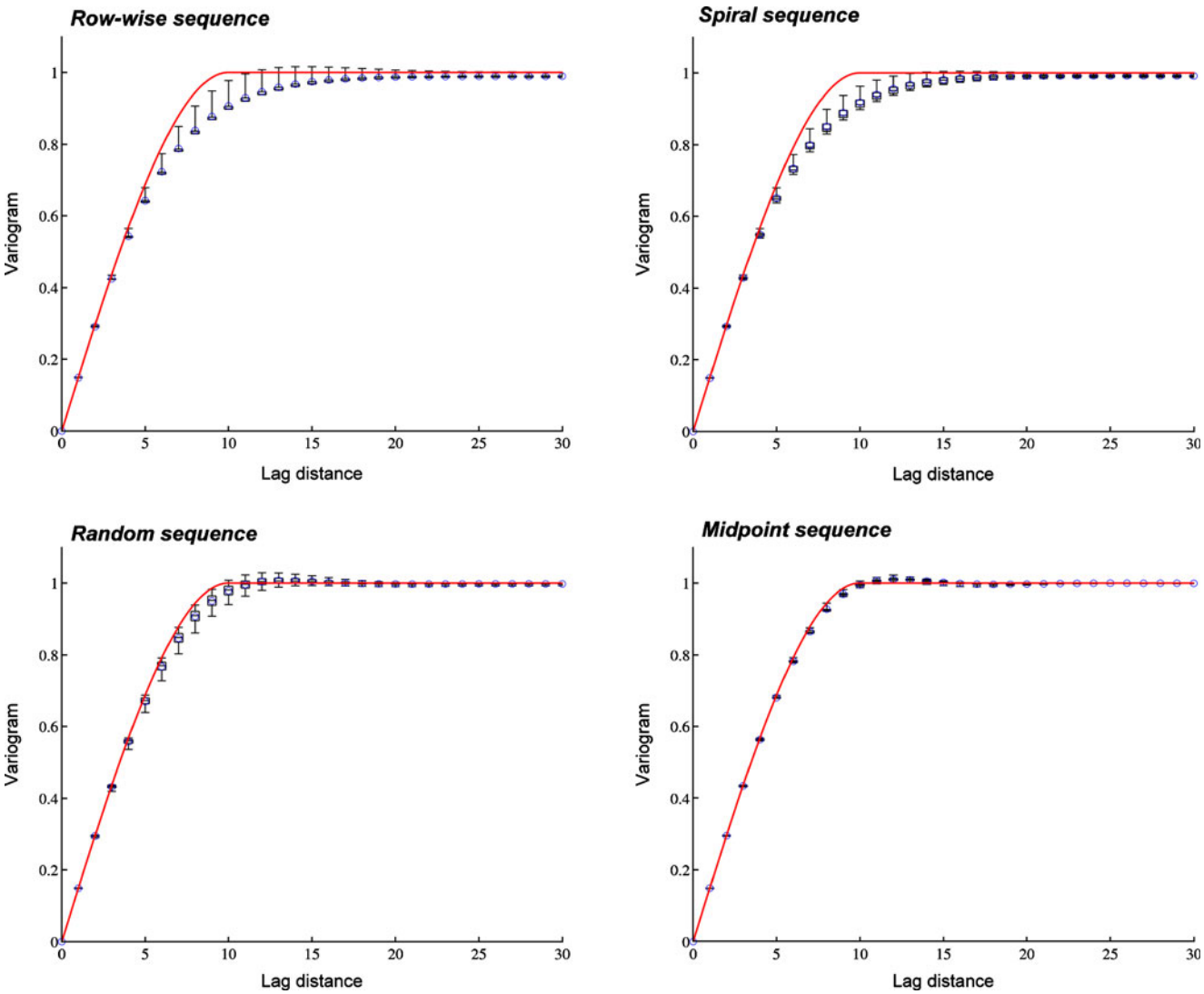

Fig. 2 Prior variogram model (*solid line*) and distributions of the variograms between pairs of simulated components (*circles* indicate mean values, *box plots* indicate quantiles at 2.5%, 25%, 50%, 75%, and 97.5%). Implementation with a spherical model with range 10 and no nugget, and a neighborhood with 20 previously simulated variables

scales (exaggerated range for the row-wise, spiral and random sequences; apparition of a slight hole effect for the random and midpoint sequences).

These biases are also perceptible on Fig. 3, which maps the covariances between the variables at two particular grid nodes (nos. 2113 and 3089) and the variables at surrounding nodes: the midpoint displacement sequence yields the covariance maps closest to the model, in terms of correlation range and isotropic variations (circular patterns).

Two additional comments deserve to be mentioned at this stage.

