Direct sampling method to perform multiple-point geostatistical simulations

Gregoire Mariethoz,[1,2] Philippe Renard,[1] and Julien Straubhaar[1]

Received 9 December 2008; revised 13 July 2010; accepted 6 August 2010; published XX Month 2010.

[1] Multiple-point geostatistics is a general statistical framework to model spatial fields displaying a wide range of complex structures. In particular, it allows controlling connectivity patterns that have a critical importance for groundwater flow and transport problems. This approach involves considering data events (spatial arrangements of values) derived from a training image (TI). All data events found in the TI are usually stored in a database, which is used to retrieve conditional probabilities for the simulation. Instead, we propose to sample directly the training image for a given data event, making the database unnecessary. Our method is statistically equivalent to previous implementations, but in addition it allows extending the application of multiple-point geostatistics to continuous variables and to multivariate problems. The method can be used for the simulation of geological heterogeneity, accounting or not for indirect observations such as geophysics. We show its applicability in the presence of complex features, nonlinear relationships between variables, and with various cases of nonstationarity. Computationally, it is fast, easy to parallelize, parsimonious in memory needs, and straightforward to implement.

Citation: Mariethoz, G., P. Renard, and J. Straubhaar (2010), Direct sampling method to perform multiple-point geostatistical simulations, *Water Resour. Res.*, *46*, XXXXXX, doi:10.1029/2008WR007621.

1. Introduction

[2] Geological heterogeneity has a critical influence on groundwater flow and related processes such as solute transport or rock-water interactions. Consequently, a broad range of models of heterogeneity have been developed over the last 50 years to improve the understanding of groundwater-related processes in complex media [*Dagan*, 1976, 1986; *De Marsily et al.*, 2005; *Freeze*, 1975; *Koltermann and Gorelick*, 1996; *Matheron*, 1966, 1967; *Sanchez-Vila et al.*, 2006]. These models are used on the one hand to investigate the influence of heterogeneity on the processes, see for example *Rubin* [2003] or *Zhang* [2002] for recent and detailed synthesis of the most important results. On the other hand, even if the stochastic models of heterogeneity are not used as much as they could be in practice [*Dagan*, 2004; *Renard*, 2007], they make it possible to quantify the uncertainty related to the lack of data and therefore constitute a base for rationale water management under uncertainty [*Alcolea et al.*, 2009; *Feyen and Gorelick*, 2004; *Freeze et al.*, 1990]. Within this general framework, the most standard mathematical model of heterogeneity is the multi-Gaussian model [*Dagan*, 1989; *Gelhar*, 1993; *Rubin*, 2003; *Zhang*, 2002]. However, alternative methods are used when one is interested in specific connectivity patterns [*Capilla and Llopis-Albert*, 2009; *Emery*, 2007; *Gómez-Hernández and Wen*, 1998; *Kerrou et al.*, 2008; *Klise et al.*, 2009; *Knudby and Carrera*, 2005; *Neuwiler and Cirpka*, 2005; *Sánchez-Vila et al.*, 1996; *Schaap et al.*, 2008; *Wen and Gomez-Hernandez*, 1998; *Western et al.*, 2001; *Zinn and Harvey*, 2003]. This motivated the development of a large number of modeling techniques [*De Marsily et al.*, 2005; *Koltermann and Gorelick*, 1996]. Among them, multiple-point statistics [*Guardiano and Srivastava*, 1993] is very promising as discussed in the recent review by *Hu and Chugunova* [2008]. One of the most efficient and popular implementations of that theory is the *snesim* algorithm [*Strebelle*, 2002]. This method is now increasingly used in the oil industry [*Aitokhuehi and Durlofsky*, 2005; *Caers et al.*, 2003; *Hoffman and Caers*, 2007; *Liu et al.*, 2004; *Strebelle et al.*, 2003] and in hydrogeology [*Chugunova and Hu*, 2008; *Feyen and Caers*, 2006; *Huysmans and Dassargues*, 2009; *Michael et al.*, 2010; *Renard*, 2007]. It has also been applied with inverse modeling techniques [*Alcolea and Renard*, 2010; *Caers and Hoffman*, 2006; *Ronayne et al.*, 2008; G. Mariethoz et al., Bayesian inverse problem and optimization with Iterative Spatial Resampling, submitted to *Water Resources Research*, 2010]. Although the method is gaining popularity, it still suffers from several shortcomings. Some of the most acute ones are the difficulties involved in simulating continuous variables and performing cosimulations, as well as the computational burden involved.

[3] In this paper, we propose an alternative multiple-point simulation technique (Direct Sampling) that can deal both with categorical data, such as rock types, and continuous variables, such as permeability, porosity, or geophysical attributes, and can also handle cosimulations. The primary use of the direct sampling method in hydrogeology is the simulation of geological heterogeneity. Its main advantages are simplicity and flexibility. The approach allows for the construction of models presenting a wide variety of connectivity patterns. Furthermore, nonstationarity is a very frequent

[1]Centre for Hydrogeology, University of Neuchâtel, Neuchâtel, Switzerland.
[2]ERE Department, Stanford University, Stanford, California, USA.

0043-1397/10/2008WR007621

feature in most real case situations. Therefore, a special effort has been devoted to developing a set of techniques that can be applied when nonstationarity occurs. Because the method can handle cosimulation between categorical and continuous variable when the relation between the variables is complex, it allows for the integration of geophysical measurements and categorical rock types observations in the model. Furthermore, even though the method has been developed with the aim to improve the characterization of heterogeneous aquifers, it is general and can be applied to other fields of water resources such as rainfall simulation, integration of remote sensing data, and flood forecasting. These last aspects will not be treated in the present paper. Instead, we will focus only on the presentation of the direct sampling method and its use for heterogeneity modeling. The first section of the paper provides an overview of multiple-point geostatistics and highlights the novel aspects of the direct sampling method (DS). Section 2 is a detailed description of the DS algorithm. The following sections illustrate the possibilities offered by the method, such as simulating continuous properties, addressing multivariate problems, and dealing with nonstationarity. Finally, the last section discusses a recursive syn-processing method that is an improvement from existing postprocessing algorithms [*Stien et al.*, 2007; *Strebelle and Remy*, 2005; *Suzuki and Strebelle*, 2007]. It offers a way of controlling the trade-off between numerical efficiency and quality of the simulation. The syn-processing is applied in conjunction with DS but could be used with any other multiple-point simulation algorithm.

2. Background on Multiple-Point Geostatistics

[4] Multiple-point geostatistics is based on three conceptual changes that were formalized by *Guardiano and Srivastava* [1993]. The first one is to state that data sets may not be sufficient to infer all the statistical features that control what the modeler is interested in. For example, on the basis only of the on point data, it is not possible to know whether the high values of hydraulic conductivity are connected or belong to isolated blocks [*Gómez-Hernández and Wen*, 1998]. Therefore, any statistical inference based only on the analysis of point data (even if it uses complex statistics) will be blind to that characteristic of the underlying field [*Sánchez-Vila et al.*, 1996; *Zinn and Harvey*, 2003].

[5] The second conceptual change is to adopt a non-parametric statistical framework to represent heterogeneity [*Journel*, 1983; *Wasserman*, 2006]. The proposal of *Guardiano and Srivastava* [1993] is to use a training image (TI), i.e., a grid containing spatial patterns deemed representative of the spatial structures to simulate. The training image can be viewed as a conceptual model of the heterogeneity in the case of aquifer characterization but should be seen more generally as an explicit prior model [*Journel and Zhang*, 2006]. The statistical model is then based not on the data only but also on the choice of the training image and on the algorithm and parameters that control its behavior [*Boucher*, 2007]. One can choose training images that reflect various spatial models [*Suzuki and Caers*, 2008] and that integrate external information about spatial variability, such as geological knowledge not contained in the data itself. This is especially useful in cases where data are too scarce for the inference of a spatial model. Conversely, when large amounts of hard data are present, it is possible to abandon the TI and to adopt an entirely data-driven approach by inferring multiple-point statistics from these data [*Mariethoz and Renard*, 2010; *Wu et al.*, 2008].

[6] The use of a TI makes the third conceptual change possible, which is to evaluate the statistics of multiple-point data events [*Guardiano and Srivastava*, 1993]. The multiple-point statistics are expressed as the cumulative density functions for the random variable $Z(\mathbf{x})$ conditioned to local data events $\mathbf{d}_n = \{Z(\mathbf{x}_1), Z(\mathbf{x}_2), \cdots, Z(\mathbf{x}_n)\}$, i.e., the values of Z in the neighboring nodes $\mathbf{x}_i$ of $\mathbf{x}$,

$$\mathrm{F}(z, \mathbf{x}, \mathbf{d}_n) = \mathrm{Prob}\{Z(\mathbf{x}) \leq z | \mathbf{d}_n\}. \quad (1)$$

Simulations based on multiple-point statistics proceed sequentially. At each successive location, the conditional cumulative distribution function (ccdf) $\mathrm{F}(z, \mathbf{x}, \mathbf{d}_n)$ is conditioned to both the previously simulated nodes and the actual data. A value for $\mathrm{Z}(\mathbf{x})$ is drawn from the probability distribution and the algorithm proceeds to the next location. Since $\mathrm{F}(z, \mathbf{x}, \mathbf{d}_n)$ depends on the respective values and relative positions of all the neighbors of $\mathbf{x}$ simultaneously, it is very rich in terms of information content. To estimate the nonparametric ccdf (1) at each location, *Guardiano and Srivastava* [1993] proposed to scan entirely the training image at each step of the simulation. The method was inefficient and therefore could not be used in practice.

