Math Geosci (2018) 50:97–120
https://doi.org/10.1007/s11004-017-9699-5

Which Path to Choose in Sequential Gaussian Simulation

Raphaël Nussbaumer[1] · **Grégoire Mariethoz**[2] · **Erwan Gloaguen**[3] · **Klaus Holliger**[1]

Received: 20 April 2017 / Accepted: 24 July 2017 / Published online: 9 August 2017

Abstract Sequential Gaussian Simulation is a commonly used geostatistical method for populating a grid with a Gaussian random field. The theoretical foundation of this method implies that all previously simulated nodes, referred to as neighbors, should be included in the kriging system of each newly simulated node. This would, however, require solving a large number of linear systems of increasing size as the simulation progresses, which, for computational reasons, is generally not feasible. Traditionally, this problem is addressed by limiting the number of neighbors to the ones closest to the simulated node. This does, however, result in artifacts in the realization. The simulation path, that is, the order in which nodes are visited, is known to influence the location and magnitude of these artifacts. So far, few rigorous studies linking the simulation path to the associated biases are available and, correspondingly, recommendations regarding the choice of the simulation path are largely based on empirical evidence. In this study, a comprehensive analysis of the influence of the path on the simulation errors is presented, based on which guidelines for choosing an optimal path were developed. The most common path types are systematically assessed based on the comparison of the simulation covariance matrices with the covariance of the underlying spatial model. Our analysis indicates that the optimal path is defined as the one minimizing the information lost by the omission of neighbors. Classification into clustering paths,

The original version of this article was revised: The spelling of the third author was incorrect. The correct name is Erwan Gloaguen.

✉ Raphaël Nussbaumer
raphael.nussbaumer@unil.ch

1 Institute of Earth Sciences, University de Lausanne, Lausanne, Switzerland

2 Institute of Earth Surface Dynamics, University de Lausanne, Lausanne, Switzerland

3 Centre Eau Terre Environnement, Institut National de la Recherche Scientifique, Québec, Canada

that is, paths simulating consecutively close nodes, and declustering paths, that is, paths simulating consecutively distant nodes, was found to be an efficient way of determining path performance. Common examples of the latter are multi-grid, mid-point, and quasi-random paths, while the former include row-by-row and spiral paths. Indeed, clustering paths tend to inadequately approximate covariances at intermediate and large lag distances, because their neighborhood is only composed of nearby nodes. On the other hand, declustering paths minimize the correlation among nodes, thus ensuring that the neighbors are more diverse, and that only weakly correlated neighbors are omitted.

Keywords Visiting sequence · Sequential simulation · Artifact · Covariance matrix · Spiral path · Random path · Multi-grid path · Quasi-random path

1 Introduction

Sequential Gaussian Simulation (SGS) is a popular technique to populate stochastically a grid with a Gaussian random field (Johnson 1987; Journel 1989; Isaaks 1991; Deutsch and Journel 1992; Gómez-Hernández and Cassiraga 1994). In practice, SGS consists of visiting sequentially each node of the grid, computing the conditional probability distribution based on existing values using kriging, and assigning a value sampled from this distribution to the target node. This stochastic simulation technique has been applied in a wide range of disciplines, such as reservoir simulation (Verly 1993), mining (Dimitrakopoulos et al. 2002; Zhao et al. 2007), hydrogeology (Lee et al. 2007), hydrology (Delbari et al. 2009), geophysics (Day-Lewis and Lane 2004; Hansen et al. 2006; Abdu et al. 2008), soil science (Lin et al. 2001; Goovaerts 2001), environmental science (Juang et al. 2004), and ecology (Mowrer 1997).

A main drawback of this technique is its high computational cost originating from the computation of the kriging estimation of each node. The most widespread solution to this problem is the so-called limited neighborhood approach, that is, to keep only a limited number of conditioning nodes in the kriging estimation. However, the omission of information associated with this approach biases the kriging estimate, which in turn causes artifacts in the realizations. The simulation path, that is, the order in which nodes are simulated, has been recognized to influence the correctness of the simulation, however, these observations are largely based on empirical evidence (Tran 1994; McLennan 2002). A comprehensive study on how the sequential simulation path affects the occurrence and magnitude of bias is as of yet not available. This kind of information is, however, essential to allow for an educated choice of the path.

Here, this topic was addressed by presenting a comprehensive analysis of the connection between the simulation path and its impact on the deterioration of the realizations. Based on this, the most efficient path to preserve the desired simulation properties was investigated. Commonly used paths are thoroughly described, numerically assessed, and compared based on the evaluation of the simulation covariance matrix (Emery and Peláez 2011), thus allowing to draw general conclusions on the usefulness of each path.

The paper is organized as follows: Sect. 2 reviews the methodological background of SGS, limited neighborhood, and the simulation path; Sect. 3 presents the method

used to quantify the bias in SGS; Sect. 4 attempts to provide the characteristics of an optimal path; Sect. 5 presents the analysis of various common paths; finally, Sect. 6 discusses limitations and constraints related to the choice of a given path.

2 Sequential Gaussian Simulation, Limited Neighborhood and Simulation Path

2.1 Background of Sequential Gaussian Simulation

SGS generates realizations $z^{(l)}(\mathbf{u})$ of a regionalized Gaussian random field $Z(\mathbf{u})$ at a discrete set of locations $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ by iteratively sampling a value for the field at each of these locations $z^{(l)}(\mathbf{u}_i)$. Based on the knowledge of the covariance C_Z, the estimation of $Z(\mathbf{u}_i)$ is performed by inferring the conditional mean and variance using kriging based on the previously simulated locations, often referred to as neighbors. With a zero-mean random field, SGS can be written as (Chilès and Delfiner 1999)

$$z^{(l)}(\mathbf{u}_i) = \sum_{j=1}^{i-1} \lambda_j(\mathbf{u}_i) z^{(l)}(\mathbf{u}_j) + \sigma_E(\mathbf{u}_i) U_i, \quad \forall i = 1, \ldots, N, \tag{1}$$

where λ_j are the kriging weights, σ_E^2 is the kriging variance error, and U_i is a value sampled from a standard Gaussian random variable. When the simulation starts with some known initial values $\{z(\mathbf{u}_1), \ldots, z(\mathbf{u}_{N_0})\}$, commonly referred to as hard data, Eq. (1) becomes

$$z^{(l)}(\mathbf{u}_i) = \sum_{k=1}^{N_0} \lambda_k(\mathbf{u}_i) z(\mathbf{u}_k) + \sum_{j=N_0+1}^{i-1} \lambda_j(\mathbf{u}_i) z^{(l)}(\mathbf{u}_j) + \sigma_E(\mathbf{u}_i) U_i, \quad \forall i = N_0+1, \ldots, N. \tag{2}$$

Since SGS is typically used on grids, our study is limited to gridded coordinate systems and, correspondingly, the terminology "node", rather than "point" is employed. In this study, the term "estimation" was favored in the context of kriging as opposed to "prediction", because of the temporal connotation of the latter (Chilès and Delfiner 1999).

2.2 Computational Efficiency, Limited Neighborhood, and Bias

A major drawback of SGS is that, in order to be mathematically rigorous, it needs to include all previously simulated nodes in the kriging estimation. As the simulation goes on, the number of previously simulated nodes grows, which in turn increases the size of the covariance matrix. Because the computational complexity of solving a kriging system of n nodes with the widely used LU, QR, or Cholesky solvers is $O(n^3)$ (Trefethen and Bau III 1997), the simulation of a grid with N nodes has a complexity of $O(N^4)$ (Dimitrakopoulos and Luo 2004; Srinivasan et al. 2008). The corresponding computational cost is excessive for typical applications.

To alleviate this problem, it is common practice to consider only a small number of representative neighbors in the estimation of each simulated node. This is generally referred to as the moving or limited neighborhood approach, as opposed to the unique or full neighborhood approach when all nodes are considered. Using a maximum of n neighbors reduces the overall computational complexity to $O(N^3 n)$ (Dimitrakopoulos and Luo 2004). The retained neighbors are typically chosen based on their proximity to the estimated node, taking into account the correlation range and orientation. Indeed, due to the screening effect (Omre et al. 1993; Chilès and Delfiner 1999), many of the distant nodes have a kriging weight close to zero, and their absence implies only a small approximation.

Although inevitable in practice, the use of a limited neighborhood creates artifacts in the simulated fields (Meyer 2004). By omitting certain nodes, the correlation of these nodes with the simulated node is neglected, and the correlation between the omitted and retained nodes. This is reflected by a bias in the simulation covariance matrix and leads to errors in the simulation of the nodes. As SGS re-uses previously simulated nodes as neighbors, the error is propagated and cumulated, thus creating significant artifacts in the final realizations.

In this paper, "error" is used as a generic term for any deviation from the true value, "artifact" is used for the resulting perceived error of a process, and "bias" is used for a systematic error, measured as the difference of the expected value of a variable to its true value. More specifically, in the context of this study, errors in SGS are due to biases in the sampling of each variable and result in tangible artifacts in the final realizations.

2.3 Simulation Path

The decomposition of the joint probability into conditional probabilities used in SGS does not assume any specific order in which the nodes are simulated. SGS with a unique neighborhood is therefore independent of the simulation path (Goovaerts 1997) so that any type of path can be used without creating artifacts. However, with a limited neighborhood, the simulation path determines which nodes are available as neighbors. Hence, the simulation path influences indirectly the location and magnitude of the bias, resulting in the covariance matrix not being completely honored in the realizations. By extension, the path also defines which of these nodes are not taken into account in the neighborhood, and therefore, which covariances are not correctly reproduced. Moreover, in determining which nodes are used for conditioning, the path can also influence the effect of cumulative biases, where nodes simulated with a bias are used as conditioning, thus propagating their error to the newly simulated nodes. Consequently, while the limited neighborhood is the origin of bias, the simulation path can be considered as the vector spreading the bias across the simulation.