1) Reproducing the complete variance–covariance matrix is much more demanding than reproducing only the covariance or the variogram on average over all the pairs of grid nodes. Traditional approaches for validating the sequential simulation algorithm often focus on the latter point, by comparing the sample covariances or variograms of a few realizations with the underlying theoretical model. In contrast, the proposed approach allows inspecting the covariance or the variogram of any particular pair of simulated variables.
2) Despite a widespread belief, the use of a moving neighborhood does not yield random fields with a covariance structure identical to the desired model for distances less than the neighborhood radius: some deviations between the actual and desired covariances or variograms are likely to be observed,

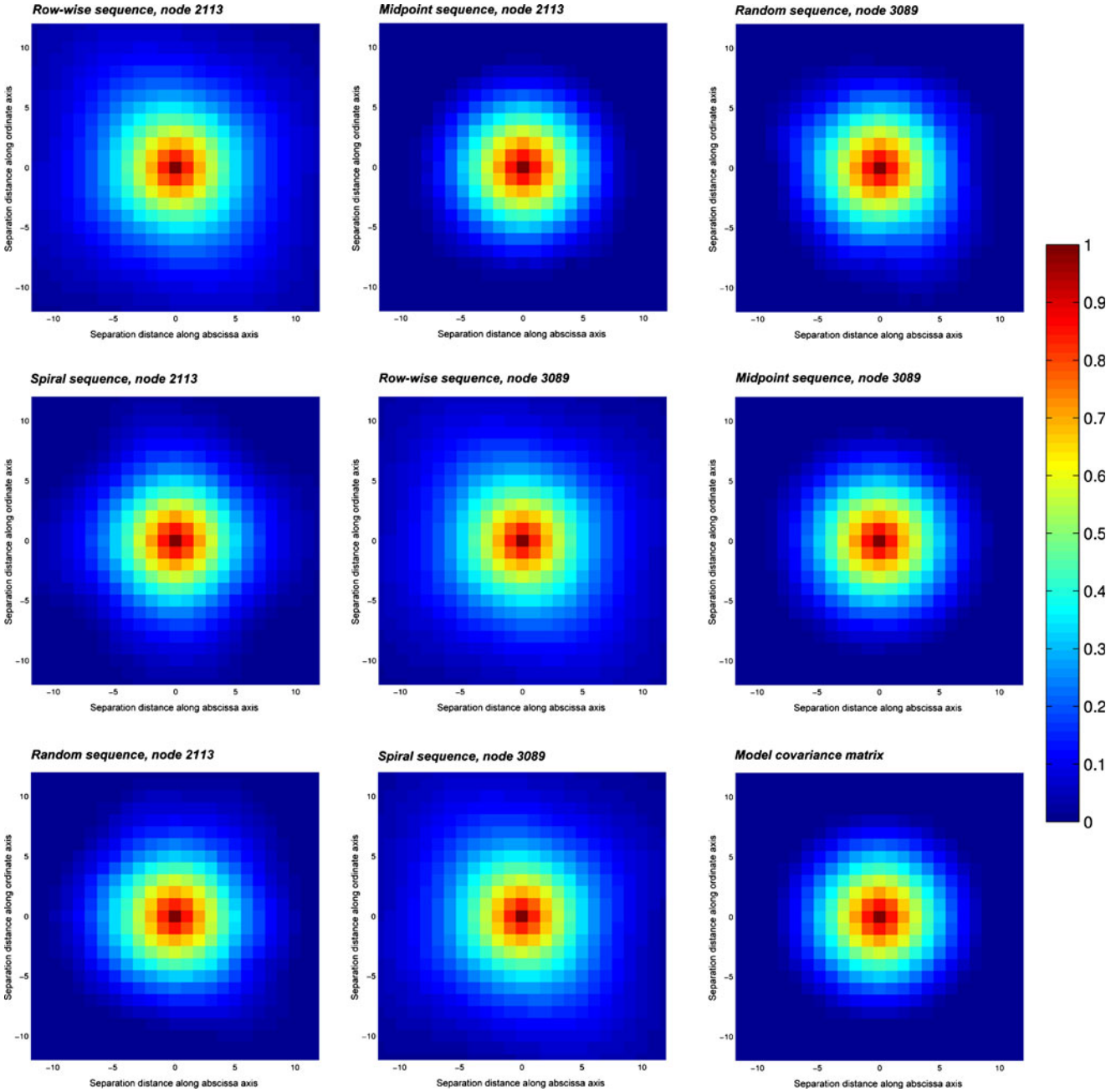

Fig. 3 Covariance maps between two given vector components and their surrounding components. Implementation with a spherical model with range 10 and no nugget, and a neighborhood with 20 previously simulated variables

even at very small distances. This is explained because a perturbation in the ith row of $\mathbf{\Lambda}$ propagates into the ith to nth rows of $\mathbf{\Lambda}^{-1}$ and of $\mathbf{C}_{\tilde{\mathbf{Y}}}$ (Eq. 5). To illustrate these statements, let us go back to the example of Fig. 2, in which the neighborhood contains 20 variables (hence, all the variables within a radius of 2 or more units): the variogram of the simulated random variables calculated along the abscissa axis at lags 1 and 2 fluctuate around the model variogram values (0.1495 and 0.2960), with small biases and with slightly higher fluctuations in the case of the random sequence (Tables 3 and 4).

