B002
MP Simulations Without Computing MP Statistics

G. Mariethoz* (Stanford University), P. Renard (University of Neuchatel), J. Straubhaar (University of Neuchatel) & J. Caers (Stanford University)

SUMMARY

In recent years, multiple-point simulation has become an invaluable tool to integrate geological concepts in subsurface models. However, due to the high CPU and RAM demand, its use is restricted to relatively small problems with limited structural complexity. Moreover, it only allows for the simulation of univariate fields.

We present an alternative method that produces conditional realizations honoring the high-order statistics of uni- or multivariate training images. It is based on a sampling method introduced by Shannon (1948), strictly equivalent to the original method of Guardiano and Srivastava (1993), but that does not need to compute conditional probabilities and to store them. In the sampling process, we use a distance between data configurations that allows simulating both discrete and continuous variables.

As a result, the simulation algorithm is drastically simplified and has more possibilities. Since nothing is stored, neighborhoods can have virtually any size. Moreover, the neighborhoods are not restricted to a template, making multiple-grids unnecessary. Multivariate data configurations can be considered, allowing to generate realizations presenting a given multivariate multiple-point dependence.

In addition to having virtually no RAM requirement, the method is straightforward to parallelize. Hence it can produce very large and complex realizations.

Introduction

Multiple-point geostatistics (Guardiano and Srivastava 1993; Hu and Chugunova 2008) has become an invaluable tool to integrate geological concepts in subsurface models. The core principle is to infer high-order spatial statistics from a training image (TI), i.e. a grid containing spatial patterns deemed representative of the spatial structures to simulate.

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. in press; Renard 2007; Ronayne, et al. 2008). 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. Moreover, it only allows for the simulation of univariate fields.

In this paper, we present 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 co-simulations (Mariethoz 2009; Mariethoz and Renard 2010; Mariethoz, et al. 2009). 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 modeling a wide variety of connectivity patterns. Furthermore, non-stationarity is a very frequent feature in most real case situations. Therefore, a special effort has been devoted to developing a set of techniques that can be applied when non-stationarity occurs. Because the method can handle co-simulation 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.

Simulation algorithms express multiple-point statistics 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 $\mathbf{x}_\mathrm{i}$ of $\mathbf{x}$:

$$\mathrm{F}(z, \mathbf{x}, \mathbf{d}_n) = \mathrm{Prob}\{Z(\mathbf{x}) \le z \mid \mathbf{d}_n\}. \tag{1}$$

The sequential simulation paradigm is used for simulation. At each successive location, the conditional cumulative distribution function (ccdf) F(z,x, dn) is conditioned to both the previously simulated nodes and the actual data. A value for Z(x) is drawn from the probability distribution and the algorithm proceeds to the next location.

To estimate the non parametric ccdf (1) at each location, Guardiano and Srivasta (1993) proposed entirely scanning the training image at each step of the simulation. The method was inefficient and therefore could not be used in practice. A solution to that problem was developed by Strebelle (2002): the *snesim* simulation method consists in 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 and does not allow capturing large-scale features such as channels. To palliate this problem, Strebelle (2002) introduced multi-grids (or multi-scale grids) to simulate the large-scale structures first, and later the small-scale features. Although multi-grids allow good reproduction at different scales, they generate problems related to the migration of conditioning data at each multi-grid level. Artifacts may appear, especially with large datasets that cannot be fully used on the coarsest multi-grids levels. Since all configurations of pixel values that are found in the TI are stored in the search tree, *snesim* is rapidly 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 3D grids. 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. In addition, to account for non-stationarity 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). One approach to mitigate the memory problem is to store multiple-point statistics in lists instead of tree structures (Straubhaar, et al. submitted; Straubhaar, et al. 2008).

The approaches described above can only deal with categorical variables because of the difficulty to estimate (1) for a continuous variable. Zhang et al. (2006) propose an alternative method where 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.

In this paper, we adopt the point of view that sampling 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, conditionally 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 from Shannon (1948) who produced Markovian sequences of random English by drawing letters from a book conditionally to previous occurrences.

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 used in the simulation. Choosing an appropriate measure of distance allows dealing with either categorical or continuous variables, and to accommodate complex multivariate problems such as correlation between categorical and continuous variables.

The Direct Sampling algorithm

The aim of the Direct Sampling method is to simulate a random function Z(x). The input data are a simulation grid (SG) whose nodes coordinates are denoted x, a training image (TI) whose nodes coordinates are denoted 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 Figure 1 and Figure 2 and proceeds as follows.

1) Each conditioning data is assigned to the closest grid node in the SG.
2) Define a path through the remaining nodes of the SG.
3) For each successive location **x** in the path:
   a) Find the neighbors of **x**. They consist of a maximum of the n closest grid nodes $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ that were already assigned or simulated in the SG.
   b) Compute the lag vectors $\mathbf{L} = \{\mathbf{h}_1, \ldots, \mathbf{h}_n\} = \{\mathbf{x}_1 - \mathbf{x}, ..., \mathbf{x}_n - \mathbf{x}\}$ defining the neighborhood of x, $\mathbf{N}(\mathbf{x}, \mathbf{L}) = \{\mathbf{x} + \mathbf{h}_1, ..., \mathbf{x} + \mathbf{h}_n\}$.
   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.
   d) Define the search window in the TI. It is the ensemble of the locations **y** such that all the nodes **N**(**y**,**L**) are located in the TI.
   e) Randomly draw a location y in the search window and from that location scan systematically the search window. For each location **y**:
      i) Find the data event $\mathbf{d}_n(\mathbf{y}, \mathbf{L})$ in the training image.
      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.

iii) Store **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.