[7] A solution to that problem was developed by *Strebelle* [2002]: the *snesim* simulation method proceeds by scanning the training image for all pixel configurations of a certain size (the template size) and storing their statistics in a catalogue of data events having a tree structure before starting the sequential simulation process. The tree structure is then used to rapidly compute the conditional probabilities at each simulated node. In general, to limit the size of the tree in memory, the template size is kept small, which prevents capturing large-scale features such as channels. To palliate this problem, *Strebelle* [2002] introduced multigrids (or multiscale grids) to simulate the large-scale structures first and later the small-scale features. Although multigrids allow good reproduction at different scales, they generate problems related to the migration of conditioning data at each multigrid level. Artifacts may appear, especially with large data sets that cannot be fully used on the coarsest multigrids levels. Since all configurations of pixel values that are found in the TI are stored in the search tree, the use of *snesim* is often limited by the memory usage. The size of the template, the number of lithofacies, and the degree of entropy of the training image directly control the size of the search tree and therefore control the memory requirement for the algorithm. In practice, these parameters are limited by the available memory especially for large 3-D grids. For example, with four lithofacies and a template made of 30 nodes, there can be up to 4^{30} possible data events, which by far exceeds the memory limit of any present-day computer (although in practice, the number of data events is limited by the size of the TI). This imposes limits on the number of facies and the template size, and hence complex structures described in the TI can often not be properly reproduced. J. Straubhaar et al. (An improved parallel multiple-point algorithm, submitted to *Mathematical Geosciences*, 2010) mitigate this problem by storing multiple-point statistics in lists instead of tree struc-

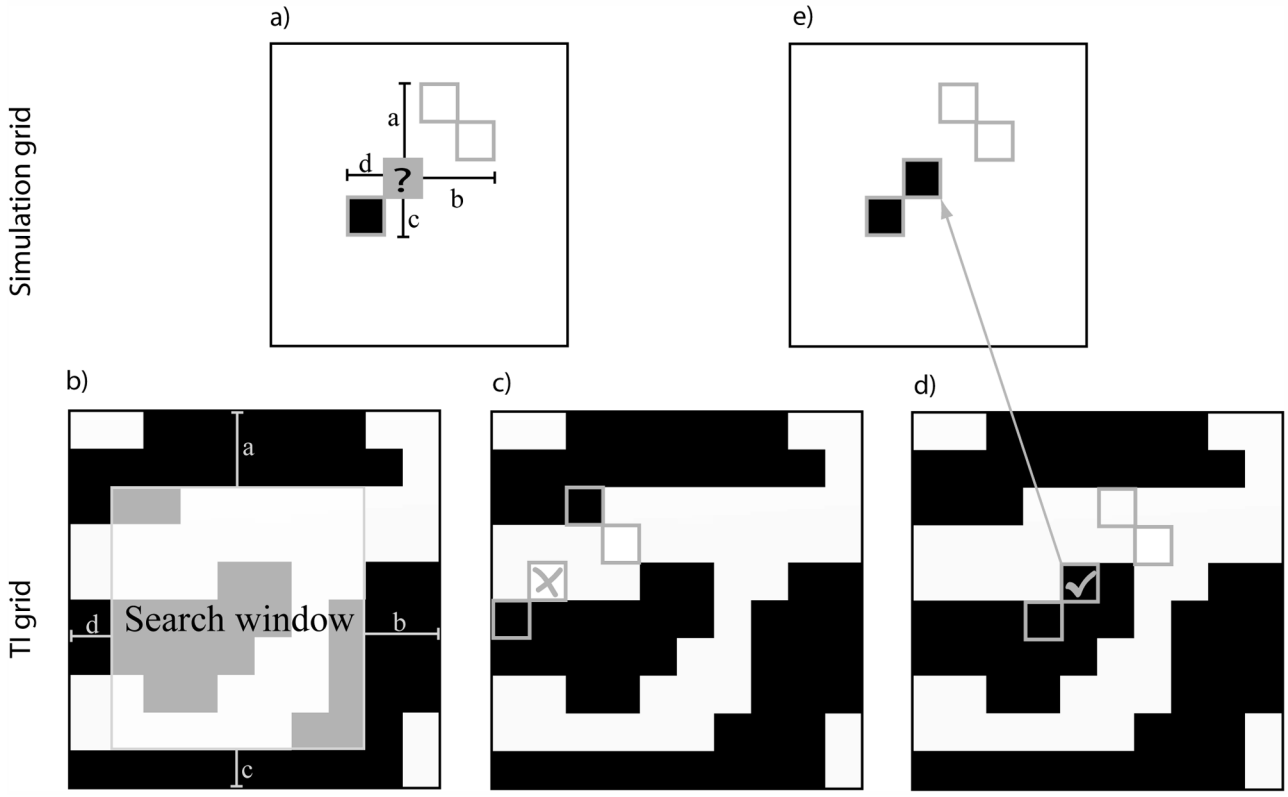

Figure 1. Illustration of the direct sampling (DS) method. (a) Define the data event in the simulation grid. The question mark represents the node to be simulated. The two white and the black pixels represent nodes that have been previously simulated. (b) Define a search window in the TI grid by using the dimensions a, b, c, d of the data event. (c) Linearly scan the search window starting from a random location until (d) the simulation data event is satisfactorily matched. (e) Assign the value of the central node of the first matching data event to the simulated node.

tures. In addition, to account for nonstationarity either in the training image or in the simulation, it is necessary to include additional variables that further increase the demand for memory storage [*Chugunova and Hu*, 2008].

[8] The approaches described in the previous paragraphs can only deal with categorical variables because of the difficulty to infer (1) from a continuous TI. *Zhang et al.* [2006] propose an alternative method in which the patterns are projected (through the use of filter scores) into a smaller dimensional space in which the statistical analysis can be carried out. The resulting *filtersim* algorithm does not simulate nodes one by one sequentially, but proceeds by pasting groups of pixels (patches) into the simulation grid. It uses the concept of similarity measure between groups of pixels and can be applied both to continuous or categorical variables. For completeness, it should be noted that *Arpat and Caers* [2007] and *El Ouassini et al.* [2008] also proposed alternative techniques based on pasting entire patterns.

[9] In this paper, we adopt the point of view that generating simulations satisfying the ccdf expressed in equation (1) does not involve explicitly computing this ccdf. We therefore suggest that the technical difficulties involved in the computation of the ccdf can be avoided. Instead of storing and counting the configurations found in the training image, it is more convenient to directly sample the training image in a random manner but conditional to the data event. Mathematically, this is equivalent to using the training image (TI) to compute the ccdf and then drawing a sample from it. The resulting direct sampling (DS) algorithm is inspired by *Shannon* [1948], who produced Markovian sequences of random English by drawing letters from a book conditionally to previous occurrences.

[10] In addition, we use a distance (mismatch) between the data event observed in the simulation and the one sampled from the TI. During the sampling process, if a pattern is found that matches exactly the conditioning data or if the distance between these two events is lower than a given threshold, the sampling process is stopped and the value at the central node of the data event in the TI is directly pasted in the simulation. Choosing an appropriate measure of distance makes it possible to deal with either categorical or continuous variables and to accommodate complex multivariate problems such as relationships between categorical and continuous variables.

3. Direct Sampling Algorithm

[11] The aim of the direct sampling method is to simulate a random function $Z(\mathbf{x})$. The input data are a simulation grid (SG) whose nodes are denoted $\mathbf{x}$, a training image (TI) whose nodes are denoted $\mathbf{y}$, and, if available, a set of N conditioning data $z(\mathbf{x}_i)$, $i \in [1, \cdots, N]$ such as borehole observations. The principle of the simulation algorithm is illustrated in Figures 1 and 2 and proceeds as follows.

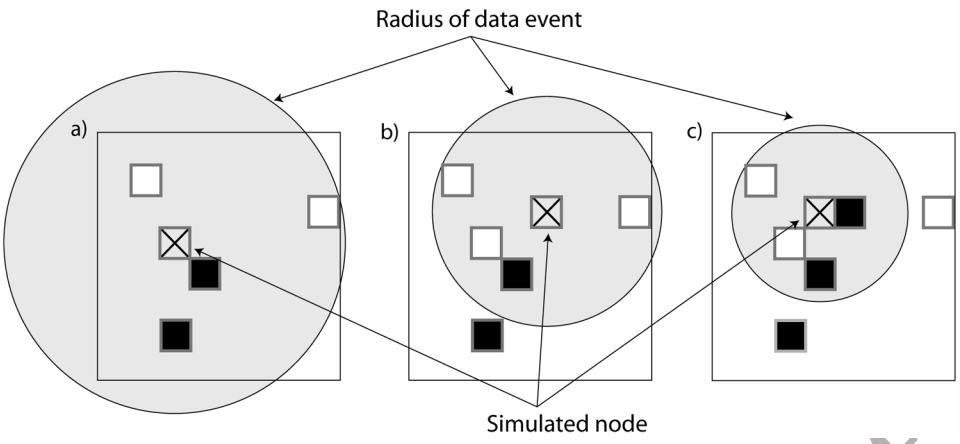

Figure 2. Illustration of the natural reduction of the data events size. The neighborhoods for simulating three successive grid nodes a, b, and c are defined as the four closest grid nodes. As the grid becomes more densely informed, the data events become smaller.

[12] 1. Each conditioning data is assigned to the closest grid node in the SG. If several conditioning data should be assigned to the same grid node, we assign the closest one to the center of the grid node.

[13] 2. Define a path through the remaining nodes of the SG. The path is a vector containing all the indices of the grid nodes that will be simulated sequentially. Random [*Strebelle*, 2002], unilateral (where nodes are visited in a regular order starting along one side of the grid [e.g. *Daly*, 2004]) or any other path can be used.

[14] 3. For each successive location $\mathbf{x}$ in the path:

[15] a. Find the neighbors of $\mathbf{x}$. They consist of a maximum of the n closest grid nodes $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ that were already assigned or simulated in the SG. If no neighbor is found for $\mathbf{x}$ (e.g., for the first node of an unconditional simulation), randomly take a node $\mathbf{y}$ in the TI and assign its value $Z(\mathbf{y})$ to $Z(\mathbf{x})$ in the SG. The algorithm can then proceed to the next node in the path.

[16] b. Compute the lag vectors $\mathbf{L} = \{\mathbf{h}_1, \cdots, \mathbf{h}_n\} = \{\mathbf{x}_1 - \mathbf{x}, \cdots, \mathbf{x}_n - \mathbf{x}\}$ defining the neighborhood of $\mathbf{x}$, $\mathbf{N}(\mathbf{x}, \mathbf{L}) = \{\mathbf{x} + \mathbf{h}_1, \cdots, \mathbf{x} + \mathbf{h}_n\}$. For example, in Figure 1a the neighborhood of the gray pixel (that represents the node to be simulated) consists of three lag vectors $\mathbf{L} = \{(1, 2), (2, 1), (-1, 1)\}$ corresponding to the relative locations of the three already simulated grid nodes.

[17] c. Define the data event $\mathbf{d}_n(\mathbf{x}, \mathbf{L}) = \{Z(\mathbf{x} + \mathbf{h}_1), \cdots, Z(\mathbf{x} + \mathbf{h}_n)\}$. It is a vector containing the values of the variable of interest at all the nodes of the neighborhood. In the example of Figure 1a, the data event is $\mathbf{d}_n(\mathbf{x}, \mathbf{L}) = \{0, 0, 1\}$.

[18] d. Define the search window in the TI. It is the ensemble of the locations $\mathbf{y}$ such that all the nodes $\mathbf{N}(\mathbf{y}, \mathbf{L})$ are located in the TI. The size of the search window is defined by the minimum and maximum values of the individual components of the lag vectors (Figure 1b).

[19] e. Randomly draw a location $\mathbf{y}$ in the search window and from that location scan systematically the search window. For each location $\mathbf{y}$:

[20] i. Find the data event $\mathbf{d}_n(\mathbf{y}, \mathbf{L})$ in the training image. In Figure 1c, a random grid node has been selected in the search window of the TI. The data event is retrieved and is found to be $\mathbf{d}_n(\mathbf{y}, \mathbf{L}) = \{1, 0, 1\}$.