3.3.4 An alternative: separate simulation of the nugget component

To improve the reproduction of the covariance models with nugget effect, an option is to simulate (via the

Table 3 Statistics on the variogram between simulated variables at grid nodes separated by 1 unit

Sequence	Row-wise	Spiral	Random	Midpoint
Minimum	0.1483	0.1483	0.1465	0.1491
Maximum	0.1496	0.1500	0.1512	0.1497
Mean	0.1484	0.1489	0.1493	0.1495

Desired value is 0.1495

Table 4 Statistics on the variogram between simulated variables at grid nodes separated by 2 units

Sequence	Row-wise	Spiral	Random	Midpoint
Minimum	0.2903	0.2903	0.2812	0.2942
Maximum	0.2960	0.2972	0.3008	0.2967
Mean	0.2910	0.2931	0.2947	0.2955

Desired value is 0.2960

sequential algorithm) a random vector without such a nugget effect (or with a very small nugget effect for smooth covariance models such as the Gaussian), then to add a Gaussian random vector with uncorrelated components and with variance equal to the nugget effect variance. This way, only the structured part of the covariance is affected by inaccuracies due to the implementation parameters (moving neighborhood and visiting sequence), while the nugget part is perfectly reproduced.

The standardized Frobenius norms of the differences between the target and actually simulated covariance matrices are indicated in Tables 1 and 2 for the covariance models with 30% nugget effect. For the random and midpoint displacement visiting sequences, one observes a significant improvement in the reproduction of the target covariance matrix. Also, the smallest norms are always associated with the midpoint displacement sequence.

3.4 Conditional simulation

To limit the number of experiments, we will now focus on two specific covariance models (spherical of range 10, without nugget and with 30% relative nugget effect), one visiting sequence (midpoint displacement), and two kriging options (simple and ordinary). Two configurations of conditioning data are considered, both with 36 data: a regular grid and a configuration with aligned data points, emulating drill holes (Fig. 4).

3.4.1 Reproduction of the conditional moments

In each case, sequential simulation perfectly reproduces the conditional expectation, since the moving neighborhoods contain all the original conditioning data (Appendix 1). The accuracy of the simulation is assessed by examining the conditional variance–covariance matrix and by calculating the standardized Frobenius norm of $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0} - \mathbf{C}_{\mathbf{Y}|\mathbf{Y}_0}$.

Using ordinary kriging leads to a poorer reproduction of the conditional covariance matrix than using simple kriging (Table 5). This is explained because, in ordinary kriging with a unique neighborhood, data located far from the target location may receive non-negligible weights, since they implicitly contribute to the estimation of the unknown mean [34, 35]: the moving neighborhood, therefore, discards relevant data and significantly deteriorates the results with respect to a unique neighborhood implementation. Moreover, these results turn out to be the poorest (1) in the presence of a nugget effect and (2) with the second configuration of conditioning data: in both cases, data closest to the target kriging location may not screen out the influence of farther data due to the nugget effect and/or a string effect [8, 13, 14, 34], so that there is a significant loss of accuracy when using a moving neighborhood.

3.4.2 An alternative: two-step simulation

To improve the reproduction of the conditional covariance matrix when using ordinary kriging, a two-step approach can be used [22]:

1) Non-conditional sequential simulation at the data locations and target grid nodes, using simple kriging in order to minimize the loss of accuracy caused by deficient screening effect and string effect.
2) Conditioning to original data by ordinary kriging:
   a) Calculate the residuals (data values minus non-conditional simulation) at the data locations.
   b) Estimate the residuals at the target grid nodes by ordinary kriging.
   c) Add the estimated residuals to the non-conditional simulation.

In the cases under consideration, non-conditional simulation is first made at the 36 conditioning data locations (unique neighborhood), then at the target grid nodes (moving neighborhood, as in the previous experiments), while conditioning by ordinary kriging is performed in a unique neighborhood due to the small amount of conditioning data (36).

The conditional covariance matrix $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0}$ of the simulated vector can be calculated from the covariance