In this paper, six types of paths commonly used in SGS (Fig. 1) were explored and those characteristics are discussed in the following.

(i) The row-by-row path visits consecutive adjacent nodes (Daly 2005). Its advantage is to ensure a nearly constant structure of neighboring nodes, thus allowing the re-use of the same kriging weights and saving computational time (Deutsch

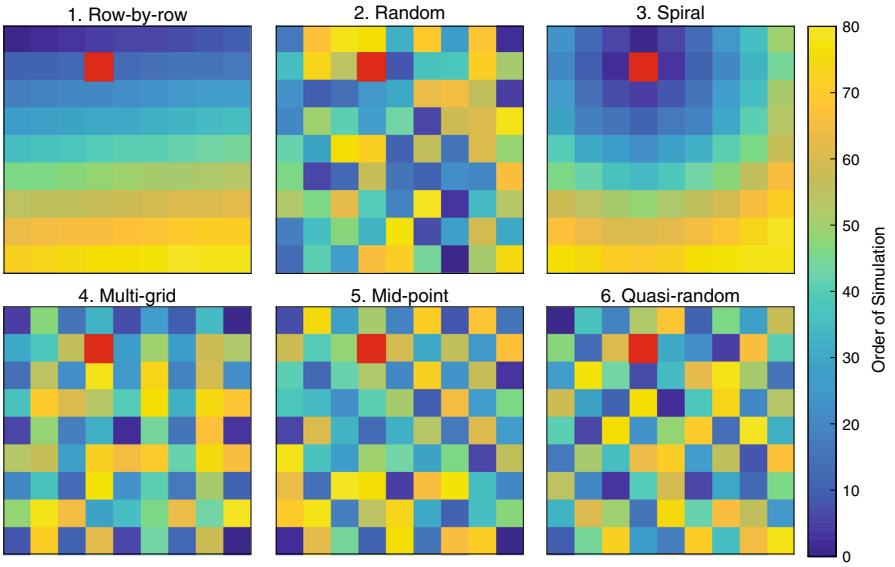

Fig. 1 Illustration of the 6 types of paths considered in this study on a 9×9 grid with one hard datum (*red square*). The *color* denotes the order in which the nodes are simulated

and Journel 1992; McLennan 2002). However, because of its strong characteristic patterns, this path is qualified as regular and leads to artifacts in the chosen direction of the path (Deutsch and Journel 1992; Gómez-Hernández and Cassiraga 1994). McLennan (2002) does, however, note that the suspicions regarding the origin of the bias are not sufficiently well documented to be conclusive.

(ii) The random path entirely shuffles the order in which nodes are simulated. It is commonly recommended in the literature, because it does not seem to assume any specific structure and correspondingly, is viewed as minimizing the creation of artifacts (Deutsch and Journel 1992; Gómez-Hernández and Journel 1993; Goovaerts 1997).

(iii) The spiral path sorts the nodes according to their distance to hard data. This naturally results in a path spiralling away from the hard data. McLennan (2002) explains the use of this path by the perception that simulating nodes near hard data would honor their influence more faithfully so that a spiral path would improve the representativeness of hard data and prevent artifacts.

(iv) A multi-grid path refers to a nested grid system where the simulation starts at the coarsest scale and moves to progressively finer scales by re-using the values simulated at the previous scale. At each grid scale, the simulation path can follow any of the types described above. In our study, a random path was used at each scale. From the early days of sequential simulation, it has been recognized that operating directly on a fine grid may not adequately reproduce large-scale features (Deutsch and Journel 1992; Gómez-Hernández and Journel 1993). Correspondingly, Gómez-Hernández and Journel (1993) proposed to use a multi-grid approach by first simulating nodes located on a coarser grid and the remaining nodes on subsequently refined grids. Tran (1994) and McLennan (2002) provide

empirical tests showing the improvement of covariance reproduction when the domain is simulated by subsequently refined multiple grids. Conversely, Emery (2004) argues that multi-grid paths only delay the introduction of bias and that the number of pairs of nodes with erroneous covariance is not reduced.

(v) The mid-point path consecutively simulates the node that has the largest distance to its closest neighbor (Fournier et al. 1982; Barnsley et al. 1988; Emery and Peláez 2011). For unconditional simulations, this path is similar to the multi-grid path if the first simulated node is located in one of the corners, with the exception of preferentially simulating the nodes located in the diagonal of the previous grid level, resulting in the so-called diamond-square algorithm (Fournier et al. 1982). Emery and Peláez (2011) find a better reproduction of the second-order statistics for a mid-point path compared to the row-by-row or spiral paths. Omre et al. (1993) use a mid-node path to minimize simulation errors by allowing to use only the closest conditioning node in all directions. This will be discussed in more detail in the context of the screening effect (Sect. 4) and the Markov property (Sect. 5.1.1).

(vi) The quasi-random path is an alternative to the random path where the grid is visited more homogeneously (Chilès and Delfiner 1999). It is rarely used, and thus relatively unknown in SGS. Our implementation of this path relies on transforming a two-dimensional Halton sequence (Halton 1960; Kocis and Whiten 1997) into coordinates of the considered grid by using an acceptance/rejection method to avoid re-visiting nodes.

Amongst the paths described above, the row-by-row and spiral paths can be considered as deterministic since the same path will be used from one realization to another. The realizations produced are nevertheless not deterministic because the value of each simulated node is drawn using a different seed number. All other paths can be randomized, that is, a different specific path of the same path type is used for each realization. Note that here the term random path refers to the fact that the order in which the nodes are simulated is random while the term randomized path refers to each realization being generated with a different path. Randomizing the random, multi-grid and quasi-random path types is possible by changing the seed number, while the mid-point path requires the randomization of the selection of nodes which have an equal distance to their closest neighbors.

3 Quantification of Bias

As the underlying random field Z is assumed to be Gaussian, the evaluation of simulation can be based solely on reproducing its first- and second-order statistics defined by the mean and the covariance (Leuangthong et al. 2004). The former can be assessed based on the difference between the expected value of the simulation $\mathrm{E}[Z]$ and the kriging estimate. The latter is fully defined by the discrepancy between the simulation covariance matrix $\mathbf{C}_{Z^{(l)}}$ and the model covariance matrix $\mathbf{C}_Z$.

Emery and Peláez (2011) present an elegant way to calculate theoretically the simulation covariance matrix from the kriging weights and the kriging errors variance. The strength of this approach is that it computes the exact covariance error for every

single pair of nodes. In comparison, traditional covariance-function-based empirical assessment techniques (Emery 2004; Leuangthong et al. 2004; Safikhani et al. 2017) struggle to account for the inherent fluctuation of the node statistics and perform an averaging of the covariance errors, which in turn results in a smoothing of the covariance.

3.1 Computation of the Simulation Covariance Matrix for Unconditional Simulations

Following Emery and Peláez (2011), SGS (Eq. 1) can be re-written in matrix form to isolate $\mathbf{U}$ on the left-hand side and to combine the previously and currently simulated values on the right-hand side

$$\mathbf{U} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{\lambda_1^{n-1}}{\sigma_{n-1}} & \cdots & \frac{1}{\sigma_{n-1}} & 0 \\ -\frac{\lambda_1^{n}}{\sigma_n} & \cdots & -\frac{\lambda_{n-1}^{n}}{\sigma_n} & \frac{1}{\sigma_n} \end{bmatrix} \mathbf{z}^{(l)} = \mathbf{\Lambda} \mathbf{z}^{(l)}, \tag{3}$$

where $\lambda_j^i = \lambda_j(\mathbf{u}_i)$ and $\sigma_i = \sigma_E(\mathbf{u}_i)$. $\mathbf{\Lambda}$ is referred to as the lambda matrix and is built exclusively with kriging weights and variances.

If $E\left[\mathbf{z}^{(l)}\right] = 0$, the simulation covariance matrix can be computed based only on $\mathbf{\Lambda}$, as $\mathbf{U}$ has independent values and, therefore, its covariance matrix is equal to the identity matrix

$$\begin{aligned} \mathbf{C}_{Z^{(l)}} &= E\left[\left(\mathbf{z}^{(l)} - E\left[\mathbf{z}^{(l)}\right]\right)\left(\mathbf{z}^{(l)} - E\left[\mathbf{z}^{(l)}\right]\right)^T\right] \\ &= E\left[\mathbf{z}^{(l)}\mathbf{z}^{(l)T}\right] \\ &= E\left[\left(\mathbf{\Lambda}^{-1}\mathbf{U}\right)\left(\mathbf{\Lambda}^{-1}\mathbf{U}\right)^T\right] \\ &= \mathbf{\Lambda}^{-1} E\left[\mathbf{U}\mathbf{U}^T\right](\mathbf{\Lambda}^{-1})^T \\ &= \mathbf{\Lambda}^{-1}(\mathbf{\Lambda}^{-1})^T. \end{aligned} \tag{4}$$

In practice, $\mathbf{\Lambda}$ is constructed line-by-line according to the simulation path. This results in a sparse lower-triangular square matrix, because the computed weights correspond only to the previously simulated nodes.