[21] ii. Compute the distance $d\{\mathbf{d}_n(\mathbf{x}, \mathbf{L}), \mathbf{d}_n(\mathbf{y}, \mathbf{L})\}$ between the data event found in the SG and the one found in the TI. The distance is computed differently for continuous or discrete variables. Therefore we will describe this step more in detail later in the paper.

[22] iii. Store $\mathbf{y}$, $Z(\mathbf{y})$ and $d\{\mathbf{d}_n(\mathbf{x}, \mathbf{L}), \mathbf{d}_n(\mathbf{y}, \mathbf{L})\}$ if it is the lowest distance obtained so far for this data event.

[23] iv. If $d\{\mathbf{d}_n(\mathbf{x}, \mathbf{L}), \mathbf{d}_n(\mathbf{y}, \mathbf{L})\}$ is smaller than the acceptance threshold t, the value $Z(\mathbf{y})$ is sampled and assigned to $Z(\mathbf{x})$. This step is illustrated in Figure 1d. In that case, the current data event in the TI matches exactly the data event in the SG. The distance is zero and the value $Z(\mathbf{y}) = 1$ is assigned to the SG (Figure 1e).

[24] v. If the number of iterations of the loop i–iv exceeds a certain fraction of the size of the TI, the node $\mathbf{y}$ with the lowest distance is accepted and its value $Z(\mathbf{y})$ is assigned to $Z(\mathbf{x})$.

[25] The definition of the data event by considering the n closest informed grid nodes is very convenient as it allows the radius of the data events to decrease as the density of informed grid nodes becomes higher. This natural variation of the data events size has the same effect as multiple grids [*Strebelle*, 2002] and makes their use unnecessary. Figure 2 illustrates the decrease of the data events radius with neighborhoods defined by the four closest grid nodes.

[26] In the proposed method, the concept of a distance between data events $d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\}$ is extremely powerful, because it is flexible and can be adapted to the simulation of both continuous and categorical attributes. For categorical variables, we propose to use the fraction of nonmatching nodes in the data event, given by the indicator variable a that equals 0 if two nodes have identical value and 1 otherwise,

$$d\{\boldsymbol{d}_n(\boldsymbol{x}), \boldsymbol{d}_n(\boldsymbol{y})\} = \frac{1}{n}\sum_{i=1}^{n} a_i \quad \in [0,1],$$
$$\text{where } a_i = \begin{cases} 0 & \text{if } Z(\boldsymbol{x}_i) = Z(\boldsymbol{y}_i) \\ 1 & \text{if } Z(\boldsymbol{x}_i) \neq Z(\boldsymbol{y}_i) \end{cases}. \tag{2}$$

This measure of distance gives the same importance to all the nodes of the data event regardless of their location relative to the central node. It may be preferable to weight equation (2) according to the distance of each node in the template from

the central node, such as the norm of the lag vector $\mathbf{h}_i$ using a power function of order δ,

$$d\{\boldsymbol{d}_n(\boldsymbol{x}), \boldsymbol{d}_n(\boldsymbol{y})\} = \frac{\sum_{i=1}^{n} a_i \, \|\boldsymbol{h}_i\|^{-\delta}}{\sum_{i=1}^{n} \|\boldsymbol{h}_i\|^{-\delta}} \quad \in [0,1],$$

$$\text{where } a_i = \begin{cases} 0 & \text{if } Z(\boldsymbol{x}_i) = Z(\boldsymbol{y}_i) \\ 1 & \text{if } Z(\boldsymbol{x}_i) \neq Z(\boldsymbol{y}_i) \end{cases}. \tag{3}$$

Specific weights can be defined if some of the data event nodes are conditioning data, as described by *Zhang et al.* [2006]. This can be used to enforce more pattern consistency in the neighborhood of conditioning data or to give less importance to data presenting measurement errors. For all examples presented in this paper, we did not define specific weights for conditioning data. We also used $\delta = 0$ (i.e., all nodes of the data event have the same importance), which generally gives good results. Nevertheless, adjusting δ may be a way of obtaining images more representative of the TI while reducing CPU time.Kriging weights could be used here instead of power distance weighting, but this would involve tedious adjustment of covariance functions. Moreover, the CPU overburden involved in inverting a kriging matrix for each simulated node would be a high price to pay.

[27] For continuous variables, we propose to use a weighted Euclidian distance,

$$d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\} = \sqrt{\sum_{i=1}^{n} \alpha_i [Z(\mathbf{x}_i) - Z(\mathbf{y}_i)]^2} \quad \in [0,1], \tag{4}$$

where

$$\alpha_i = \frac{\|\mathbf{h}_i\|^{-\delta}}{d_{\max}^2 \sum_{j=1}^{n} \|\mathbf{h}_j\|^{-\delta}}, \quad d_{\max} = \max_{y \in TI} Z(y) - \min_{y \in TI} Z(y). \tag{5}$$

The proposed distance is the square root of the weighted mean square differences between $\mathbf{d}_n(\mathbf{x})$ and $\mathbf{d}_n(\mathbf{y})$. In practice, the data event $\mathbf{d}_n(\mathbf{y})$ matching perfectly $\mathbf{d}_n(\mathbf{x})$ is often not found in the TI, especially for continuous variables. This is why an acceptance threshold t is introduced. When $d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\}$ is smaller than t, the data event $\mathbf{d}_n(\mathbf{y})$ is accepted.

[28] The numerator in α_i weights the contribution of the data event nodes according to their distance to the central node. The denominator, although not needed for comparing distances between data events, is useful in practice to ensure that the distances are defined within the interval [0,1], making it easier to choose an appropriate acceptance threshold (for example, numerical tests have shown that 0.05 is a low threshold and 0.5 is a high threshold, whereas it can be more tedious without normalization).

[29] We do not suggest that the distances proposed above are exhaustive or appropriate for all possible situations. Other distances than the ones proposed above can be developed. For example, an alternative to (4) for continuous variables could be the normalized pair wise Manhattan distance,

$$d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\} = \frac{1}{n} \sum_{i=1}^{n} \frac{|Z(\mathbf{x}_i) - Z(\mathbf{y}_i)|}{d_{\max}} \quad \in [0,1]. \tag{6}$$

The choice of the distance measure used to compare data events of the simulation and of the TI should be adapted to the nature of the variable to simulate. For example, using distance (4) for the simulation of a categorical variable such as lithofacies would induce order relationships between the facies (i.e., facies 1 would be closer to facies 2 than to facies 3), which is conceptually wrong because facies codes are arbitrarily attributed. In section 7.1, we show how custom distances can be defined for specific problems.

[30] The quality of the pattern reproduction in the generated images depends on the size of the neighborhoods, the value of the acceptance threshold and the fraction of the TI that can be scanned for the simulation of each node. Certain settings of these parameters can be expensive in terms of CPU time. However, CPU burden can be alleviated using parallelization. Parallelizing the DS algorithm is straightforward on shared memory machines: each CPU performs the search in a limited portion of the TI. Our experience showed that this parallelization technique, using the OpenMP libraries, is very efficient in terms of speed-up. On a dual-core processor, the code runs about 1.9 times faster on two cores than on one, using various test cases. Moreover, recent parallelization strategies using Graphics Processing Units (GPU) may allow much shorter computation times. Parallelization on distributed memory machines is more challenging, but specific methods have been developed and have proven to be very efficient when applied to DS, showing good performance with as much as 54 processors [*Mariethoz*, 2010]. Nevertheless, even without parallelization, DS takes about the same time as traditional multiple-point simulators to obtain images of a similar quality.

4. Simulation of a Continuous Variable

[31] Flow and transport simulators deal with continuous properties, such as hydraulic conductivity, storativity, porosity, etc. However, categorical image generation methods are often used to obtain realistic connectivity patterns by reproducing the facies architecture of the subsurface. The simulated facies are then populated with continuous properties using other geostatistical techniques [*Caers*, 2005]. By directly simulating continuous variables, DS does not need this two-step approach to generate continuous variables fields presenting realistic connectivity patterns.

[32] Figure 3 shows a simulation using a TI borrowed from *Zhang et al.* [2006], consisting of a continuous variable with high connectivity of the low values. The TI (Figure 3a) and the simulation (Figure 3b) have the same size of 200 by 200 grid nodes. Distance (4) was used in the DS simulation. Conditioning data are 100 values taken in the TI and located at random positions in the simulation. This ensures that the conditioning data are not spatially coherent with the model but belong to the univariate marginal distribution. Despite this situation, the DS algorithm produces realizations that are consistent with the TI (high connectivity of the low values) and satisfactorily respect the conditioning data. Figure 3c shows the histogram reproduction. Note that a unilateral path was used here [*Daly*, 2004; *Pickard*, 1980]. Conditioning to data is possible with the unilateral path; this is accomplished by using large data events (80 nodes) including distant data points, which was not easily feasible with traditional multiple-point methods.

[33] This example shows that the DS method is able to simulate complex fields of continuous variables while con-

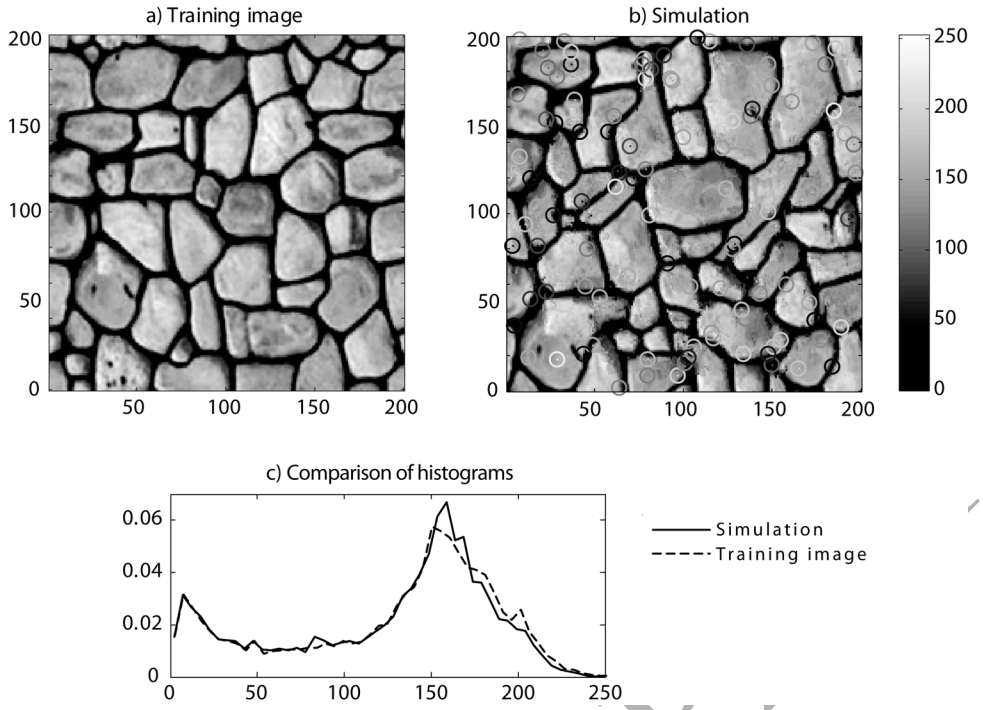

Figure 3. Illustration of the method using a continuous variable. (a) Training image with continuous variable. (b) One simulation using the unilateral path with 100 randomly located conditioning data (n = 80, t = 0.01). Positions of conditioning data are marked by circles whose colors indicate the values of the data. (c) Comparison of the histograms.

straining properties such as the statistical distribution and the connectivity patterns. Therefore, the method can produce specific types of heterogeneity that control the flow and transport behavior of the model.