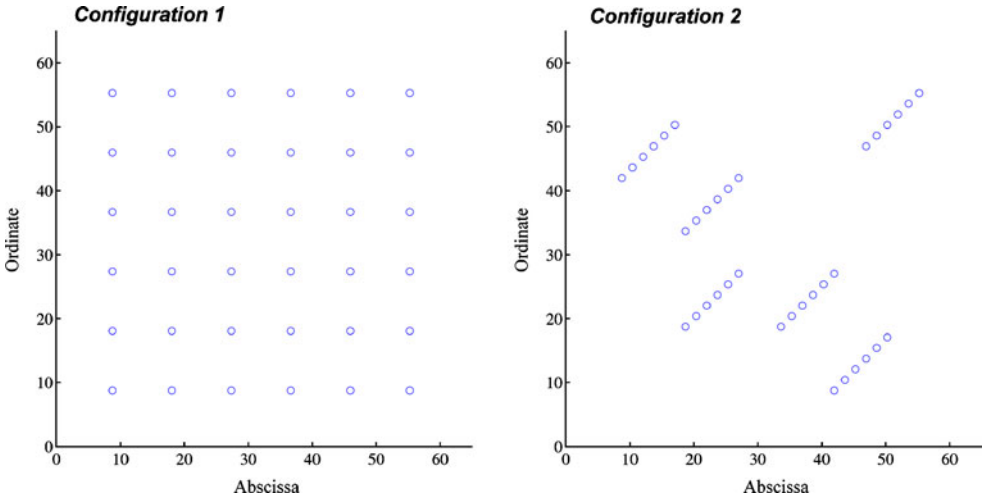

Fig. 4 Locations of conditioning data

matrix of the non-conditional simulation at the data locations and target grid nodes (Section 2.1) and the ordinary kriging weights used in the conditioning step. On the other hand, the model covariance matrix $\mathbf{C}_{\mathbf{Y}|\mathbf{Y}_0}$ is the variance–covariance matrix of kriging errors (ordinary kriging of $\mathbf{Y}$ from $\mathbf{Y}_0$).

It is observed (Table 5) that the standardized Frobenius norm of $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0} - \mathbf{C}_{\mathbf{Y}|\mathbf{Y}_0}$ is much smaller than that obtained with direct sequential simulation using ordinary kriging, even smaller than that obtained with direct sequential simulation using simple kriging. The two-step approach therefore allows performing conditional simulation without a significant loss of accuracy when using a moving neighborhood and ordinary kriging to determine the conditional moments.

3.5 Multivariate simulation

We finally consider the cosimulation of two jointly Gaussian random vectors $\mathbf{Y}$ and $\mathbf{Z}$ under an intrinsic correlation model (proportional covariance model) [8, 41]:

$$\mathbf{C}_{\mathbf{Y}} = \mathbf{C}_{\mathbf{Z}} \text{ and } \mathbf{C}_{\mathbf{YZ}} = \rho\, \mathbf{C}_{\mathbf{Y}} \tag{27}$$

with ρ the correlation coefficient between components of $\mathbf{Y}$ and $\mathbf{Z}$. The intrinsic correlation model is a particular case of reverse Markov model, for which full co kriging reduces to multi-collocated co kriging (Appendix 2) [36, 37].

The experiments will be run with the following implementation parameters:

- Covariance type: spherical with range 10, without nugget and with 30% nugget
- Correlation coefficient between components of $\mathbf{Y}$ and $\mathbf{Z}$: $\rho = 0.7$
- Cosimulation type: non-conditional and conditional (dataset configuration 2, as shown in Fig. 4)
- First vector ($\mathbf{Z}$) perfectly simulated (unique neighborhood)

Table 5 Standardized Frobenius norms of $\mathbf{C}_{\tilde{\mathbf{Y}}|\mathbf{Y}_0} - \mathbf{C}_{\mathbf{Y}|\mathbf{Y}_0}$ for conditional simulation with spherical covariance model of range 10

Nugget	Conditioning dataset	Conditioning kriging	Two-step simulation?	Moving neighboorhhood			
				10 nodes	20 nodes	50 nodes	100 nodes
0%	1	Simple	No	0.1865	0.0740	0.0365	0.0220
0%	1	Ordinary	No	0.2204	0.1295	0.0726	0.0360
0%	1	Mixed	Yes	0.1805	0.0716	0.0353	0.0213
30%	1	Simple	No	0.2024	0.1606	0.0678	0.0399
30%	1	Ordinary	No	0.2899	0.2522	0.1576	0.0997
30%	1	Mixed	Yes	0.1879	0.1494	0.0629	0.0371
0%	2	Simple	No	0.1960	0.0628	0.0298	0.0183
0%	2	Ordinary	No	0.4321	0.3079	0.1112	0.0440
0%	2	Mixed	Yes	0.1491	0.0478	0.0226	0.0139
30%	2	Simple	No	0.1865	0.1554	0.0614	0.0356
30%	2	Ordinary	No	0.5093	0.4086	0.2191	0.1226
30%	2	Mixed	Yes	0.1317	0.1097	0.0433	0.0251

Smallest norms are indicated in blue, largest norms in red. Two-step simulation uses non-conditional simulation with simple kriging, followed by conditioning to data by ordinary kriging

Table 6 Standardized Frobenius norms of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$, $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}} - \mathbf{C}_{\mathbf{YZ}}$, $\mathbf{C}_{\tilde{\mathbf{Y}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{Y}|(\mathbf{Y}_0,\mathbf{Z}_0)}$ and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{YZ}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, for cosimulation under an intrinsic correlation model (spherical covariance model of range 10)

Smallest norms are indicated in blue, largest norms in red. Italics indicate improvement over univariate simulation