3.2 Computation of the Simulation Covariance Matrix for Conditional Simulations

SGS with hard data (Eq. 2) can be similarly rewritten by separating the weight of hard data $\mathbf{\Lambda_0}(n \times n_0)$ and the weight of the previously simulated data $\mathbf{\Lambda}$ $((n-n_0) \times (n-n_0))$

$$\mathbf{U} = \begin{bmatrix} -\frac{\lambda_1^{n_0+1}}{\sigma_{n_0+1}} & \cdots & -\frac{\lambda_{n_0}^{n_0+1}}{\sigma_{n_0+1}} \\ \vdots & \ddots & \vdots \\ -\frac{\lambda_1^{n}}{\sigma_{n}} & \cdots & -\frac{\lambda_{n_0}^{n}}{\sigma_{n}} \end{bmatrix} \mathbf{z}_0 + \begin{bmatrix} \frac{1}{\sigma_{n_0+1}} & 0 & \cdots & & 0 \\ -\frac{\lambda_{n_0+1}^{n_0+2}}{\sigma_{n_0+2}} & \frac{1}{\sigma_{n_0+2}} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\frac{\lambda_{n_0+1}^{n-1}}{\sigma_{n-1}} & \cdots & -\frac{\lambda_{n-2}^{n-1}}{\sigma_{n-1}} & \frac{1}{\sigma_{n-1}} & 0 \\ -\frac{\lambda_{n_0+1}^{n}}{\sigma_{n}} & \cdots & & -\frac{\lambda_{n-1}^{n}}{\sigma_{n}} & \frac{1}{\sigma_{n}} \end{bmatrix} \mathbf{z}^{(l)}. \tag{5}$$

The simulated values can then be isolated

$$\mathbf{z}^{(l)} = \boldsymbol{\Lambda}^{-1} \left(\mathbf{U} - \boldsymbol{\Lambda}_0 \mathbf{z}_0 \right), \tag{6}$$

and incorporated into the conditional covariance

$$\begin{aligned} \mathbf{C}_{Z^{(l)}|Z_0} &= E\left[\left(\mathbf{z}^{(l)} - E[\mathbf{z}^{(l)} \mid \mathbf{z}_0] \right) \left(\mathbf{z}^{(l)} - E[\mathbf{z}^{(l)} \mid \mathbf{z}_0] \right)^T \mid \mathbf{z}_0 \right] \\ &= E\left[\mathbf{z}^{(l)} \mathbf{z}^{(l)\,T} \mid \mathbf{z}_0 \right] - E\left[\mathbf{z}^{(l)} \mid \mathbf{z}_0 \right] E\left[\mathbf{z}^{(l)\,T} \mid \mathbf{z}_0 \right] \\ &= \boldsymbol{\Lambda}^{-1} \left(E\left[\mathbf{U}\mathbf{U}^T + (\boldsymbol{\Lambda}_0 \mathbf{z}_0)(\boldsymbol{\Lambda}_0 \mathbf{z}_0)^T - 2\mathbf{U}\boldsymbol{\Lambda}_0 \mathbf{z}_0 \mid \mathbf{z}_0 \right] \right) (\boldsymbol{\Lambda}^{-1})^T \\ &- E\left[\boldsymbol{\Lambda}^{-1}\mathbf{U} - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_0 \mathbf{z}_0 \mid \mathbf{z}_0 \right] E\left[\mathbf{U}^T (\boldsymbol{\Lambda}^{-1})^T - \mathbf{z}_0^T \boldsymbol{\Lambda}_0^T (\boldsymbol{\Lambda}^{-1})^T \mid \mathbf{z}_0 \right] \\ &= \boldsymbol{\Lambda}^{-1} \left(\mathbf{I} + \boldsymbol{\Lambda}_0 \mathbf{z}_0 \mathbf{z}_0^T \boldsymbol{\Lambda}_0^T - 0 \right) (\boldsymbol{\Lambda}^{-1})^T - \boldsymbol{\Lambda}^{-1} \boldsymbol{\Lambda}_0 \mathbf{z}_0 \mathbf{z}_0^T \boldsymbol{\Lambda}_0^T (\boldsymbol{\Lambda}^{-1})^T \\ &= \boldsymbol{\Lambda}^{-1} (\boldsymbol{\Lambda}^{-1})^T, \end{aligned} \tag{7}$$

which turns out to be identical to the unconditional case (Eq. 4).

As opposed to the unconditional case, the expected value of the simulated data $\mathrm{E}\left[\mathbf{z}^{(l)}\right]$ is not always equal to zero in conditional simulations

$$E[\mathbf{z}^{(l)} \mid \mathbf{z}_0] = \boldsymbol{\Lambda}^{-1} E\left[\mathbf{U} - \boldsymbol{\Lambda}_0 \mathbf{Z}_0 \mid \mathbf{z}_0 \right] = -\boldsymbol{\Lambda}^{-1} \boldsymbol{\Lambda}_0 \mathbf{z}_0. \tag{8}$$

3.3 Illustration

In order to demonstrate the exactness of the evaluation of the covariance matrix, 500 realizations of a one-dimensional grid of 257 nodes were simulated using a spherical covariance function with a range of 10. Figure 2 illustrates that the empirical covariance function agrees with the simulation covariance function. Both diverge from the model covariance function because of the use of a limited neighborhood. The calculation of the simulation covariance matrix thus proves to be accurate.

4 Characteristics of Optimal Paths

Based on the discussion on the origin of the bias in Sects. 2.2 and 2.3, the best path can be adequately characterized as the one minimizing the influence of the approximation

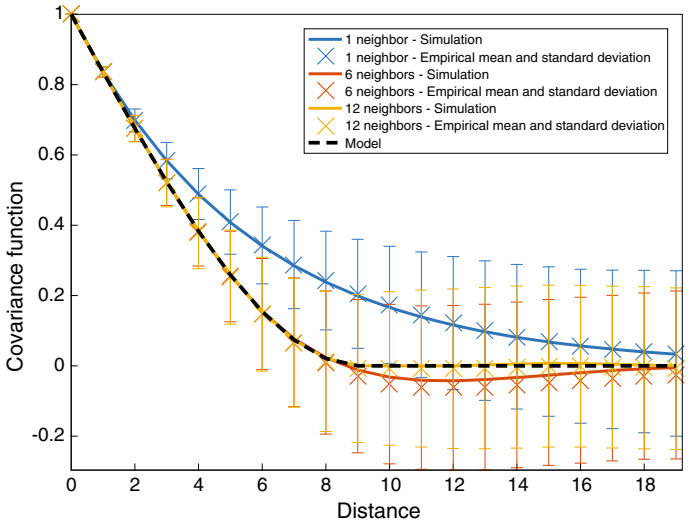

Fig. 2 Comparison of covariance function computed with the simulation covariance matrix (*solid lines*) and the empirical covariance function based on 500 realizations (*crosses* and *error bars*). The simulations were performed on a one-dimensional grid of 257 nodes using a spherical covariance function of range 10 and 3 different neighborhood sizes (1, 6, and 12 nodes). The *dashed line* denotes the model covariance function

resulting from the use of a limited neighborhood. However, this influence is complex and indirect, making it difficult to narrow down a single best simulation path. Here, the analysis of the link between path and limited neighborhood was performed to infer some guidelines as to how to choose the optimal path.

Two levels of analysis are identified and described separately: the node level describing which is the optimal node to simulate in a given configuration of neighbors and, the simulation level characterizing which sequence of nodes minimizes the overall error.

In the following analysis, it is assumed that the neighborhood search strategy is operating by selecting a fixed number of nodes that are closest to the simulated node, regardless of the correlation range. The effects described in the following apply to most traditional covariance models, but may fail for some less common ones such as, for example, the pure nugget model.

At the level of node simulation, minimum bias is achieved by choosing to simulate the node whose conditional probability distribution is the least affected by the absence of neighbors due to the limited neighborhood. This means that the optimal node to simulate is the one whose neighborhood search excludes only unimportant nodes. The importance of a neighbor regarding the simulation of a node is measured by both its kriging weight and the modification of the kriging weights of other neighbors resulting from its removal from the neighborhood. Figures 3a to 3e illustrate five effects influencing the selection of the optimal node to simulate with a simple example. The simulation grids are one-dimensional for a–d and two-dimensional for e. For each example, two cases A and B simulating a different node are compared. Cases A are generally leading to smaller error than cases B. The simulated node is denoted as a