5. Multivariate Case

[34] Contrary to existing multiple-point simulation techniques, DS is not limited by the dimension of the data events because there is no need to store their occurrences. Hence the data events can be defined through several variables that can be simulated jointly or used for conditioning following the same principle as cosimulation (it may be collocated or not). The training image is a multivariate field comprising m variables $Z_1(\mathbf{x}), \cdots, Z_m(\mathbf{x})$. Such multivariate fields are presented as "vector images" by *Hu and Chugunova* [2008]. Accounting for multiple-point dependence between variables means to respect cross correlations between all combinations of nodes within multivariate data events. The conditional cumulative density function (1) for the variable Z_k is then expressed as

$$\mathrm{F_k}\left(z, \mathbf{x}, \mathbf{d}^1_{n_1}, \cdots, \mathbf{d}^m_{n_m}\right) = \mathrm{Prob}\left\{Z_k(\mathbf{x}) \le z | \mathbf{d}^1_{n_1}, \cdots, \mathbf{d}^m_{n_m}\right\}, \quad k = 1, \cdots, m. \tag{7}$$

Each variable Z_k involved in the multivariate analysis can have a different neighborhood and a specific data event $\mathbf{d}^k_{n_k}(\mathbf{x}, \mathbf{L}^k) = \{Z_k(\mathbf{x} + \mathbf{h}^k_1), \cdots, Z_k(\mathbf{x} + \mathbf{h}^k_{n_k})\}$. The number n_k of nodes in the data event of each variable can be different, as well as the lag vectors $\mathbf{L}^k$. To simplify the notation, we just extend the previous concept of data event to the multivariate case: here the data event $\mathbf{d}_n(\mathbf{x})$ is the joint data event including all the individual data events $\mathbf{d}_n(\mathbf{x}) = \{\mathbf{d}^1_{n_1}(\mathbf{x}, \mathbf{L}^1), \cdots, \mathbf{d}^m_{n_m}(\mathbf{x}, \mathbf{L}^m)\}$. The distance between a joint data event found in the simulation and one found in the TI is defined as a weighted average of the individual distances defined previously,

$$d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\} = \sum_{k=1}^{m} w_k d\left\{\mathbf{d}^k_{n_k}(\mathbf{x}, \mathbf{L}^k), \mathbf{d}^k_{n_k}(\mathbf{y}, \mathbf{L}^k)\right\} \in [0,1], \quad \text{with } \sum_{k=1}^{m} w_k = 1, \text{ and } w_k \ge 0. \tag{8}$$

The weights w_k are defined by the user. They can for the fact that the pertinent measure of distance may be different for each variable. Multivariate simulations are performed using a single (random) path that visits all components of vector Z at all nodes of the SG.

[35] Figure 4 shows an example of a joint simulation of two variables that are spatially Dependent by some unknown function. For this synthetic example, the TI for variable 1 (Figure 4a) is a binary image representing a channel system [*Strebelle*, 2002]. The TI for variable 2 (Figure 4b) was obtained by smoothing variable 1 using a moving average with a window made of the 500 closest nodes and then adding an uncorrelated white noise uniformly distributed between 0 and 0.5. This secondary variable could represent the resistivity map corresponding to the lithofacies given by variable 1. The result is a bivariate training image where variables 1 and 2 are related via a multiple-point dependency. Figures 4c and 4d show one unconditional bivariate simulation using the TI described above. The categorical variable 1 uses distance (3)

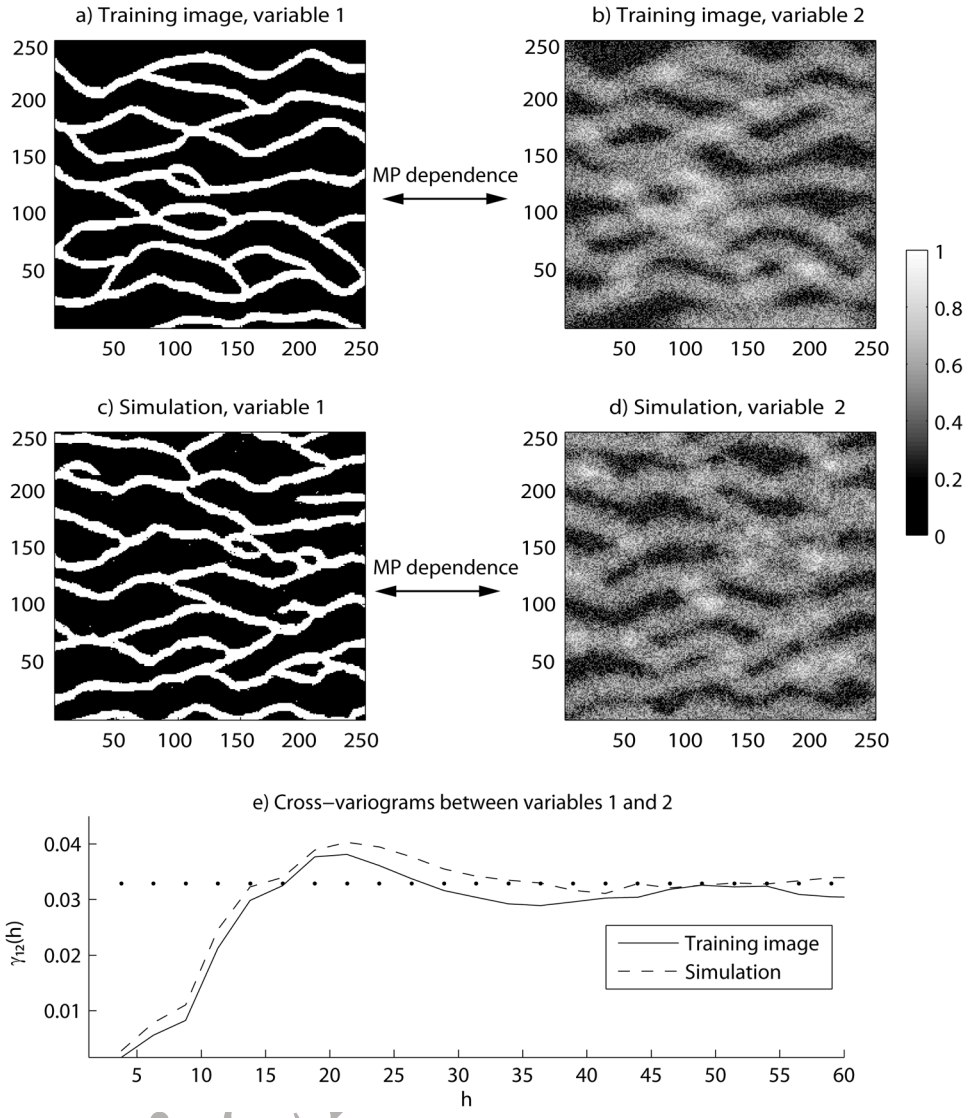

Figure 4. Joint simulation of two variables ($n_1 = 30$, $n_2 = 30$, $t = 0.01$, $w_1 = 0.5$, $w_2 = 0.5$). (a and b) The bivariate training image, with a complex multiple-point dependence. (c and d) One resulting bivariate simulation, where the MP dependence is reproduced. (e) Cross-variograms reproduction. Note that no variogram adjustment was necessary.

and the continuous variable 2 uses distance (4). The multiple-point dependence relating both variables is well reproduced, both visually and in terms of cross variograms (Figure 4e), which is a measure of two-point correlation. Note that addressing dependencies between categorical and continuous variables is usually awkward. The scatter plot depends on the facies numbering (which is arbitrary) and correlation factors are meaningless. Here DS is able to reproduce multiple-point dependence, including statistical parameters more complex than the scatterplot (e.g., cross variograms).

[36] Problems traditionally addressed by including exhaustively known secondary variables [e.g., *Mariethoz et al.*, 2009] are particular cases of the multivariate DS approach. Whereas existing MP methods consider only the secondary variable at the central node **x**, DS accounts for complex spatial patterns of the secondary variable because multiple-point statistics are considered for both primary and secondary variables.

[37] When one (or several) of the joint variables is already known, DS uses this information as indirect conditioning data (secondary variable) guiding the simulation of the other variables (primary variables) and then reducing uncertainty. In the following example, the aim is to simulate the primary variable knowing only the secondary variable and the multiple-point statistical relationship between primary and secondary variables, which is given via the bivariate TI. For illustration, consider Figures 4a and 4b as the bivariate TI and Figure 5b as the auxiliary variable for the simulation grid. Figure 5b was obtained as follows. First, Figure 5a was generated with an univariate unconditional simulation using Figure 4a as TI. Then, Figure 5b was computed from Figure 5a, applying a moving average followed by addition of a white noise. Hence, the aim is to reconstruct the reference field Figure 5a from Figure 5b and the multiple-point dependence given by the bivariate TI (Figures 4a and 4b), using multivariate DS.