Nugget	Conditioning dataset	Cokriging type	Covariance matrix	Moving neighborhood			
				10 nodes	20 nodes	50 nodes	100 nodes
0%	None	Strictly collocated	$\mathbf{C}_{\mathbf{Y}}$	0.2867	0.1914	0.1679	0.1647
			$\mathbf{C}_{\mathbf{YZ}}$	0.4448	0.4031	0.3910	0.3894
0%	None	Multi-collocated	$\mathbf{C}_{\mathbf{Y}}$	*0.1069*	*0.0305*	*0.0146*	*0.0089*
			$\mathbf{C}_{\mathbf{YZ}}$	0	0	0	0
30%	None	Strictly collocated	$\mathbf{C}_{\mathbf{Y}}$	0.3419	0.3938	0.3534	0.3390
			$\mathbf{C}_{\mathbf{YZ}}$	0.5580	0.5889	0.5700	0.5631
30%	None	Multi-collocated	$\mathbf{C}_{\mathbf{Y}}$	*0.1314*	*0.1219*	*0.0925*	*0.0888*
			$\mathbf{C}_{\mathbf{YZ}}$	0	0	0	0
0%	2	Strictly collocated	$\mathbf{C}_{\mathbf{Y}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	0.2577	0.1748	0.1532	0.1499
			$\mathbf{C}_{\mathbf{YZ}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	0.4185	0.3830	0.3724	0.3709
0%	2	Multi-collocated	$\mathbf{C}_{\mathbf{Y}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	*0.1000*	*0.0320*	*0.0152*	*0.0093*
			$\mathbf{C}_{\mathbf{YZ}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	0	0	0	0
30%	2	Strictly collocated	$\mathbf{C}_{\mathbf{Y}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	0.3282	0.3777	0.3437	0.3313
			$\mathbf{C}_{\mathbf{YZ}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	0.5391	0.5702	0.5543	0.5484
30%	2	Multi-collocated	$\mathbf{C}_{\mathbf{Y}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	*0.1373*	*0.1277*	0.1027	0.0993
			$\mathbf{C}_{\mathbf{YZ}\|(\mathbf{Y}_0,\mathbf{Z}_0)}$	0	0	0	0

- Visiting sequence: midpoint displacement
- Moments of the conditional distributions determined by either strictly collocated or by multi-collocated simple co kriging. In the former case, the components of $\mathbf{Y}$ are simulated sequentially by using a moving neighborhood containing up to 10, 20, 50 or 100 previously simulated Y variables, a single Z variable (the one collocated with the Y variable targeted for simulation) and, for conditional cosimulation, all the original Y data. In the latter case (multi-collocated co kriging), we also consider the Z variables collocated with the selected Y variables.

In each case, the expectations of $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{Y}}$ perfectly match the model expectations (zero in the non-conditional cases; co kriging predictions in the conditional cases, see Appendix 1). The accuracy of cosimulation is assessed by calculating the standardized Frobenius norms of $\mathbf{C}_{\tilde{\mathbf{Y}}} - \mathbf{C}_{\mathbf{Y}}$ and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}} - \mathbf{C}_{\mathbf{YZ}}$ or $\mathbf{C}_{\tilde{\mathbf{Y}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{Y}|(\mathbf{Y}_0,\mathbf{Z}_0)}$ and $\mathbf{C}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} - \mathbf{C}_{\mathbf{YZ}|(\mathbf{Y}_0,\mathbf{Z}_0)}$, depending on whether or not conditioning data are considered (Table 6).

In every case, it is observed that cosimulation with strictly collocated co kriging leads to poor results due to the loss of information caused by discarding data on the covariate (Z) (Appendix 2). In contrast, cosimulation with multi-collocated co kriging leads to a more accurate reproduction of the direct covariance (sometimes, better than for univariate simulation, as per Tables 1 and 5) and a perfect reproduction of the cross covariance. The latter is explained because of Eqs. 16 or 23, accounting for the following facts:

- The intrinsic correlation model (Eq. 27) being a particular case of reverse Markov coregionalization model (Eq. 31 in Appendix 2), Eq. 32 holds, even when working in a moving neighborhood: $-\mathbf{\Lambda}^{-1}\,\mathbf{\Omega} = \rho\mathbf{I}$ or $-\mathbf{\Lambda}_Y^{-1}\,\mathbf{\Omega}_Y = \rho\mathbf{I}$
- $\mathbf{Z}$ is simulated in a unique neighborhood, so that $\mathbf{C}_{\tilde{\mathbf{Z}}} = \mathbf{C}_{\mathbf{Z}}$ or $\mathbf{C}_{\tilde{\mathbf{Z}}|(\mathbf{Y}_0,\mathbf{Z}_0)} = \mathbf{C}_{\mathbf{Z}|(\mathbf{Y}_0,\mathbf{Z}_0)}$.

4 Conclusions

In this paper, we presented a methodology to calculate the theoretical first- and second-order moments (expectation and variance–covariance matrix) of a Gaussian random vector simulated by the sequential algorithm. In practice, these moments may deviate from the theoretically expected ones, depending on the implementation of the simulation algorithm, in particular on the visiting sequence and on the number of variables selected in the moving neighborhood.