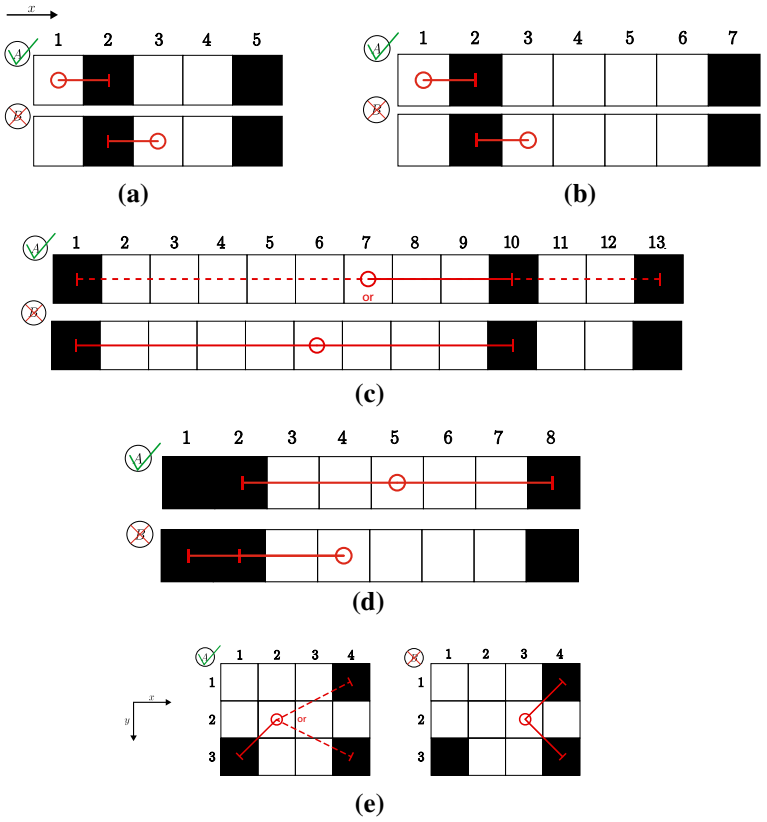

Fig. 3 Illustration of the five effects described above based on the comparison of two cases A and B with a different simulated node denoted as *red circles*. *Black squares* correspond to conditioning nodes, but only those linked to the simulated node with a red line are used in the simulation because of the limited neighborhood. The correlation range r is 5 nodes. The neighborhood size is one node in (**a**, **b**) and two nodes in (**c**–**e**). **a** A is optimal because the distance between the simulated node and the excluded neighbor is larger than for B: $d(x_1, x_5) > d(x_3, x_5)$. **b** In A, x_7 can be safely ignored because $\mathrm{d}(x_1, x_7) \geq r$ and $\mathrm{d}(x_2, x_7) \geq r$. This is not the case for B. **c** In B, although $\mathrm{d}(x_{13}, x_6) \geq r$, x_{13} should be included because of the relay effect with x_{10}: $\mathrm{d}(x_6, x_{10}) < r$ and $\mathrm{d}(x_{10}, x_{13}) < r$. Instead, with a traditional search neighborhood, x_1 would be included because it is closer to x_6 even though it has no influence on the simulated node $\mathrm{d}(x_1, x_6) \geq r$. In A, as both conditional nodes x_{13} and x_1 have the same distance to x_7, either of them could be selected as neighbors (*dotted red line*). Yet, including x_{13} would lead to an unbiased simulation while including x_1 does not. **d** For both A and B, the excluded node has the same distance to the simulated node $\mathrm{d}(x_5, x_1) = \mathrm{d}(x_5, x_8)$. However, because of the screening effect, the weight of x_1 in A is reduced by the presence of x_2 while in B the weight of x_8 is not reduced. **e** Because of the declustering effect, the weights of the two neighbors, (x_4, y_1) and (x_4, y_3) in B are lower than if only one of them was selected. In A, this effect is less pronounced as the two neighbors are further apart. Note that (x_4, y_1) brings more diverse information than (x_4, y_3) because it is further away from (x_1, y_3)

red circle and is linked with red lines to its neighbors. The remaining black nodes are, therefore, neglected neighbors of the simulation. Dashed lines indicate that two neighbors are equidistant. The correlation range is 5 nodes and the neighborhood size is a unique node for a–c and two nodes for d, e.

(i) Decreasing distance effect: The importance of a node decreases as the distance to the simulated node increases. Therefore, the excluded nodes should be as far as possible from the simulated node (Fig. 3a).
(ii) Finite range effect: A node whose distance to both the simulated node and all neighboring nodes is larger or equal than the correlation range can be safely ignored. Therefore, such nodes can be excluded (Fig. 3b).
(iii) Relay effect (Rivoirard 1984; Chilès and Delfiner 1999): A node whose distance to the simulated node is equal to or greater than the range may not be removed without altering the simulation in the presence of an intermediate node correlated to both this node and the simulated node. This effect has various consequences depending on the neighborhood search strategy and the covariance model (Fig. 3c).
(iv) Screening effect (Rivoirard 1984; Deutsch and Journel 1992; Chilès and Delfiner 1999): The information content of a node is reduced by the presence of an intermediate node. Therefore, it is better to exclude a node behind another one than to exclude an isolated node (Fig. 3d).
(v) Declustering effect: selecting two nearby nodes provides less information than selecting two isolated nodes. Therefore, the simulated node including neighbors from various directions is more favorable than clustered neighbors (Fig. 3e).

At the level of the global simulation, the optimal path is not necessarily the combination of all locally optimal nodes. Indeed, choosing the optimal node to simulate at any specific moment may lead to larger errors later in the simulation. Instead, the optimal path should take into account the fact that a simulated node subsequently becomes a conditioning node. In addition, the effect of cumulative bias can become significant at this level and, therefore, the optimal path should be built to avoid that a node with a large bias becomes a conditioning node. Moreover, the comparison of paths relies on a multi-dimensional error analysis because each path leads to different errors for each node. Therefore, the comparisons of different paths also depends on the way these errors are aggregated.

At both the node and the simulation levels, the neighborhood size and the shape of the covariance function model have a strong influence on the optimal path. For instance, a smaller neighborhood may result in excluding an influential node, thus changing the optimal node to choose. In addition, the boundary of the grid also influences the node selection, thus making the optimal path sensitive to the grid size.

Despite the challenges of generalizing the design of an optimal path, the effects described above reveal one of its key characteristics: maximizing the distance between successively simulated nodes. Indeed, when the correlation between nodes is minimized, the relay, screening, and declustering effects are also minimized.

5 Assessment of the Different Types of Paths

Based on the above description, two general families of paths can be identified: clustering and declustering paths. Row-by-row and spiral paths can be classified as clustering paths as they simulate consecutively the nearest nodes thus creating clusters. Con-

versely, the multi-grid, mid-node, and quasi-random paths tend to spread-out the simulated nodes and hence are referred to as declustering paths.

For the following analysis, six commonly used covariance functions have been selected to represent a diversity of structures: exponential, Gaussian, spherical, hyperbolic, k-Bessel, and cardinal sine. Their ranges were normalized to have similar integral values (Lantuéjoul 2002). For the two-dimensional simulations, the neighborhood sizes (4, 8, 12, 20, 32, 52, 108) were chosen such that the neighbors are homogeneously distributed around the simulated node.

5.1 Clustering Paths

5.1.1 Row-by-Row Path

For the sake of simplicity, the row-by-row path is first assessed on a one-dimensional grid. Thus, the neighborhoods are exclusively composed by of previously simulated nodes. This structure ensures a perfect propagation of the information of previously simulated nodes. Yet, because the neighborhood is only occupied by nodes whose distance to the simulated node is equal to or less than the neighborhood size, larger lag distances are never considered. Consequently, the reproduction of larger lag distances of the simulation covariance matrix can only be approximated based on covariances of shorter lags.

Let us consider a theoretical example with four evenly spaced nodes simulated with a row-by-row path and a neighborhood size of one node, that is, using only the previously simulated node. The variance is normalized to 1 and the covariance of lag h is denoted by $c_h = 1 - \gamma(h)$. Thus, $\mathbf{\Lambda}$ becomes

$$\mathbf{\Lambda} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{c_1}{1-c_1^2} & \frac{1}{1-c_1^2} & 0 & 0 \\ 0 & -\frac{c_1}{1-c_1^2} & \frac{1}{1-c_1^2} & 0 \\ 0 & 0 & -\frac{c_1}{1-c_1^2} & \frac{1}{1-c_1^2} \end{pmatrix}, \tag{9}$$

and, using Eq. (4), the simulation covariance matrix can be derived as

$$\mathbf{C} = \begin{pmatrix} 1 & c_1 & c_1^2 & c_1^3 \\ c_1 & 1 & c_1 & c_1^2 \\ c_1^2 & c_1 & 1 & c_1 \\ c_1^3 & c_1^2 & c_1 & 1 \end{pmatrix}^2. \tag{10}$$

In this example, the simulation exactly reproduces the model covariance for lag distances smaller than the neighborhood size, that is, c_1, but approximates the values at larger lags by a power-law-type extrapolation of the covariance at lag 1, such as, for example, $c_2 \approx c_1^2$. The overall error of the simulation is the mismatch measured

by $|c_1^h - c_h|$. This error will be different for each covariance function, depending on how well the covariance function can be approximated with this power law.

When the neighborhood size increases, the covariance of the simulation becomes more complicated to compute, but it always corresponds to a polynomial expression of the covariance at shorter lags. For example, with two neighbors, the covariance at lag 3 is approximated by

$$c_3 = \frac{c_1 + c_1 c_2^2 - 2c_1 c_2}{\left(c_1^2 - 1\right)^2}. \tag{11}$$

Using the symbolic toolbox in MATLAB, the mathematical expressions of a row-by-row simulation with a grid of 12 nodes were computed using neighborhood sizes between one and six nodes. Because the row-by-row path ensures that the neighborhood has the same configuration, the kriging weights are the same and the simulation covariance matrix has equal diagonal elements, that is, the covariance of all pairs of nodes with the same lag distance are identical (e.g., Eq. 10). Because of this property, no averaging is required in the computation of the covariance function (Fig. 4).

In all cases, the covariances of lags shorter than the neighborhood size are perfectly reproduced. However, each covariance function behaves differently for longer-range lags. The exponential covariance function exhibits no error because of its Markov property, which implies that a simulated node depends only on its immediate neighbors (Omre et al. 1993; Chilès and Delfiner 1999). The previously considered theoretical example confirms this property as the error $\|c_1^h - c_h\|$ vanishes for an exponential covariance function with $\left(e^{-1}\right)^h = e^{-h}$. This property has also been described as a perfect one-dimensional screening effect, where the first encountered node shields the influence of those behind it (Chilès and Delfiner 1999). Autoregressive processes generalized this effect to covariance models composed of damped exponentials and damped sine waves (Box et al. 2008; Chilès and Delfiner 1999). The unstable behavior associated with Gaussian and cardinal sine covariance functions can be explained by the difficulty of fitting a polynomial to their functions because of the presence of an inflection point. For the spherical, k-Bessel and hyperbolic covariance functions, the simulation covariance follows an exponential-type function fitted to the first n correct lag distance covariances.