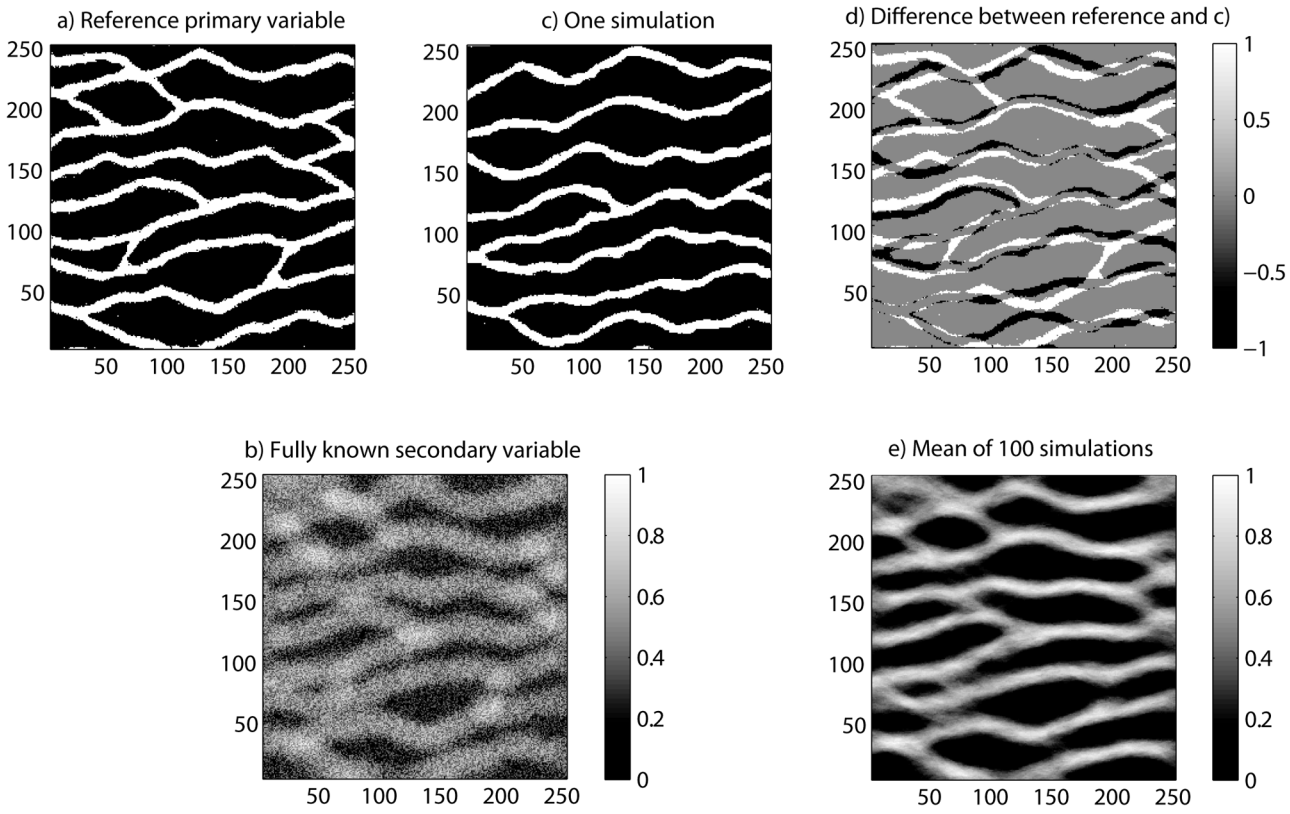

Figure 5. The use of a secondary variable to guide the simulation of a primary variable. (a) The reference primary variable, obtained with a univariate unconditional simulation using Figure 4a as TI. (b) The reference secondary variable computed by transformations of the primary variable (see text for details). The bivariate training images a and b describe the MP relationship between primary and secondary variables. (c) One multivariate simulation generated using the fully known secondary variable b as conditioning data ($n_1 = 30$, $n_2 = 30$, $t = 0.01$, $w_1 = 0.5$, $w_2 = 0.5$). (d) Superposition of one simulation and the reference. (e) Mean of 100 simulations.

[38] Figure 5c displays one realization of the primary variable, conditional to the exhaustively known secondary variable (Figure 5b). No conditioning data are available for the primary variable. The features of the reference field are correctly inferred from the information contained in the secondary variable, as shown in Figure 5d, where the reference (Figure 5a) and the simulation (Figure 5c) are superposed. In Figure 5e, the mean of 100 simulations is presented. In average, the channels are correctly located when compared to the reference.

[39] This technique could be applied for example, when a ground penetrating radar survey provides an exhaustive data set (secondary variable) and when the primary variable that needs to be characterized is the hydraulic conductivity [e.g., *Langsholt et al.*, 1998]. The relation between both variables is complex and not necessarily linear. DS can be applied to this type of problem if one can provide a bivariate TI. A possibility to construct the bivariate TI is to use first a TI of the hydraulic conductivity and then use a forward geophysical model to simulate the secondary variable.

6. Dealing With Nonstationarity

[40] Geological processes are intrinsically nonstationary. The ability to address nonstationarity is vital for the applicability of a geostatistical method in Earth Sciences. For existing MP methods, several techniques can be found in the literature to account for nonstationarity either of the TI or of the simulated field [*Chugunova and Hu*, 2008; *De Vries et al.*, 2009; *Journel*, 2002; *Strebelle*, 2002]. One of the ways of dealing with nonstationary TIs is to divide a nonstationary TI in stationary zones, each considered as a separate stationary TI [*Boucher*, 2009; *De Vries et al.*, 2009]. The simulation domain is then also divided into zones, each corresponding to a specific TI. In the framework of traditional multiple-point statistics, using multiple TIs involves creating one data events catalogue per training image [*Wu et al.*, 2008]. Although, it may be difficult in practice to define the stationary zones, it could be applied easily with DS by scanning a different part of a TI or different TIs for each simulated zone. There would be no limitations to the number of TIs and zones related to memory requirements. More generally, all the techniques cited above can be used with DS, but new possibilities are also offered by exploiting the specificities of DS.

6.1. Addressing Nonstationarity With Specific Distances

[41] We discussed above how the distance measure should be chosen according to the nature of the variables at stake.

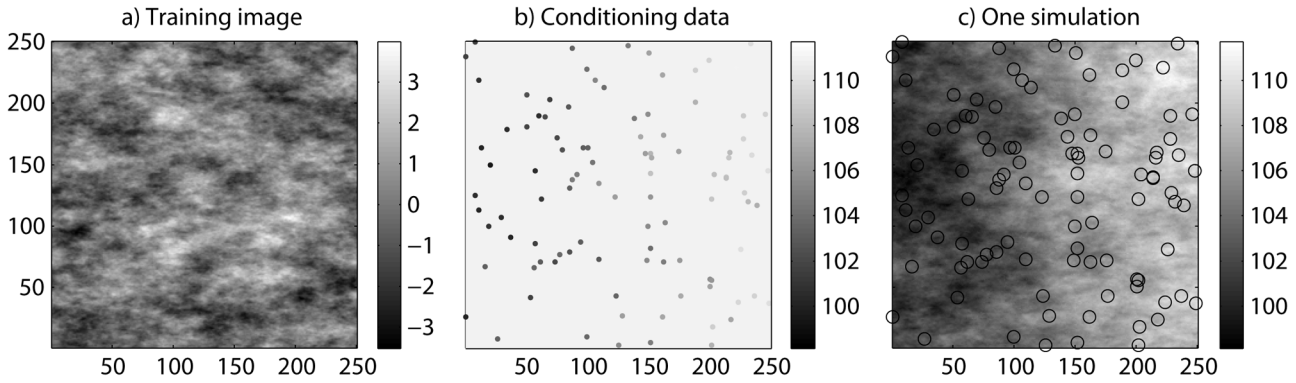

Figure 6. Simulation using a variation-based distance. (a) Multi-Gaussian stationary training image. (b) Nonstationary data set (100 points data), with values in a different range than those of the training image. (c) One simulation with variation-based distance ($n = 15$, $t = 0.01$). Circles represent the location of the 100 conditioning data.

Following this idea, we propose to forge distances adapted to nonstationary cases. An example of such custom distance measure is the pair-wise Euclidean distance relative to the mean of the data event,

$$d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\} = \left(\sum_{i=1}^{n} \alpha_i [(Z(\mathbf{x}_i) - \bar{Z}(\mathbf{x})) - (Z(\mathbf{y}_i) - \bar{Z}(\mathbf{y}))]^2\right)^{1/2} \in [0,1], \quad (9)$$

with $\overline{Z}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} Z(\mathbf{x}_i)$. When a matching data event is found in the TI, the local mean of the SG data event is added to the value found in the TI. Therefore, the value $Z(\mathbf{y}) - \overline{Z}(\mathbf{y}) + \overline{Z}(\mathbf{x})$ is attributed to the simulated node. The distance described in equation (9) compares data events by their relative variations only and not their actual values. This variation-based distance can be very useful when considering first-order nonstationary phenomena. We illustrate this situation with the example depicted in Figure 6. The available training image (Figure 6a) is a multi-Gaussian field with zero mean and unit variance, resulting in minimum and maximum values of −3.52 and 3.99, respectively. It was generated using an exponential variogram model, with ranges of 35 units along the x axis and 25 units along the y axis. Its size is 250 by 250 grid nodes. One hundred conditioning data are available (Figure 6b), but they are not compatible with the training image, as their values span between a minimum of 99.55 and a maximum of 110.92, with a mean of 105.12. Moreover, these data show nonstationarity. Because the distance (9) is based on the variations of the values in the data event, it is possible to find matches between the data events found in the data and the ones of the TI despite the difference in the range and the nonstationarity. The resulting simulations (one is shown in Figure 6c) display the same variable range (minimum, 98.13; maximum, 111.72; mean, 104.87) and the same nonstationary behavior as the data, but also a spatial structure similar to what is found in the TI. In this case, nonstationarity can be seen as a locally varying mean, and therefore distance (9) can accommodate it well. If the nonstationarity was more complex, such as, for example, structures ranging from channels to lenses, this distance measure would not be appropriate.

[42] This example shows that variation-based distance can be used when a conceptual model allows the geologist to provide a training image, but when the data indicate the presence of nonstationarity and inadequacy of the ranges given in the TI. Moreover, it emphasizes the flexibility offered by using distances between data events, which is one of the major advantages of the DS approach.

6.2. Addressing Nonstationarity With Transformation of Data Events

[43] Traditional multiple-point simulation implementations such as *snesim* include the possibility of imposing transformations on the structures found in the TI. This is done by first constructing the data events catalogue using a transformed template and then simulating values with a non-transformed template [*Strebelle*, 2002]. The most commonly implemented transformations are rotation and affinity (application of homothetic transformations on the template). This feature is very useful when the modeler has a single stationary training image and wants to use it for the simulation of nonstationary fields. If many different transformations have to be applied on the simulation grid, most approaches store as many data events catalogues. The DS approach also allows these transformations. Simply scanning the TI with a transformed data event gives the same results as the traditional technique. Moreover, transformations are not defined by zones, but as a continuum, because the transformation can be different for each simulated node. In some cases, rotation or affinity may result in large data events that do not fit in the TI. In such cases, the data event nodes located outside of the TI are ignored until it becomes possible to scan the TI with this new, reduced data event.