Although limited, the numerical experiments show the following:

1) In general, the model covariance matrix is better reproduced by using a midpoint displacement sequence, by increasing the number of data in the neighborhood and/or by simulating separately the nugget effect component.
2) When using ordinary kriging, the model covariance matrix is poorly reproduced with the direct sequential approach, but well-reproduced with a two-step approach.

3) In the multivariate case, under an intrinsic correlation model, the direct and cross covariance matrices are better reproduced when using multi-collocated co kriging than when using strictly collocated co kriging.

The proposed approach can also help to determine, *prior to generating any realization*, suitable implementation settings in order to accurately reproduce the target moments: which visiting sequence should be chosen and how many initial conditioning data and previously simulated variables should be included in the moving neighborhood, including covariates in the multivariate case. To this end, the following test strategy is suggested:

- Restrict the experiments to a grid size adapted to the computer numerical capacity; for instance, the experiments presented in Section 3 considered a grid with 4,225 nodes. The implementation settings found with such a grid are expected to still be appropriate with larger grids.
- Select a visiting sequence for simulation. Following the experimental results obtained in Section 3.3, it is recommended to choose a midpoint displacement or, in the more general case when the grid size is not a power of 2, a multiple-grid sequence [24, 40].
- Perform numerical tests with moving neighborhoods containing an increasing number of variables, until the mismatch measure η (Eq. 26) is less than, say, 5% or until it reaches a certain plateau. If the reproduction of the model covariance matrix is unacceptable (mismatch measure η too high), some alternatives can be explored, such as a separate simulation of the nugget component, a two-step simulation or another choice of the visiting sequence.

For instance, according to the results given in Table 1, for non-conditional simulation of a Gaussian random field with a spherical covariance of range 10 and with a midpoint displacement sequence, a neighborhood containing 50 previously simulated variables is sufficient, while this number can be reduced to 20 variables if the range of the spherical covariance is 40. The same neighborhoods can be kept in the presence of a 30% relative nugget effect, provided that the nugget component is simulated separately.

Acknowledgements This research was funded by the Chilean Fund for Science and Technology Development (FONDECYT 1090013) and by Chilean MECESUP UCN0711. The authors are grateful to the anonymous reviewers for their comments on a former version of this article, and to the Advanced Laboratory for Geostatistical Supercomputing (ALGES) at the University of Chile for supporting this research.

Appendix 1: Expectation of conditional simulation

In the scope of sequential simulation with a moving neighborhood implementation and in the presence of conditioning data $\mathbf{Y}_0$ (Section 2.2), it is of interest to determine whether or not the expectation of the simulated vector $\tilde{\mathbf{Y}}$ could coincide with the kriging of $\mathbf{Y}$ from $\mathbf{Y}_0 = (Y_0, Y_{-1}, \ldots Y_{1-p})^T$. To this end, consider the weighting matrix defined in Eq. 9:

$$\mathbf{W} = -\mathbf{\Lambda}^{-1}\mathbf{\Lambda}_0 \tag{28}$$

so that $-\mathbf{\Lambda}\mathbf{W} = \mathbf{\Lambda}_0$. Accounting for the definition of matrices $\mathbf{\Lambda}$ and $\mathbf{\Lambda}_0$, this is equivalent to writing:

$$\forall i = 1 \ldots n, \forall k = 1 \ldots p, \lambda_{k-p,i} + \sum_{j=1}^{i-1} \lambda_{j,i} w_{jk} = w_{ik} \tag{29}$$

with $\lambda_{j,i}$ the weight assigned to Y_j when kriging Y_i (zero if Y_j is not selected in the kriging neighborhood associated with Y_i).

Assume that, for any $i \in \{1 \ldots n\}$, this neighborhood includes all the conditioning dataset ($\mathbf{Y}_0$) as well as a subset of the previously simulated variables $\{Y_j,\ j \in S_i\}$ with $S_i \subseteq \{1 \ldots i\text{–}1\}$. Equation 29 then reduces to:

$$\forall i = 1 \ldots n, \forall k = 1 \ldots p, \quad \lambda_{k-p,i} + \sum_{j \in S_i} \lambda_{j,i} w_{jk} = w_{ik} \tag{30}$$

By using the additivity relationship between kriging weights [21], it is seen that Eq. 30 is fulfilled when, for any $i \in \{1 \ldots n\}$ and $k \in \{1 \ldots p\}$, w_{ik} is the weight assigned to Y_{k-p} when kriging Y_i from $\mathbf{Y}_0$ alone. Accordingly, the conditional expectation of the simulated vector (Eqs. 9 and 28) matches the kriging predictor of $\mathbf{Y}$ from $\mathbf{Y}_0$ and is thus perfectly accurate.