The extension of these results to two dimensions is not trivial, because anisotropic effects occur. Figure 5 shows the two-dimensional covariance functions for a 64×64 grid with varying neighborhood size and covariance function. Because of the unilateral order of simulation, the bottom half of the two-dimensional covariance functions is a symmetric image of the upper half and thus is not shown.

In two dimensions, errors appear in the diagonal of the covariance function due to the direction of the path. For a Gaussian covariance function, the row-by-row path creates very large sinusoidal artifacts similar to the one-dimensional case. For the spherical covariance function, the sign of the error is similar to the one-dimensional case with an overestimation of the covariance for small neighborhoods and the largest errors in the vicinity of the range. For the exponential covariance function, the perfect screening

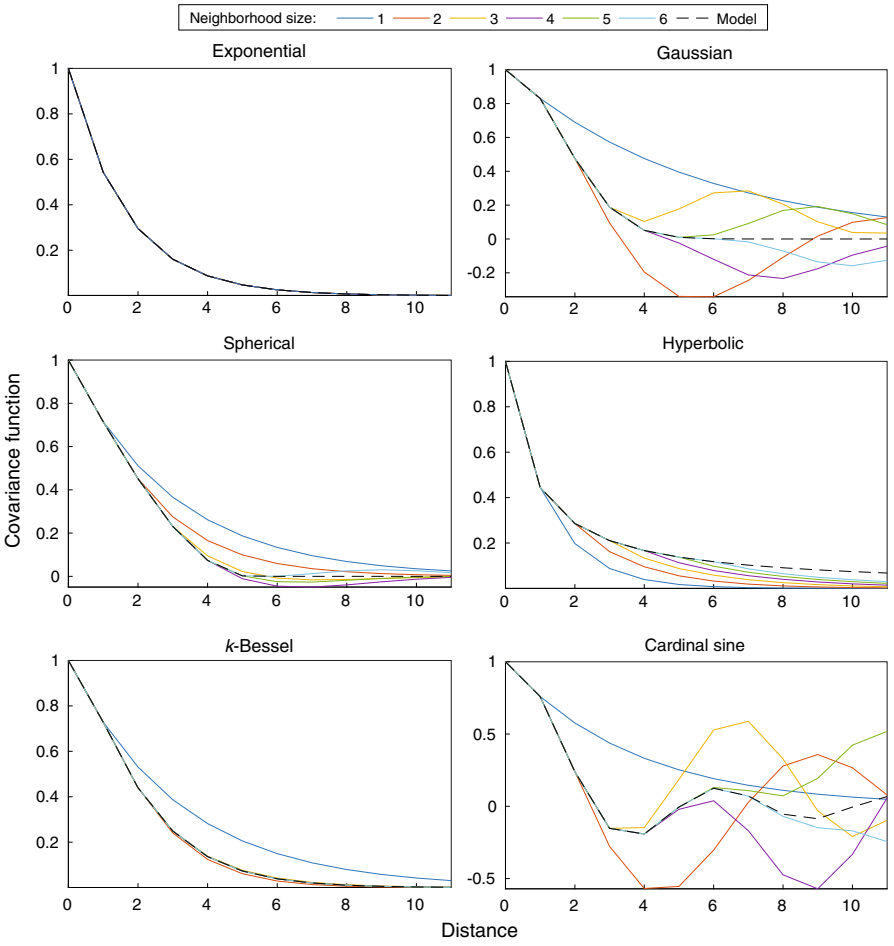

Fig. 4 Simulation covariance functions for a one-dimensional grid of 12 nodes with a row-by-row path for six different covariance functions and neighborhood sizes. The *dashed line* denotes the model covariance function

effect described in one dimension does not hold in two dimensions and, consequently, errors arise. Chilès and Delfiner (1999) showed that a covariance function that can be factorized along its components, such as the k-Bessel covariance function has a perfect two-dimensional screening effect. However, in order to generate a perfect simulation, all neighborhoods have to form a closed contour around the simulated node. The neighboring search strategy to build such a closed contour is complex and, hence has not been implemented here. This explains the small, but persistent errors encountered with the k-Bessel covariance function. Overall, exponential and k-Bessel covariance functions remain the most suitable models for the row-by-row path. The reader is referred to Boulanger (1990) for the extension of autoregressive processes in two dimensions.

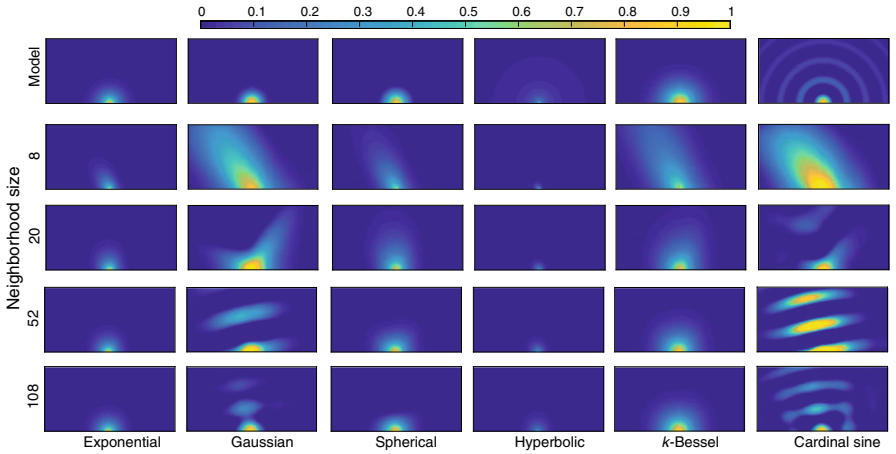

Fig. 5 Two-dimensional covariance functions for a grid of 64 × 64 nodes using a row-by-row path for different number of neighbors and covariance functions. Only the *upper half* of the covariance function map is displayed as the *bottom half* is a symmetric image of the *upper half*. The *first row* shows the model covariance function

5.1.2 Spiral Path

A common misconception associated with this path is its assumption that the most important nodes are the ones closest to the hard data. Thus, the rule that the most important area has to be simulated first (Gómez-Hernández and Journel 1993) causes the path to spiral away from hard data.

This path ensures a perfect reproduction of the influence of hard data on the close-by nodes because the hard data will always be included in their neighborhoods. However, as soon as there are too many nodes in the neighborhood, the hard data will be among the first to be eliminated in the kriging system, and its influence will be approximated by intermediate nodes similarly to the row-by-row path. As a result, the spiral path is particularly problematic, because it minimizes the presence of hard data in the neighborhood of simulated nodes.

To illustrate this, a 65 × 65 grid containing a single hard datum with a value of 1 centered in the middle of the grid was simulated using a spiral path. The expected value is symmetric around the hard datum, and thus a one-dimensional cross-section of the expected value is sufficient to illustrate adequately the resulting artifacts. Figure 6 shows the expected values $\mathrm{E}[\mathbf{z}^{(l)} \mid \mathbf{z}_0]$ along a one-dimensional section starting in the middle of the grid, such that the nodes are simulated from left to right. The expected values of the first simulated nodes match the kriging estimation perfectly, but once the hard datum is no longer included in the neighborhood, the expected value of the simulated nodes shows that the information of the hard datum is propagated by a polynomial that differs from the kriging estimator. These errors exhibit a similar structure as the covariance function errors generated by the row-by-row path. Indeed, the field of expected values of a simulation with a single hard datum with a value of one is exactly equal to the covariance function of the distance to the hard datum. Similar

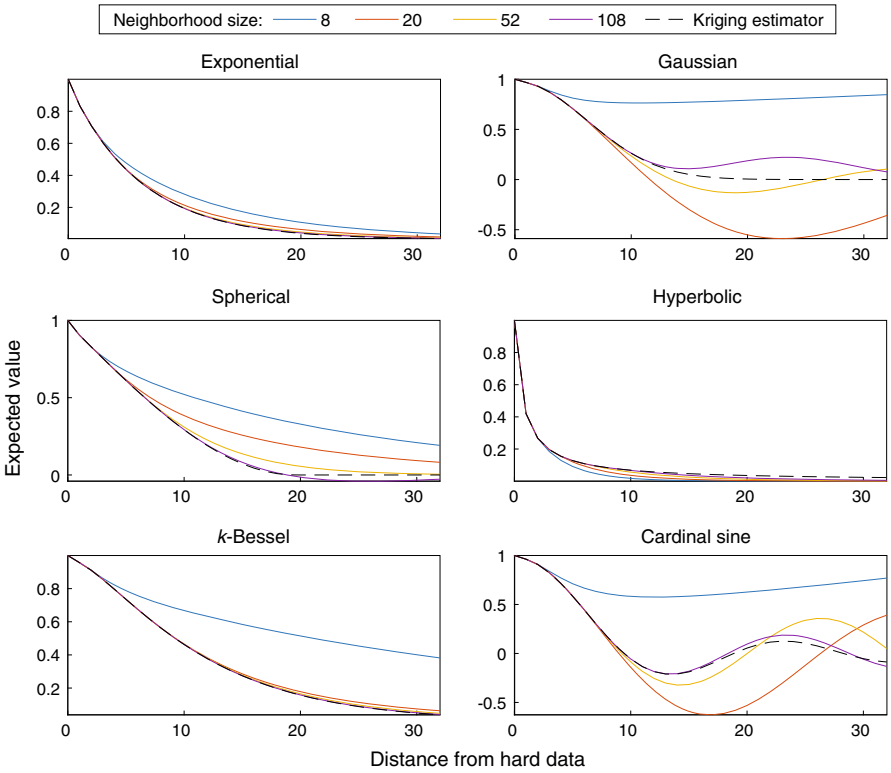

Fig. 6 Cross-section of the expected values starting from the center of the 65×65 grid where a single hard datum with value of 1 is present. The *black dotted line* is the kriging estimation and represents the target for the expected value of the simulation

to the row-by-row path, certain covariance functions are well approximated by this extrapolation (exponential, hyperbolic, k-Bessel), while others are poorly reproduced (Gaussian, spherical, cardinal sine).