[44] Figure 7 shows an example of such transformation, with angle and affinity maps (Figures 7a and 7b) defined by continuous variables. All angles between −180° and 180° are represented, and the affinity ratios range from 1 at the center of the image to 0.4 in the corners (meaning that all structures are reduced to 40% of the size they have in the TI). The training image (Figure 7c) is much smaller (250 by 250 nodes) than the simulation domain (1000 by 1000 nodes) and represents horizontal channels. This combined transformation (rotations + affinities) results in channels oriented in all

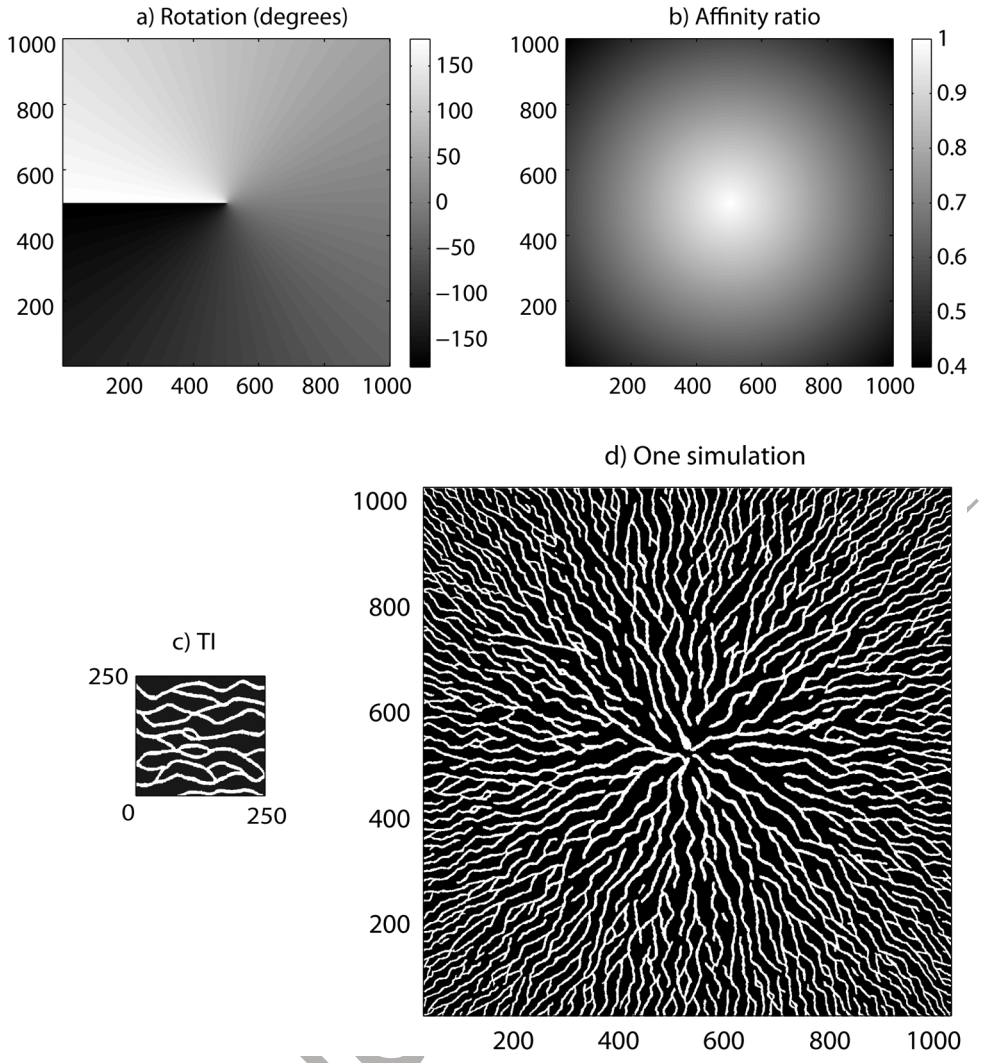

Figure 7. Transformations of the data events. (a) Rotation map. (b) Affinity map. (c) Stationary training image. (d) Simulation with transformed data events ($n = 30$, $t = 0$).

directions and becoming thinner as they are located further away from the centre (Figure 7d).

6.3. Addressing Nonstationarity With a Secondary Variable

[45] A situation where a secondary variable can be extremely powerful occurs when the training image itself is nonstationary. This is the case, for example, when the TI is taken from direct field observation or when it is obtained from a process based simulation. When this type of nonstationarity occurs, one can introduce one or several secondary variables to model the nonstationarity in the TI and in the simulation, as it was proposed by *Chugunova and Hu* [2008]. The approach uses spatially continuous auxiliary variables to distinguish the regions where similar patterns occur. This secondary variable can be rather abstract, it just needs to be defined on the training image and on the simulation grid and it must have similar values in regions where similar patterns occur. While this idea was implemented by *Chugunova and Hu* [2008] by modifying the probability tree structure of *snesim*, it is accomplished in a straightforward manner with DS by using a multivariate TI with variable 1 being the variable of interest and the other variables describing the non-stationarity of variable 1.

[46] Figure 8 illustrates this concept in a simple situation. The TI for the primary variable is binary and shows a set of rotating channels. The orientation of the channels changes as a function of the X coordinate (Figure 8a). Therefore, a simple way to describe this nonstationarity is to use the X coordinate as the secondary variable (Figure 8b). On the simulation, if ones want to have horizontal channels on the top, vertical channels in the bottom and a smooth transition in between, one first generates a map of the secondary variable such as the values of this map describe the required nonstationarity: in this case, the secondary variable (X coordinate map) is rotated so that zeros are at the bottom, ones are on top, and intermediate values are in between (Figure 8d). Using that secondary information and the standard multivariate collocated cosimulation DS method presented earlier, the resulting simulation displays the desired non-stationary behavior (Figure 8c). For these simulations, the neighborhoods are made of $n_1 = 30$ nodes for the primary variable and $n_2 = 1$ for the secondary variable, because a single node is enough to characterize the nonstationarity. The weights of both variables are kept equal, with $w_1 = w_2 = 0.5$.

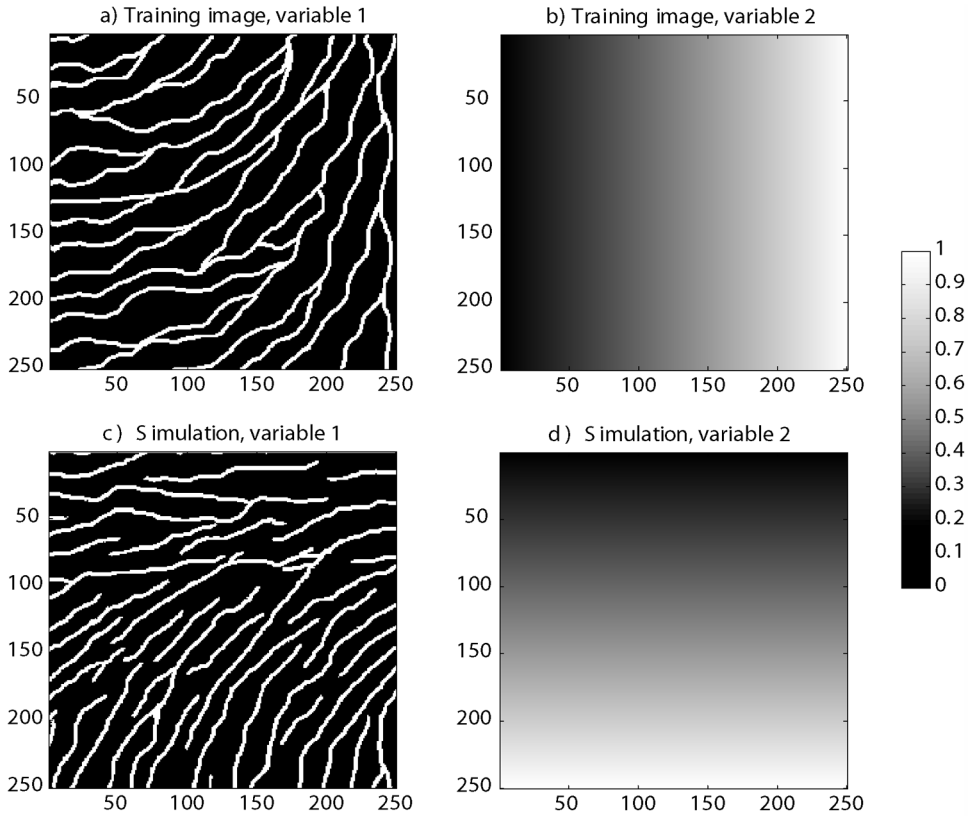

Figure 8. The use of a secondary variable to model nonstationarity. (a) Variable 1 of nonstationary training image. (b) Dependent joint variable describing the nonstationarity of variable 1 in training image. (c) Resulting simulation for variable 1 ($n_1 = 30$, $n_2 = 1$, $t = 0.01$, $w_1 = 0.5$, $w_2 = 0.5$). (d) Dependent joint variable (exhaustively known) describing the nonstationarity of variable 1 in simulation.

[47] Although this is a simple example, the use of a continuous secondary variable to describe nonstationarity allows accounting for very rich types of nonstationarity such as a change in the type of structures encountered.

7. Improving Pattern Reproduction

[48] Accurate pattern reproduction can be jeopardized when a data event cannot be found in the TI. This problem is common to all multiple-point simulations methods and is more acute when a random path is used in the simulation grid. In traditional multiple-point simulation algorithms, this issue is usually dealt with by dropping the neighbor node that is the farthest away from the central node and, by performing a search in the data events catalogue for this new, reduced pattern [*Strebelle*, 2002]. The main drawback of this procedure is that it induces a degradation of the pattern reproduction by artificially reducing the template size for the computation of the ccdf (1). Such degradation can lead to a lack of spatial continuity of the simulated structures (such as channels). Several authors have proposed methods to improve patterns reproduction. *Strebelle and Remy* [2005] locate the nodes that were simulated using a reduced neighborhood and resimulate the dropped neighbors at the end of each multigrid step. This method does not remove all the inconsistencies in the simulated patterns but performs additional simulation attempts with updated neighborhoods. As problematic values are temporarily accepted (until the entire multigrid is simulated), they propagate inconsistencies to nodes that are simulated later. Therefore, if a node is successfully resimulated, it is not guaranteed that all its neighbors are consistent between each other. Another algorithm, proposed by *Stien et al.* [2007], does not temporarily accept values generating conflicts but deletes the problematic nodes in the neighborhood. At the end of a multigrid level, these nodes are simulated. The process is iterative and needs specific parameters to ensure convergence. Although this method avoids the propagation of inconsistencies by deleting them, it does not resolve the problem of the origin of these problematic patterns. Indeed, inconsistencies exist because other nearby problematic patterns occurred previously in the simulation process. In our opinion, the only way to deal with this problem is to immediately address the entire cascade of causes at the origin of problematic patterns.

[49] In the context of simulations using a unilateral path [*Daly*, 2004], *Suzuki and Strebelle* [2007] developed the real-time postprocessing method (RTPP) that walks back the unilateral path when problematic neighborhoods are encountered and resimulates the most recent nodes until the produced patterns satisfactorily match the ones of the TI. The limits of this method are that it is applicable to the first stage of the simulation only (the first multigrid) and only when using a unilateral path. Therefore, like all simulation methods using the unilateral model, it suffers from difficulties in honoring conditioning data. Nevertheless, this method has the advantage of correcting all inconsistencies because it resimulates the neighborhoods of the problematic nodes and

not only the problematic nodes themselves. As inconsistencies are resimulated immediately, it avoids propagation to their neighbors.