Appendix 2: Markov model and collocated simple co kriging

Consider the non-conditional simulation of two standard, jointly Gaussian random vectors $\mathbf{Z}$ and $\mathbf{Y}$ such that the cross covariance matrix is proportional to the variance–covariance matrix of $\mathbf{Z}$:

$$\mathbf{C_{YZ}} = \rho \mathbf{C_Z} \tag{31}$$

where ρ stands for the correlation coefficient between components of $\mathbf{Y}$ and $\mathbf{Z}$. When $\mathbf{Y}$ is viewed as the main variable and $\mathbf{Z}$ as a covariate, Eq. 31 corresponds to the

so-called *reverse Markov* [8] or *Markov 2* coregionalization model [27].

Let us assume that $\mathbf{Z}$ is simulated prior to $\mathbf{Y}$ and that sequential cosimulation is performed using simple co kriging in a unique neighborhood, so that the model covariance matrices are perfectly reproduced. Equations 16 and 31 entail the following identity:

$$\mathbf{\Omega} = -\rho\mathbf{\Lambda} \tag{32}$$

In particular, $\mathbf{\Omega}$ is a triangular matrix: for $i = 1...\ n$, the components of $\mathbf{Z}$ with indexes greater than i are assigned zero co kriging weights and can, therefore, be discarded from the neighborhood without any loss of accuracy. Full co kriging boils down to *multi-collocated co kriging* [36, 37, 39].

It is also of interest to determine whether or not full co kriging under a reverse Markov model (Eq. 31) could reduce to *strictly collocated co kriging* [2, 42], in which only one $\mathbf{Z}$-component (the one collocated with the Y-component to simulate) is used in the co kriging system. In such a case, $\mathbf{\Omega}$ is a diagonal matrix. Owing to Eq. 32, this happens in full co kriging in the following situations:

- $\rho = 0$, i.e., $\mathbf{Z}$ and $\mathbf{Y}$ are independent Gaussian vectors.
- $\mathbf{\Lambda}$ is a diagonal matrix: for any pair (i, j) such that $1 \leq j < i \leq n$, the co kriging weight assigned to Y_j when predicting Y_i vanishes. One, therefore, has (Eq. 15):

$$\mathbf{C_Y} = \mathbf{\Delta} + \rho^2\,\mathbf{C_Z} \tag{33}$$

with $\mathbf{\Delta} = \mathbf{\Lambda}^{-1}\left(\mathbf{\Lambda}^{-1}\right)^T$ a diagonal matrix. Since $\mathbf{Y}$ and $\mathbf{Z}$ are standard Gaussian vectors, $\mathbf{\Delta}$ must be equal to $(1 - \rho^2)\mathbf{I}$ in order to ensure a unit variance for the components of $\mathbf{Y}$. Particular cases occur if $\mathbf{C_Y}$ is diagonal (Y_j is uncorrelated with Y_i) or if $|\rho| = 1$ (Z_i perfectly screens out Y_j in the co kriging system) [4].

Except for these few specific situations, already pointed out by Rivoirard [36], strictly collocated co kriging is not identical to full co kriging.