When more conditioning data are available, the spiral path simulates the nodes in clusters around each hard datum. Figure 7 shows a section of the expected value of simulations with four hard data with different values and positions (black crosses). The random path is used as a benchmark.

This example demonstrates the problem of cluster merging associated with the spiral path. At first, each cluster evolves independently, propagating the information of each hard datum according to the covariance function approximation. As the clusters are growing, at a certain moment in the simulation, the neighborhood of one cluster will comprise a node of another cluster, which results in the merging of the two clusters. However, as this happens late in the simulation and because each cluster evolved independently, this merging is abrupt and hence creates large discontinuities in the realizations. Conversely, the random path leads to a better reproduction of the expected value.

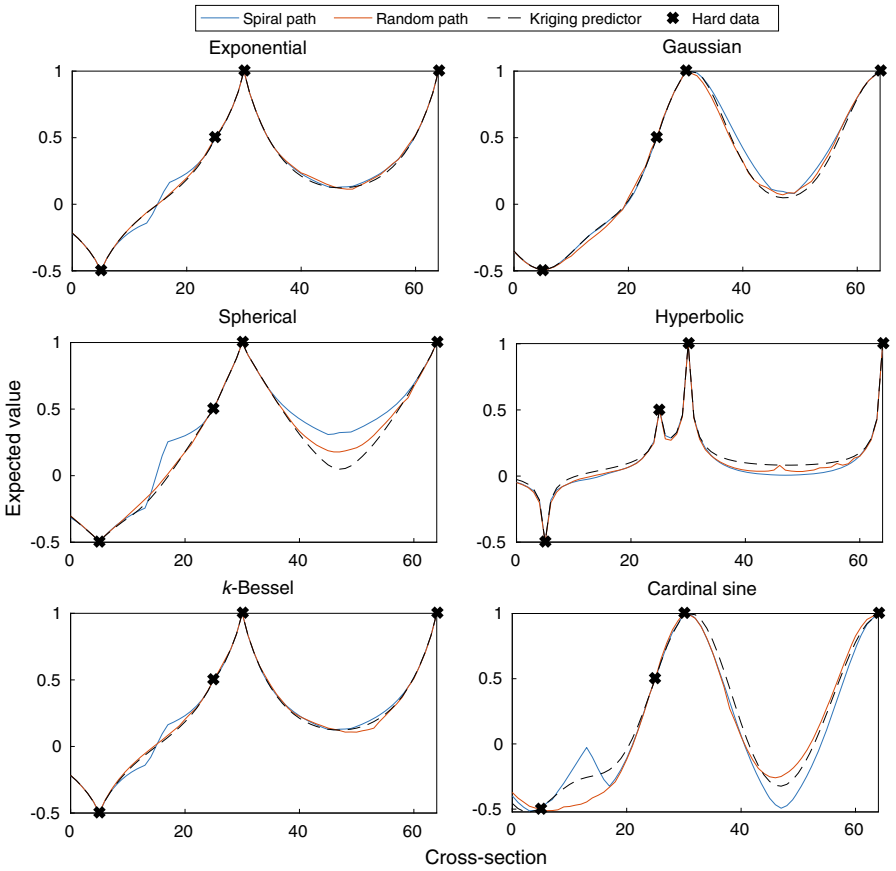

Fig. 7 Comparison of one-dimensional sections of the expected value of a simulation on a 65×65 grid with four hard data positioned along this section for spiral and random paths and different types of covariance functions

5.2 Declustering Paths

Unlike clustering paths, declustering paths favor the simulation of nodes that are far from their neighbors and thus avoid the creation of clusters. Since the three declustering paths, multi-grid, mid-point, quasi-random, are similar, they will be analyzed together.

Declustering paths allow for having neighborhoods composed of more spread-out nodes, which has two positive effects with regard to minimization of the error. First, inside the limited neighborhood, each selected node is representative of a different area of the grid, thus providing diverse and moderately correlated information. Secondly, because the correlations among the nodes inside and outside the neighborhood are weak, the removal of the nodes outside the neighborhood has little influence on the simulated node.

The error at the level of the node simulation can be measured with the relative screen effect approximation loss or RSEA (Dimitrakopoulos and Luo 2004), which corresponds to the normalized mean-square error difference of the simulated value

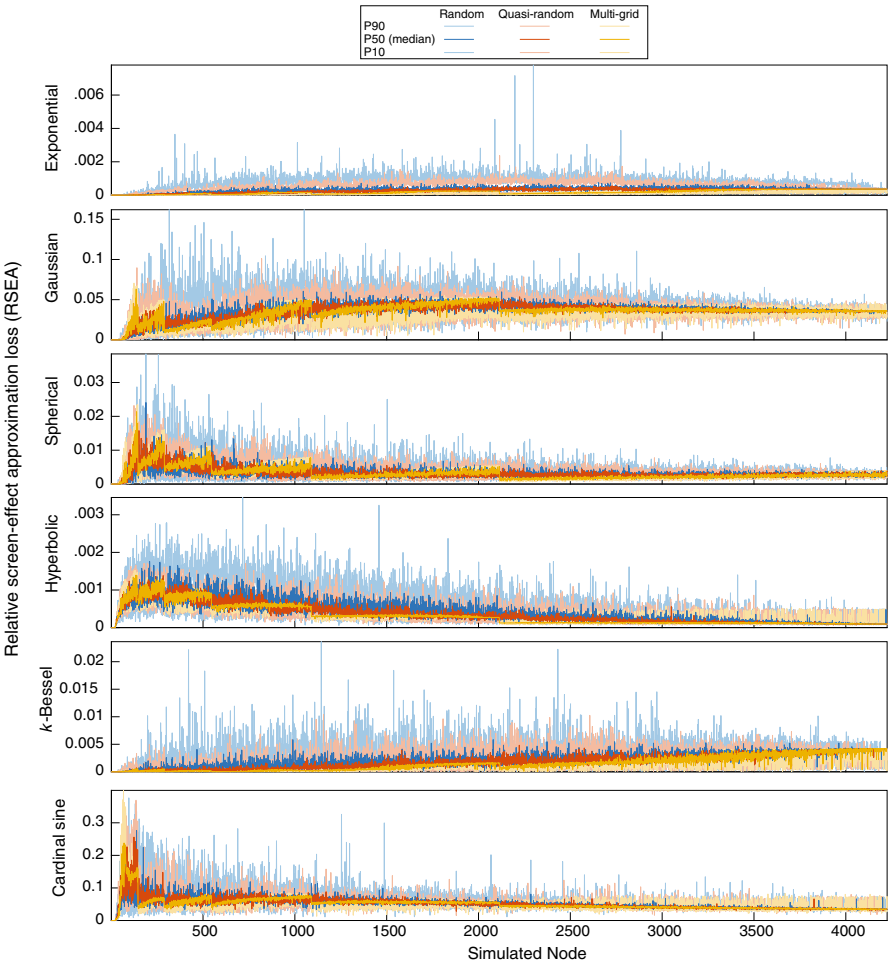

Fig. 8 10th, 50th and 90th percentiles of the RSEA (Eq. 12) for a random (*blue*), a quasi-random (*red*), and multi-grid (*yellow*) paths for various types of covariance models as a function of the simulation order. The results are based on 48 realizations, each of which has been carried out with a different randomized path

$$RSEA(\mathbf{u}_i) = \frac{1}{\mathrm{Var}\{Z^{(l)}(\mathbf{u}_i)\}} \frac{1}{2} \mathrm{E}\left\{\left[Z^{(l)}(\mathbf{u}_i) - Z(\mathbf{u}_i)\right]^2\right\} = 1 - \sqrt{\frac{\mathrm{Var}\{Z(\mathbf{u}_i)\}}{\mathrm{Var}\{Z^{(l)}(\mathbf{u}_i)\}}}, \tag{12}$$

where $\mathrm{Var}\{Z(\mathbf{u}_i)\}$ and $\mathrm{Var}\{Z^{(l)}(\mathbf{u}_i)\}$ denote the kriging variance error σ_E^2 for a full and limited neighborhood, respectively. Readers are referred the corresponding paper for the demonstration.

Figure 8 compares the RSEA according to the simulation order of three path types for each of the six covariance functions considered in this study. For each path type

and covariance model, 48 realizations were performed with a different randomized path. The percentiles P10, P50 (median), and P90 of the RSEA values for the nodes simulated at a given order of simulation are shown on Fig. 8. For all covariance functions, the very first nodes are simulated without any error because all neighbors are included. Then, all models behave differently in response to their various covariance function characteristics. The median value suggests that quasi-random and multi-grid paths perform only slightly better than the random path. However, the P90 shows that quasi-random and especially multi-grid paths minimize large RSEA values. Indeed, the main advantage of these paths is to avoid neighborhood configurations leading to large errors.