[50] In this context, we propose a new algorithm, the syn-processing, aimed at improving the reproduction of patterns. It is generic enough to be applicable to any type of path and with or without multigrids. As the RTPP, it resimulates values as soon as inconsistencies are met. It is based on the idea that when an inconsistent pattern (with respect to the TI) is found, it is because other inconsistencies occurred previously in the simulation process. Therefore, before resimulating problematic nodes, their neighborhoods also need to be (at least partially) resimulated. If inconsistencies appear during this resimulation, further resimulation needs to be performed. Hence, the algorithm is of a recursive nature.

[51] The syn-processing algorithm consists in the following steps at each simulated node $\mathbf{x}$: check if the simulated value $Z(\mathbf{x})$ is acceptable. The acceptance criterion can be a minimum number of dropped neighbor nodes in the framework of classical multiple-point implementation. In the case of DS, the criterion is that the minimum distance $d\{\mathbf{d}_n(\mathbf{x}), \mathbf{d}_n(\mathbf{y})\}$ found is below a threshold.

[52] If the criterion is not met, the simulation of $Z(\mathbf{x})$ is postponed and one of its neighbors $\mathbf{N}^{-1}(\mathbf{x})$, taken among those that do not belong to the original set of conditioning data, is resimulated taking into account the same criterion. For the simulation of the node $N^{-D}(\mathbf{x})$:

[53] 1. If the criterion is not met for $Z\{N^{-D}(\mathbf{x})\}$, delete it and resimulate one of its neighbors, $N^{-(D+1)}(\mathbf{x})$.

[54] 2. If the criterion is met for $Z\{N^{-D}(\mathbf{x})\}$, accept this value and try simulating $Z\{N^{1-D}(\mathbf{x})\}$.

[55] Note that D is the number of deleted nodes for the initial node $\mathbf{x}$. To ensure convergence, a maximum allowed number of deletions must be set.

[56] Syn-processing can sometimes delete and resimulate the same nodes in a cyclic manner. Such cycles are a waste of time as they do not improve the simulation. This can be avoided by keeping a record of all deletions. Before each deletion, the analysis of this record allows finding if the present state of the SG already occurred in the past. If it is the case, another random neighbor is chosen in order to break the cycle.

[57] Tests show that syn-processing efficiently improves pattern reproduction, as well as conditioning to local data. As the algorithm is recursive, CPU time can be adversely affected depending on the criterion to accept a simulated value. If the criterion is very strict (such as $t = 0$ for a continuous variable) and if the maximum allowed number of iterations is very large, convergence can be compromised. On the other hand, improving pattern reproduction increases the global coherence of the simulation with respect to the TI. It becomes then easier to find matching data events in the TI, thus making the scan process faster for the remaining nodes. In certain cases, syn-processing can even reduce simulation time up to a factor 2 while improving simulations quality. Moreover, tests showed that performing syn-processing only at the beginning of the simulation is sufficient to obtain better reproduction of large-scale features and general connectedness of the simulated structures, as compared to simulations without syn-processing. Therefore, a tradeoff needs to be achieved between the different parameters governing the simulation, in order to obtain optimal results at the lowest possible CPU cost.

[58] Note that syn-processing was used when generating the simulation examples presented in Figures 5a and 8. For comparison, note the difference of continuity between Figure 4c, where no syn-processing was used, and Figure 5a that was generated using syn-processing. The latter figure reproduces better the sinuosity and the connectivity of the channels found in the TI.

8. Discussion and Conclusion

[59] In this paper, we presented the direct sampling (DS) simulation method and a recursive syn-processing algorithm to enhance the quality of pattern reproductions in the simulations.

[60] As compared to traditional multiple point techniques such as *snesim* [*Strebelle*, 2002], the proposed method is able to generate exactly the same ensemble of stochastic realizations if we use the same neighborhood as *snesim* and if we use multigrids. The advantage of DS is that it allows respecting the conditional probability distribution that could be computed from the training image without having to actually compute it. Because it is not necessary to estimate this conditional probability distribution, the method can be applied in situations where the traditional approach failed such as very large number of facies, continuous variables or multivariate cases. In addition, when a classical multiple-point technique does not find a certain data configuration in a TI, it usually drops successive data points from the data event until it finds a configuration that exists in the TI. This procedure may be rather arbitrary. Here we avoid this practice by using a distance between two data events. When the data event cannot be found exactly, we select one data event that is acceptable within a predefined error range. The distance threshold between data events is an additional parameter that allows to control the model and how the DS will reproduce the patterns found in the TI. Setting this threshold above a value 0 means that the user accepts differences between the TI and the simulation. This clearly shows that the probabilistic model proposed in this work is a numerical model that includes not only the TI but also the acceptance threshold as well as the number of neighbors and any additional parameters that needs to be defined when accounting for nonstationarity. The validity of a particular model based on the DS method, like the validity of any stochastic model, can then be tested by standard cross-validation techniques on any real data set. This classical approach can be used to select the best model amongst a series of possible TI and parameter sets and also to compare its performances with other more standard stochastic models.

[61] By using distances between data events, DS offers the possibility to use training images that can either be categorical or continuous, uni- or multivariate, stationary or nonstationary. This can be extremely powerful when realistic geological structures must be modeled, as is commonly the case for hydrogeological problems like contaminant migration, which are strongly influenced by heterogeneity and connectivity of the geological structures.

[62] The multivariate framework offered by DS opens new perspectives for the integration of different data types in groundwater and surface hydrology. By accounting for multiple-point dependence between several variables, DS can exploit nonparametric relationships between variables of different nature, such as between categorical and continuous

variables. Possible applications can be very diverse since categorical variables (e.g., geology, soil type, land cover category, vulnerability class) and continuous variables (e.g., porosity, concentration, recharge rate, rainfall) are often related and widely used in hydrology but are very seldom measured exhaustively. Therefore, these variables often must be interpolated. The DS method could then be used in this context.

[63] In addition to the wide spectrum of potential applications, DS has computational advantages that make it easier to apply than traditional multiple-point methods. DS massively reduces memory usage because no catalogue of data events needs to be stored. This implies that the size of the neighborhood is not limited by memory considerations. The data event can be spread across many different variables, allowing to perform multivariate simulations of variables presenting complex multiple-point dependence.

[64] Because multiple-point statistics are not stored, DS does not need a fixed geometry of the data events. The shape of the data event can change at each simulated node and so can the search window. Hence, the data events are always adapted to the simulation path. The size of the data events is only limited by the size of the TI and is controlled by a maximum number n of nodes. In certain cases, it can be useful to limit the radius of the data events, for example, when considering nonstationary variables, to avoid capturing nonstationarity within the data events. It is also useful if the simulation is larger than the training image. In this case, very large data events can result in very small search windows, leading to a bias toward reproducing the statistical properties of a small central portion of the TI.

[65] A related advantage of the DS approach is that multigrids (a step-like decrease in the template dimension) are replaced by a progressive (linear) decrease of the size of the data event as a function of the density of simulated nodes. It ensures that structures of all sizes are present in the simulation. Abandoning multigrids avoids problems related to the migration of conditioning data on coarse multigrid levels. By avoiding multigrids, DS is easy to implement, easy to parameterize and has no problems accommodating large data sets.

[66] A very important point is that DS does not require prohibitive CPU time, with performances comparable to existing methods. This good performance is possible because the algorithm searches only a single matching data event, and therefore, the whole TI often does not need to be scanned. There is a tradeoff between CPU time and the quality of the generated images, controlled by parameters such as the size of the neighborhoods, the value of the acceptance threshold and the fraction of the TI that can be scanned for the simulation of each node. However, using parallelization allows easily increasing the performance of DS.

[67] The algorithms described in this paper are the object of an international patent application (PCT/EP2008/009819).

[68] **Acknowledgments.** We thank the Swiss National Science Foundation (grants PP002-1065557 and PBNEP2-124334) and the Swiss Confederation's Innovation Promotion Agency (CTI project 8836.1 PFES-ES) for funding this work. We also thank Roland Froidevaux, Pierre Biver, Tatiana Chugunova, Denis Allard, Olivier Besson, Alexandre Boucher, Jef Caers, Jesus Carrera, Peter Atkinson, Lin Hu, André Journel, and anonymous reviewers for their comments and advices.

References

Aitokhuehi, I., and L. J. Durlofsky (2005), Optimizing the performance of smart wells in complex reservoirs using continuously updated geological models, *J. Petrol. Sci. Eng.*, *48*(3–4), 254–264.

Alcolea, A., P. Renard, G. Mariethoz, and F. Bretone (2009), Reducing the impact of a desalination plant using stochastic modeling and optimization techniques, *J. Hydrol.*, *365*(3–4), 275–288, doi:10.1016/j.jhydrol.2008.11.034.

Alcolea, A., and P. Renard (2010), The blocking moving window sampler: Conditioning multiple-point simulations to hydrogeological data, *Water Resour. Res.*, *46*, W08511, doi:10.1029/2009WR007943.

Arpat, B., and J. Caers (2007), Conditional simulations with patterns, *Math. Geol.*, *39*(2), 177–203.

Boucher, A. (2007), *Algorithm-Driven and Representation-Driven Random Function: A New Formalism for Applied Geostatistics*, Stanford Cent. for Reservoir Forecast., Palo Alto, Calif.

Boucher, A. (2009), Considering complex training images with search tree partitioning, *Compute. Geosci.*, *35*(6), 1151–1158.

Caers, J., S. Strebelle, and K. Payrazyan (2003), Stochastic integration of seismic data and geologic scenarios: A West Africa submarine channel saga, *Leading Edge*, *22*(3), 192–196.

Caers, J. (2005), *Petroleum Geostatistics*, 88 pp., Soc. of Petroleum Eng., Richardson.

Caers, J., and T. Hoffman (2006), The probability perturbation method: A new look at Bayesian inverse modeling, *Math. Geol.*, *38*(1), 81–100.

Capilla, J., and C. Llopis-Albert (2009), Gradual conditioning of non-Gaussian transmissivity fields to flow and mass transport data: 1. Theory, *J. Hydrol.*, *371*, 66–74.

Chugunova, T., and L. Hu (2008), Multiple-point simulations constrained by continuous auxiliary data, *Math. Geosci.*, *40*(2), 133–146.

Dagan, G. (1976), Stochastic conceptual analysis of one-dimensional groundwater flow in nonuniform homogeneous media—Comment, *Water Resour. Res.*, *12*(3), 567–567, doi:10.1029/WR012i003p00567.

Dagan, G. (1986), Statistical theory of groundwater flow and transport: Pore to laboratory, 25 laboratory to formation, and formation to regional scale, *Water Resour. Res.*, *22*, 120S–134S, doi:10.1029/WR022i09Sp0120S.