References

1. Alabert, F.: The practice of fast conditional simulations through the LU decomposition of the covariance matrix. Math. Geol. **19**(5), 369–386 (1987)
2. Almeida, A.S., Journel, A.G.: Joint simulation of multiple variables with a Markov-type coregionalization model. Math. Geol. **26**(5), 565–588 (1994)
3. Anderson, T.W.: An introduction to Multivariate Statistical Analysis, 3rd edn. Wiley, Hoboken (2003)
4. Babak, O., Deutsch, C.V.: An intrinsic model of coregionalization that solves variance inflation in collocated co kriging. Comput. Geosci. **35**(3), 603–614 (2009)
5. Box, G.E.P., Jenkins, G.M.: Time Series Analysis; Forecasting and Control. Holden Day, Oakland (1976)
6. Chilès, J.P., Allard, D.: Stochastic simulation of soil variations. In: Grunwald, S. (ed.) Environmental Soil-Landscape Modeling: Geographic Information Technologies and Pedometrics, pp. 289–321. CRC, Boca Raton (2005)
7. Chilès, J.P., Delfiner, P.: Discrete exact simulation by the Fourier method. In: Baafi, E.Y., Schofield, N.A. (eds.) Geostatistics Wollongong, pp. 258–269. Kluwer, Dordrecht (1997)
8. Chilès, J.P., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. Wiley, New York (1999)
9. David, M.: Handbook of Applied Advanced Geostatistical Ore Reserve Estimation. Elsevier, Amsterdam (1988)
10. Davis, M.W.: Production of conditional simulations via the LU triangular decomposition of the covariance matrix. Math. Geol. **19**(2), 91–98 (1987)
11. Delhomme, J.P.: Spatial variability and uncertainty in groundwater flow parameters: a geostatistical approach. Water Resour. Res. **15**(2), 269–280 (1979)
12. De Marsily, G.: Quantitative Hydrogeology: Groundwater Hydrology for Engineers. Academic Press, San Diego (1986)
13. Deutsch, C.V.: Kriging in a finite domain. Math. Geol. **25**(1), 41–52 (1993)
14. Deutsch, C.V.: Kriging with strings of data. Math. Geol. **26**(5), 623–638 (1994)
15. Deutsch, C.V., Journel, A.G.: GSLIB: Geostatistical Software Library and User's Guide. Oxford University Press, New York (1992)
16. Dietrich, C.R., Newsam, G.N.: A fast and exact method for multidimensional Gaussian stochastic simulations. Water Resour. Res. **29**(8), 2861–2869 (1993)
17. Dimitrakopoulos, R., Luo, X.: Generalized sequential Gaussian simulation on group size ν and screen-effect approximations for large field simulations. Math. Geol. **36**(5), 567–591 (2004)
18. Emery, X.: Testing the correctness of the sequential algorithm for simulating Gaussian random fields. Stoch. Environ. Res. Risk Assess. **18**(6), 401–413 (2004)
19. Emery, X.: Uncertainty modeling and spatial prediction by multi-Gaussian kriging: Accounting for an unknown mean value. Comput. Geosci. **34**(11), 1431–1442 (2008)
20. Emery, X.: Statistical tests for validating geostatistical simulation algorithms. Comput. Geosci. **34**(11), 1610–1620 (2008)
21. Emery, X.: The kriging update equations and their application to the selection of neighboring data. Comput. Geosci. **13**(3), 269–280 (2009)
22. Emery, X.: Multi-Gaussian kriging and simulation in the presence of an uncertain mean value. Stoch. Environ. Res. Risk Assess. **24**(2), 211–219 (2010)
23. Emery, X., Lantuéjoul, C.: TBSIM: A computer program for conditional simulation of three-dimensional Gaussian random fields via the turning bands method. Comput. Geosci. **32**(10), 1615–1628 (2006)
24. Gómez-Hernández, J.J., Cassiraga, E.F.: Theory and practice of sequential simulation. In: Armstrong, M., Dowd, P.A. (eds.) Geostatistical Simulations, pp. 111–124. Kluwer, Dordrecht (1994)
25. Isaaks, E.H., Srivastava, R.M.: An Introduction to Applied Geostatistics. Oxford University Press, New York (1989)
26. Journel, A.G.: Geostatistics for conditional simulation of orebodies. Econ. Geol. **69**(5), 673–687 (1974)

27. Journel, A.G.: Markov models for cross-covariances. Math. Geol. **31**(8), 955–964 (1999)
28. Lantuéjoul, C.: Non conditional simulation of stationary isotropic multigaussian random functions. In: Armstrong, M., Dowd, P.A. (eds.) Geostatistical Simulations, pp. 147–177. Kluwer, Dordrecht (1994)
29. Leuangthong, O., McLennan, J.A., Deutsch, C.V.: Minimum acceptance criteria for geostatistical realizations. Nat. Resour. Res. **13**(3), 131–141 (2004)
30. Matérn, B.: Spatial Variation. Springer, Berlin (1986)
31. Matheron, G.: The intrinsic random functions and their applications. Adv. Appl. Probab. **5**, 439–468 (1973)
32. Omre, H., Sølna, K., Tjelmeland, H.: Simulation of random functions on large lattices. In: Soares, A. (ed.) Geostatistics Tróia'92, pp. 179–199. Kluwer, Dordrecht (1993)
33. Pardo-Igúzquiza, E., Chica-Olmo, M.: The Fourier integral method—an efficient spectral method for simulation of random fields. Math. Geol. **25**(2), 177–217 (1993)
34. Rivoirard, J.: Le comportement des poids de krigeage (The behavior of kriging weights). Doctoral dissertation, Paris School of Mines (1984)
35. Rivoirard, J.: Two key parameters when choosing the kriging neighborhood. Math. Geol. **19**(8), 851–856 (1987)
36. Rivoirard, J.: Which models for collocated co kriging? Math. Geol. **33**(2), 117–131 (2001)
37. Rivoirard, J.: On some simplifications of co kriging neighborhood. Math. Geol. **36**(8), 899–915 (2004)
38. Shinozuka, M., Jan, C.M.: Digital simulation of random processes and its applications. J. Sound Vib. **25**(1), 111–128 (1972)
39. Subramanyam, A., Pandalai, H.S.: Data configurations and the co kriging system: simplification by screen effects. Math. Geosci. **40**(4), 425–443 (2008)
40. Tran, T.T.: Improving variogram reproduction on dense simulation grids. Comput. Geosci. **20**(7–8), 1161–1168 (1994)
41. Wackernagel, H.: Multivariate Geostatistics—An Introduction with Applications. Springer, Berlin (2003)
42. Xu, W., Tran, T.T., Srivastava, R.M., Journel, A.G.: Integrating seismic data in reservoir modeling: the collocated co kriging alternative. In: 67th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, pp. 833–842. SPE paper 24742 (1992)
43. Yarus, J.M., Chambers, R.L.: Stochastic Modeling and Geostatistics. Principles, Methods, and Case Studies. American Association of Petroleum Geologists, Tulsa (1994)