The error over the whole simulation is assessed by the mismatch between the simulation covariance matrix and the model covariance matrix. Following an approach suggested by Emery and Peláez (2011), the error matrix can be aggregated through the standardized Frobenius norm, which corresponds to a normalized root mean square error of the form

$$\eta = \frac{\|\mathbf{C}_{Z^{(l)}} - \mathbf{C}_Z\|}{\|\mathbf{C}_Z\|}, \tag{13}$$

where $\|\cdot\|$ denotes the Frobenius norm, also referred to as $L_{2,2}$.

Figure 9 compares the standardized Frobenius norm of simulations using random and declustering paths for different types of covariance functions and neighborhood sizes. Varying grid sizes and correlation ranges were also tested, but because these parameters were not found to have a significant influence, only simulations for a 64×64 grid with a correlation range of 15 nodes are shown. The standardized Frobenius norm value is computed for 48 realizations for each path type and for each neighborhood size. Please note that each realization uses a different randomized path, either of the multi-grid, quasi-random, or random path type. Figure 9 shows the averages of the 48 realizations, as well as the their standard deviations as errorbars. These results demonstrate that the variation of the standardized Frobenius norm among realizations is much smaller compared to simulations with different path types. For all covariance functions, a significant error reduction is observed for quasi-random and multi-grid paths compared to random path. In a logarithmic scale, this error reduction remains constant with an increased neighborhood size. Figure 9 also illustrates the major influence of increasing the neighborhood size for reducing the error.

Figure 10 shows the covariance function errors for simulations with 20 neighbors and a range of 20. Because the covariance matrix contains several pairs of nodes with the same lag distance, the distribution of covariance errors is displayed for each lag distance. Each covariance function produces a different error structure, but these structures are not dependent on the path type. The neighborhood size has an important influence on the shape of these error structures. Declustering paths are associated with a major error reduction for lags between half to twice the range. This result confirms that multi-grid paths, and declustering paths in general, improve the covariance function reproduction for large lag distances.

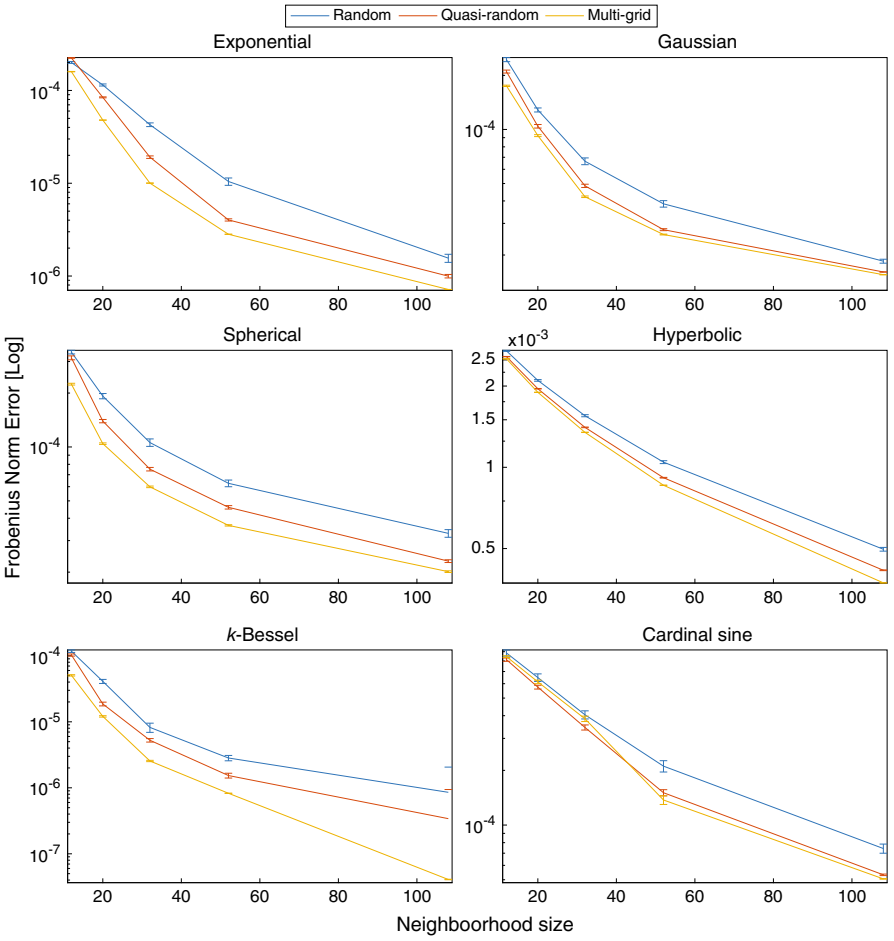

Fig. 9 Average standardized Frobenius norm for simulations using random (*blue*), quasi-random (*red*), and multi-grid (*yellow*) paths for different covariance functions and neighborhood sizes. Forty-eight realizations with a different randomized path were computed for each type of path and neighborhood size. The corresponding standard deviations of the standardized Frobenius norm error are displayed as *errorbars*

6 Discussion

6.1 Conditional Simulations

Conditional simulations can be viewed as simulations where the paths start with the known nodes being perfectly simulated. When only few hard data are available, a multi-grid path is optimal as the simulation of the remaining nodes will be only mildly affected by the hard data. However, if many hard data are present, their positions will affect the optimal path at the simulation level and hence a multi-grid path might no longer be the best choice. Yet, declustering of the nodes remains the overall approach to follow, and in such a context, the mid-node path is a good alternative. It differs

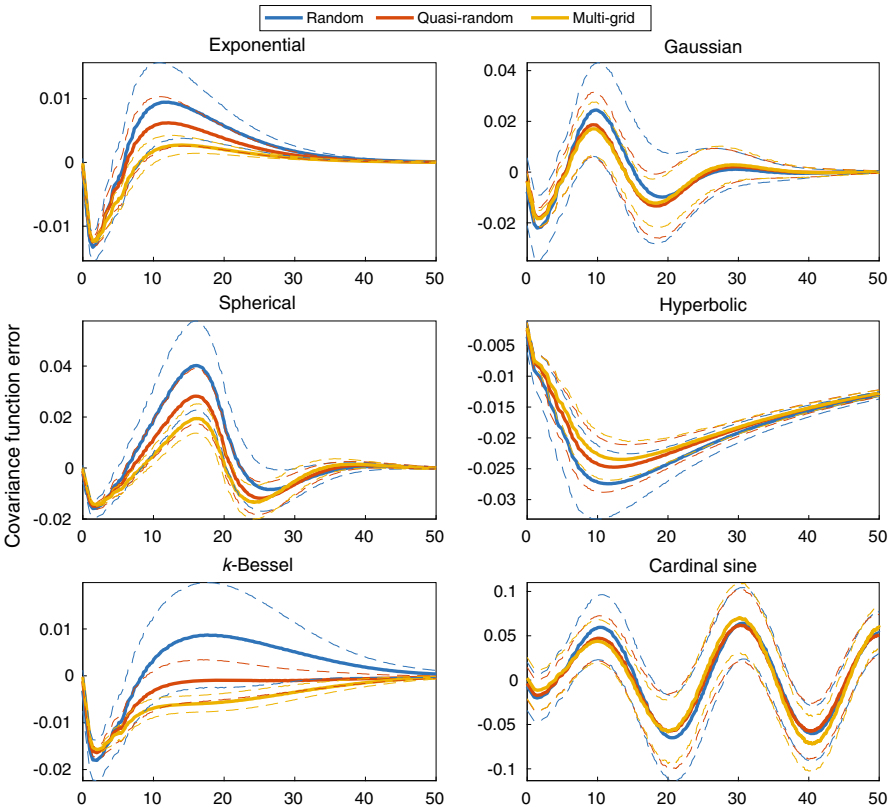

Fig. 10 Covariance function error for simulations using different types of paths and covariance functions. *Solid lines* denote the mean values and *dotted lines* the one standard deviation interval resulting from the aggregation of the covariance matrix with equal lag distance

from the multi-grid approach, in that it iteratively selects the node that has the largest distance to its closest neighbor, thus taking into account the positions of the hard data.

It is common practice to attribute high confidence to hard data and request that their influence is correctly propagated throughout the simulation. Although this objective is hard to address through the choice of the type of path, the neighborhood search method can be adapted to achieve it. Indeed, as demonstrated by Emery and Peláez (2011), if all hard data are imposed in the neighborhoods of all simulated nodes, the expected value of the simulated nodes is perfectly reproduced over the simulated domain. This type of approach can be implemented with a two-part search strategy, which sorts and selects hard data separately from previously simulated nodes (Deutsch and Journel 1992).

6.2 Search Method

Throughout this study, the selection of nodes was based only on their distance to the simulated node relative to correlation range and orientation. The multiple and interact-

ing effects described in Sect. 4 suggest that distance is not always a sufficient criterion for determining the optimal neighborhood, as for example with the relay effect illustrated in Fig. 3e. Generally, this situation only arises when more diverse information comes from a distant node. Fortunately, this scenario can be avoided through the use of declustering paths. Yet, in the presence of clustered hard data, this effect cannot be ignored, and the search method should be suitably adapted. The octant/quadrant search method (Isaaks and Srivastava 1989) has been designed to balance the directional sampling of the neighborhood by separately searching each quadrant of the domain. This is a useful approach in the presence of clusters of hard data, while for unconditional simulations with a declustering path our simulations showed significant errors. This is consistent with the findings of Goovaerts (1997) and Deutsch and Journel (1992) who report that octant/quadrant search tends to create discontinuities in the absence of clusters, because of the drastic change of neighborhood from one node to the next.