Dagan, G. (1989), *Flow and Transport in Porous Formations*, Springer, Berlin, Germany.

Dagan, G. (2004), On application of stochastic modeling of groundwater flow and transport, *Stochastic Environ. Res. Risk Assess.*, *18*(4), 266–267.

Daly, C. (2004), Higher order models using entropy, Markov random fields and sequential simulation, paper presented at Geostatistics Banff 2004, Kluwer Acad., Banff, Alberta.

De Marsily, G., F. Delay, J. Gonçalvès, P. Renard, V. Teles, and S. Violette (2005), Dealing with spatial heterogenity, *Hydrogeol. J.*, *13*(1), 161–183.

De Vries, L., J. Carrera, O. Falivene, O. Gratacos, and L. Slooten (2009), Application of multiple point geostatistics to non-stationary images, *Math. Geosci.*, *41*(1), 29–42.

El Ouassini, A., A. Saucier, D. Marcotte, and B. Favis (2008), A patchwork approach to stochastic simulation: A route towards the analysis of morphology in multiphase systems, *Chaos Sol. Frac.*, *36*, 418–436.

Emery, X. (2007), Using the Gibbs sampler for conditional simulation of Gaussian-based random fields, *Compute. Geosci.*, *33*, 522–537.

Feyen, L., and S. Gorelick (2004), Reliable groundwater management in hydroecologically sensitive areas, *Water Resour. Res.*, *40*, W07408, doi:10.1029/2003WR003003.

Feyen, L., and J. Caers (2006), Quantifying geological uncertainty for flow and transport modelling in multimodal heterogeneous formations, *Adv. Water Resour.*, *29*(6), 912–929.

Freeze, R. A. (1975), A stochastic-conceptual analysis of one dimensional groundwater flow in nonuniform homogemeous media, *Water Resour. Res.*, *11*(5), 725–741, doi:10.1029/WR011i005p00725.

Freeze, R. A., J. Massmann, L. Smith, T. Sperling, and B. James (1990), Hydrogeological decision analysis: 1. A framework, *Ground Water*, *28*(5), 738–766.

Gelhar, L. (1993), *Stochastic Subsurface Hydrology*, Prentice Hall, Englewood Cliffs, NJ.

Gómez-Hernández, J. J., and X.-H. Wen (1998), To be or not to be multi-Gaussian? A reflection on stochastic hydrogeology, *Adv. Water Resour.*, *21*(1), 47–61.

Guardiano, F., and M. Srivastava (1993), Multivariate geostatistics: Beyond bivariate moments, in *Geostatistics-Troia*, edited by A. Soares, pp. 133–144, Kluwier Acad., Dordrecht, Netherlands.

Hoffman, B. T., and J. Caers (2007), History matching by jointly perturbing local facies proportions and their spatial distribution: Application to a North Sea reservoir, *J. Petrol. Sci. Eng.*, *57*(3–4), 257–272.

Hu, L., and T. Chugunova (2008), Multiple-point geostatistics for modeling subsurface heterogeneity: A comprehensive review, *Water Resour. Res.*, *44*, W11413, doi:10.1029/2008WR006993.

Huysmans, M., and A. Dassargues (2009), Application of multiple-point geostatistics on modelling groundwater flow and transport in a cross-bedded aquifer (Belgium), *Hydrogeol. J.*, *17*, 1901–1911.

Journel, A. (1983), Nonparametric estimation of spatial distributions, *Math. Geol.*, *15*(3), 445–468.

Journel, A. (2002), Combining knowledge from diverse sources: An alternative to traditional data independence hypotheses, *Math. Geol.*, *34*(5), 573–596.

Journel, A., and T. Zhang (2006), The necessity of a multiple-point prior model, *Math. Geol.*, *38*(5), 591–610.

Kerrou, J., P. Renard, H.-J. Hendricks-Franssen, and I. Lunati (2008), Issues in characterizing heterogeneity and connectivity in non-multi-Gaussian media, *Adv. Water Resour.*, *31*(1), 147–159.

Klise, K., G. Weissmann, S. McKenna, E. Nichols, J. Frechette, T. Wawrzyniec, and V. Tidell (2009), Exploring solute transport and streamline connectivity using lidar-based outcrop images and geostatistical representations of heterogeneity, *Water Resour. Res.*, *45*, W05413, doi:10.1029/2008WR007500.

Knudby, C., and J. Carrera (2005), On the relationship between indicators of geostatistical, flow and transport connectivity, *Adv. Water Resour.*, *28*(4), 405–421.

Koltermann, C., and S. Gorelick (1996), Heterogeneity in sedimentary deposits: A review of structure-imitating, process-imitating, and descriptive approaches, *Water Resour. Res.*, *32*(9), 2617–2658, doi:10.1029/96WR00025.

Langsholt, E., N. Kitterød, and L. Gottschalk (1998), Development of three-dimensional hydrostratigraphical architecture of the unsaturated zone based on soft and hard data, *Ground Water*, *36*(1), 104–111.

Liu, Y., A. Harding, W. Abriel, and S. Strebelle (2004), Multiple-point simulation integrating wells, three-dimensional seismic data, and geology, *AAPG Bull.*, *88*(7), 905–921.

Mariethoz, G., P. Renard, and R. Froidevaux (2009), Integrating collocated auxiliary parameters in geostatistical simulations using joint probability distributions and probability aggregation, *Water Resour. Res.*, *45*, W08421, doi:10.1029/2008WR007408.

Mariethoz, G. (2010), A general parallelization strategy for random path based geostatistical simulation methods, *Compute. Geosci.*, *37*(7), 953–958, doi:10.1016/j.cageo.2009.11.001.

Mariethoz, G., and P. Renard (2010), Reconstruction of incomplete data sets or images using direct sampling, *Math. Geosci.*, *42*(3), 245–268, doi:10.1007/s11004-010-9270-0.

Matheron, G. (1966), Structure et composition des perméabilités, *Rev. de l'IFP*, *21*, 564–580.

Matheron, G. (1967), *Eléments Pour Une Théorie Des Milieux Poreux*, Masson, Paris.

Michael, H., A. Boucher, T. Sun, J. Caers, and S. Gorelick (2010), Combining geologic-process models and geostatistics for conditional simulation of 3-D subsurface heterogeneity, *Water Resour. Res.*, *46*, W05527, doi:10.1029/2009WR008414.

Neuwiler, I., and O. Cirpka (2005), Homogenization of Richards equation in permeability fields with different connectivities, *Water Resour. Res.*, *41*, W02009, doi:10.1029/2004WR003329.

Pickard, D. (1980), Unilateral Markov fields, *Adv. Appl. Probab.*, *12*, 655–671.

Renard, P. (2007), Stochastic hydrogeology: What professionals really need?, *Ground Water*, *45*(5), 531–541.

Ronayne, M., S. Gorelick, and J. Caers (2008), Identifying discrete geologic structures that produce anomalous hydraulic response: An inverse modeling approach, *Water Resour. Res.*, *44*, W08426, doi:10.1029/2007WR006635.

Rubin, Y. (2003), *Applied Stochastic Hydrogeology*, 391 pp., Oxford Univ. Press, New York.

Sanchez-Vila, X., A. Guadagnini, and J. Carrera (2006), Representative hydraulic conductivities in saturated groundwater flow, *Rev. Geophys.*, *44*, RG3002, doi:10.1029/2005RG000169.

Sánchez-Vila, X., J. Carrera, and J. P. Girardi (1996), Scale effects in transmissivity, *J. Hydrol.*, *183*(1–2), 1–22.

Schaap, J., P. Lehmann, A. Kaestner, P. Vontobel, R. Hassanein, G. Frei, G. de Rooij, E. Lehmann, and H. Flühler (2008), Measuring the effect of structural connectivity on the water dynamics in heterogeneous porous media using speedy neutron tomography, *Adv. Water Resour.*, *31*, 1233–1241.

Shannon, C. E. (1948), A mathematical theory of communication, *Bell Syst. Tech. J.*, *27*, 379–423.

Stien, M., P. Abrahmsen, R. Hauge, and O. Kolbjørnsen (2007), Modification of the Snesim Algorithm, paper presented at Petroleum Geostatistics 2007, EAGE, Cascais, Portugal, 10–14 Sept.

Strebelle, S. (2002), Conditional simulation of complex geological structures using multiple-point statistics, *Mathematical Geology*, *34*(1), 1–22.

Strebelle, S., K. Payrazyan, and J. Caers (2003), Modeling of a deepwater turbidite reservoir conditional to seismic data using principal component analysis and multiple-point geostatistics, *SPE J.*, *8*(3), 227–235.

Strebelle, S., and N. Remy (2005), Post-processing of multiple-point geostatistical models to improve reproduction of training patterns, in *Geostatistics Banff 2004*, edited by O. L. A. C. V. Deutsch, pp. 979–988, Springer, Dordrecht.

Suzuki, S., and S. Strebelle (2007), Real-time post-processing method to enhance multiple-point statistics simulation, paper presented at Petroleum Geostatistics 2007, EAGA, Cascais, Portugal, 10–14 Sept.

Suzuki, S., and J. Caers (2008), A distance-based prior model parameterization for constraining solutions of spatial inverse problems, *Math. Geosci.*, *40*, 445–469.

Wasserman, L. (2006), *All of Nonparametric Statistics*, 268 pp., Springer.

Wen, X., and J. Gomez-Hernandez (1998), Numerical modeling of macrodispersion in heterogeneous media: A comparison of multi-Gaussian and non-multi-Gaussian models, *J. Contam. Hydrol.*, *31*(1), 129–156.

Western, A., G. Blöschl, and R. Grayson (2001), Toward capturing hydrologically significant connectivity in spatial patterns, *Water Resour. Res.*, *37*(1), 83–97, doi:10.1029/2000WR900241.

Wu, J., A. Boucher, and T. Zhang (2008), A SGeMS code for pattern simulation of continuous and categorical variables: FILTERSIM, *Compute Geosci.*, *34*(12), 1863–1876.

Zhang, D. (2002), *Stochastic Methods for Flow in Porous Media*, 350 pp., Academic, San Diego, Calif.

Zhang, T., P. Switzer, and A. Journel (2006), Filter-based classification of training image patterns for spatial simulation, *Math. Geol.*, *38*(1), 63–80.

Zinn, B., and C. Harvey (2003), When good statistical models of aquifer heterogeneity go bad: A comparison of flow, dispersion, and mass transfer in connected and multivariate Gaussian hydraulic conductivity fields, *Water Resour. Res.*, *39*(3), 1051, doi:10.1029/2001WR001146.

G. Mariethoz, ERE Department, Stanford University, 367 Panama St., Rm. 65, Stanford, CA 94305, USA. (gregoire.mariethoz@unine.ch)

P. Renard and J. Straubhaar, Centre for Hydrogeology, University of Neuchâtel, 11 Rue Emile Argand, CP 158, CH-2000 Neuchâtel, Switzerland.