7 Conclusion

In this study, a systematic analysis was presented to explore the link between the simulation path and artifacts in SGS realizations. The common practice of limiting the number of neighbors in the kriging estimation results in biases. In this respect, the role of the simulation path is to determine which nodes are included or omitted in the neighborhood, and thus it ultimately decides which pairs of nodes will be incorrectly correlated and how large the associated errors will be. Exact simulation covariance matrices are computed to assess rigorously the relationship between biases in the simulations and parameters such as grid size, neighborhood size, covariance function and range, and, most importantly, the different types of paths. This assessment of different simulation paths results in a classification into clustering paths, which simulate consecutively nearby nodes, and declustering paths, which maximize the distance between consecutively simulated nodes.

Clustering paths, such as the row-by-row and spiral paths, perfectly reproduce the covariances at short lag distances, but result in large, and often inadequate, approximations for larger lag covariances. This is due to the inherent structure of these paths, which results in the kriging neighborhood being exclusively constituted by nearby nodes. For the same reason, the spiral path poorly propagates the influence of hard data. Conversely, declustering paths, such as multi-grid, mid-point, or quasi-random paths, maintain a balanced and diverse neighborhood while minimizing the correlation between nodes within the neighborhood and beyond, thus resulting in minimum biases. Their overall path structure also minimizes cumulative bias by simulating less correlated nodes first, thus avoiding the re-use of nodes associated with large biases.

Acknowledgements This study has been supported by a Grant from the Swiss National Research Foundation.

References

Abdu H, Robinson DA, Seyfried M, Jones SB (2008) Geophysical imaging of watershed subsurface patterns and prediction of soil texture and water holding capacity. Water Resour Res 44(4):W00D18. doi:10.1029/2008WR007043

Barnsley MF, Devaney RL, Mandelbrot BB, Peitgen HO, Saupe D, Voss RF (1988) The science of fractal images. Springer, New York. doi:10.1007/978-1-4612-3784-6
Boulanger F (1990) Modélisation et simulation de variables régionalisées par des fonctions aléatoires stables. Ph.D. thesis, Ecole des Mines de Paris, Fontainebleau
Box GEP, Jenkins GM, Reinsel GC (2008) Time series analysis, vol 37. Wiley, Hoboken. doi:10.1002/9781118619193
Chilès JP, Delfiner P (1999) Geostatistics, Wiley series in probability and statistics, vol 497. Wiley, Hoboken. doi:10.1002/9780470316993
Daly C (2005) Higher order models using entropy, Markov random fields and sequential simulation. In: Leuangthong O, Deutsch CV (eds) Geostatistics Banff 2004. Quantitative Geology and Geostatistics, vol 14. Springer, Dordrecht
Day-Lewis FD, Lane JW (2004) Assessing the resolution-dependent utility of tomograms for geostatistics. Geophys Res Lett 31(7):L07,503. doi:10.1029/2004GL019617
Delbari M, Afrasiab P, Loiskandl W (2009) Using sequential Gaussian simulation to assess the field-scale spatial uncertainty of soil water content. Catena 79(2):163–169. doi:10.1016/j.catena.2009.08.001
Deutsch CV, Journel AG (1992) GSLIB: Geostatistical software library and user's guide. Technical Representative, New York
Dimitrakopoulos R, Luo X (2004) Generalized sequential Gaussian simulation on group size and screen-effect approximations for large field simulations. Math Geol 36(5):567–591. doi:10.1023/B:MATG.0000037737.11615.df
Dimitrakopoulos R, Farrelly CT, Godoy M (2002) Moving forward from traditional optimization: grade uncertainty and risk effects in open-pit design. Min Technol 111(1):82–88. doi:10.1179/mnt.2002.111.1.82
Emery X (2004) Testing the correctness of the sequential algorithm for simulating Gaussian random fields. Stoch Env Res Risk Assess 18(6):401–413. doi:10.1007/s00477-004-0211-7
Emery X, Peláez M (2011) Assessing the accuracy of sequential Gaussian simulation and cosimulation. Comput Geosci 15(4):673–689. doi:10.1007/s10596-011-9235-5
Fournier A, Fussell D, Carpenter L (1982) Computer rendering of stochastic models. Commun ACM 25(6):371–384. doi:10.1145/358523.358553
Gómez-Hernández JJ, Cassiraga EF (1994) Theory and practice of sequential simulation. Kluwer Academic Publishers, Dordrecht. doi:10.1007/978-94-015-8267-4_10
Gómez-Hernández JJ, Journel AG (1993) Geostatistics Tróia '92, quantitative geology and geostatistics, vol 5. Springer, Dordrecht. doi:10.1007/978-94-011-1739-5
Goovaerts P (1997) Geostatistics for natural resources evaluation. Oxford University Press, Oxford
Goovaerts P (2001) Geostatistical modelling of uncertainty in soil science. Geoderma 103(1–2):3–26. doi:10.1016/S0016-7061(01)00067-2
Halton JH (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer Math 2(1):84–90. doi:10.1007/BF01386213
Hansen TM, Journel AG, Tarantola A, Mosegaard K (2006) Linear inverse Gaussian theory and geostatistics. Geophysics 71(6):R101–R111. doi:10.1190/1.2345195
Isaaks EH (1991) The application of Monte Carlo methods to the analysis of spatially correlated data. Ph.D. thesis, Stanford University
Isaaks EH, Srivastava RM (1989) An introduction to applied geostatistics. Oxford University Press, New York
Johnson ME (1987) Multivariate statistical simulation. Wiley series in probability and statistics. Wiley, Hoboken. doi:10.1002/9781118150740
Journel AG (1989) Fundamentals of geostatistics in five lessons, vol 16. American Geophysical Union, Washington. doi:10.1029/SC008
Juang KW, Chen YS, Lee DY (2004) Using sequential indicator simulation to assess the uncertainty of delineating heavy-metal contaminated soils. Environ Pollut 127(2):229–238. doi:10.1016/j.envpol.2003.07.001
Kocis L, Whiten WJ (1997) Computational investigations of low-discrepancy sequences. ACM Trans Math Softw 23(2):266–294. doi:10.1145/264029.264064
Lantuéjoul C (2002) Geostatistical simulation. Springer, Berlin. doi:10.1007/978-3-662-04808-5
Lee SY, Carle SF, Fogg GE (2007) Geologic heterogeneity and a comparison of two geostatistical models: sequential Gaussian and transition probability-based geostatistical simulation. Adv Water Resour 30(9):1914–1932. doi:10.1016/j.advwatres.2007.03.005

Leuangthong O, McLennan JA, Deutsch CV (2004) Minimum acceptance criteria for geostatistical realizations. Nat Resour Res 13(3):131–141. doi:10.1023/B:NARR.0000046916.91703.bb

Lin YP, Chang TK, Teng TP (2001) Characterization of soil lead by comparing sequential Gaussian simulation, simulated annealing simulation and kriging methods. Environ Geol 41(1–2):189–199. doi:10.1007/s002540100382

McLennan J (2002) The effect of the simulation path in sequential gaussian simulation. Technical Representative, University of Alberta

Meyer TH (2004) The discontinuous nature of kriging interpolation for digital terrain modeling. Cartogr Geogr Inf Sci 31(4):209–216. doi:10.1559/1523040042742385

Mowrer H (1997) Propagating uncertainty through spatial estimation processes for old-growth subalpine forests using sequential Gaussian simulation in GIS. Ecol Model 98(1):73–86. doi:10.1016/S0304-3800(96)01938-2

Omre H, Sølna K, Tjelmeland H (1993) Simulation of random functions on large lattices. In: Soares A (ed) Geostatistics Tròia '92. Kluwer Academic Publishers, Dordrecht, pp 179–199. doi:10.1007/978-94-011-1739-5_16

Rivoirard J (1984) Le comportement des poids de krigeage. Ph.D. thesis, Ecole des Mines de Paris, Fontainebleau

Safikhani M, Asghari O, Emery X (2017) Assessing the accuracy of sequential gaussian simulation through statistical testing. Stoch Env Res Risk Assess 31(2):523–533. doi:10.1007/s00477-016-1255-1

Srinivasan BV, Duraiswami R, Murtugudde R (2008) Efficient kriging for real-time spatio-temporal interpolation Linear kriging. In: 20th conference on probablility and statistics in atmospheric sciences, pp 228–235

Tran TT (1994) Improving variogram reproduction on dense simulation grids. Comput Geosci 20(7–8):1161–1168. doi:10.1016/0098-3004(94)90069-8

Trefethen LN, Bau D III (1997) Numerical linear algebra, vol 50. SIAM, Philadelphia

Verly GW (1993) Sequential Gaussian cosimulation: a simulation method integrating several types of information. In: Soares A (ed) Geostatistics Tròia '92. Kluwer Academic Publishers, Dordrecht, pp 543–554. doi:10.1007/978-94-011-1739-5_42

Zhao Y, Xu X, Huang B, Sun W, Shao X, Shi X, Ruan X (2007) Using robust kriging and sequential Gaussian simulation to delineate the copper- and lead-contaminated areas of a rapidly industrialized city in Yangtze River Delta, China. Environ Geol 52(7):1423–1433. doi:10.1007/s00254-007-0667-0