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 $Z(\mathbf{x})$.

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

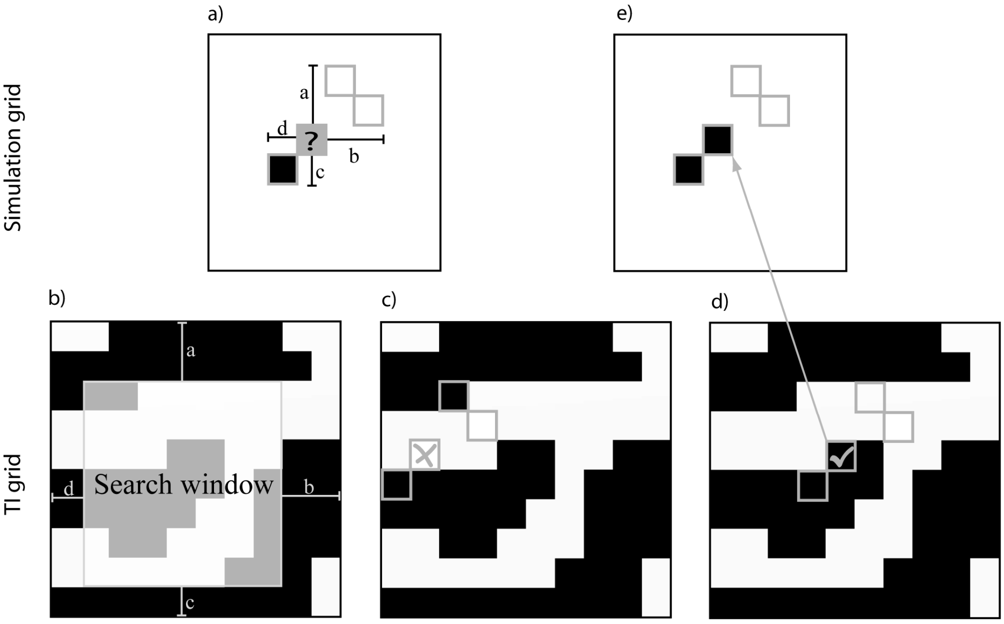

Figure 1: Illustration of the Direct Sampling (DS) method. a) Define the data event in the simulation grid. The question mark represents the node to be simulated. The 2 white and the black pixels represent nodes that have been previously simulated. b) Define a search window in the TI grid by using the 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.

The definition of the data event by considering the *n* closest informed grid nodes is very convenient as it allows for 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.

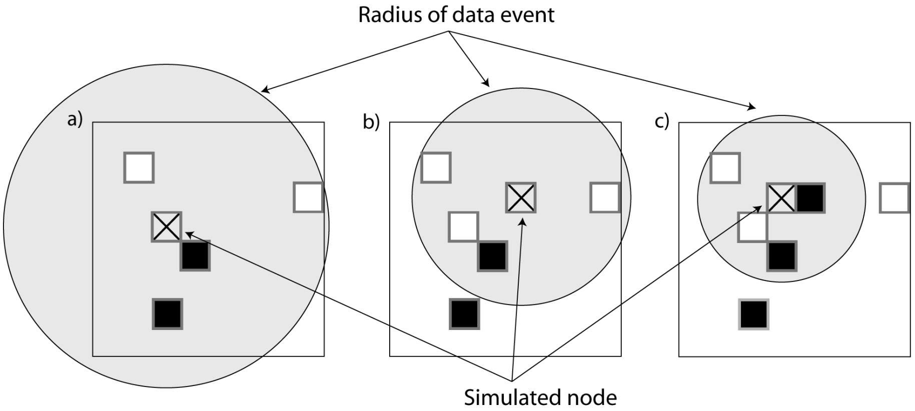

Figure 2: Illustration of the natural reduction of the data events size. The neighborhoods for simulating three successive grid nodes a), b) and c) are defined as the 4 closest grid nodes. As the grid becomes more densely informed, the data events become smaller.

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 non-matching nodes in the data event, given by the indicator variable *a* that equals 0 if two nodes have identical value and 1 otherwise:

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

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 hi using a power function of order *δ*:

$$d\{\mathbf{d}_n(\mathbf{x}),\mathbf{d}_n(\mathbf{y})\}=\frac{\sum_{i=1}^{n}a_i\|\mathbf{h}_i\|^{-\delta}}{\sum_{i=1}^{n}\|\mathbf{h}_i\|^{-\delta}} \quad \in[0,1], \quad where\ a_i=\begin{cases}0 & if\ Z(\mathbf{x}_i)=Z(\mathbf{y}_i)\\ 1 & if\ Z(\mathbf{x}_i)\neq Z(\mathbf{y}_i)\end{cases}. \tag{3}$$

Specific weights can be defined if some of the data event nodes are conditioning data, as described in 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 *δ* = 0 (i.e. all nodes of the data event have the same importance), which generally gives good results. Nevertheless, adjusting *δ* may allow obtaining images more representative of the TI while reducing CPU time.

For continuous variables, we propose to use a weighted Euclidian distance:

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

where

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

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.

The numerator in α_i allows weighting 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 a high threshold, whereas it can be more tedious without normalization).

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,

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

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 (Mariethoz 2009), we show how custom distances can be defined for specific problems.

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.

Simulation of a continuous variable

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 allows bypassing this 2-steps approach to generate continuous variables fields presenting realistic connectivity patterns.

Figure 3 shows a simulation using a TI borrowed from Zhang, et al. (2006), with continuous variable and 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.

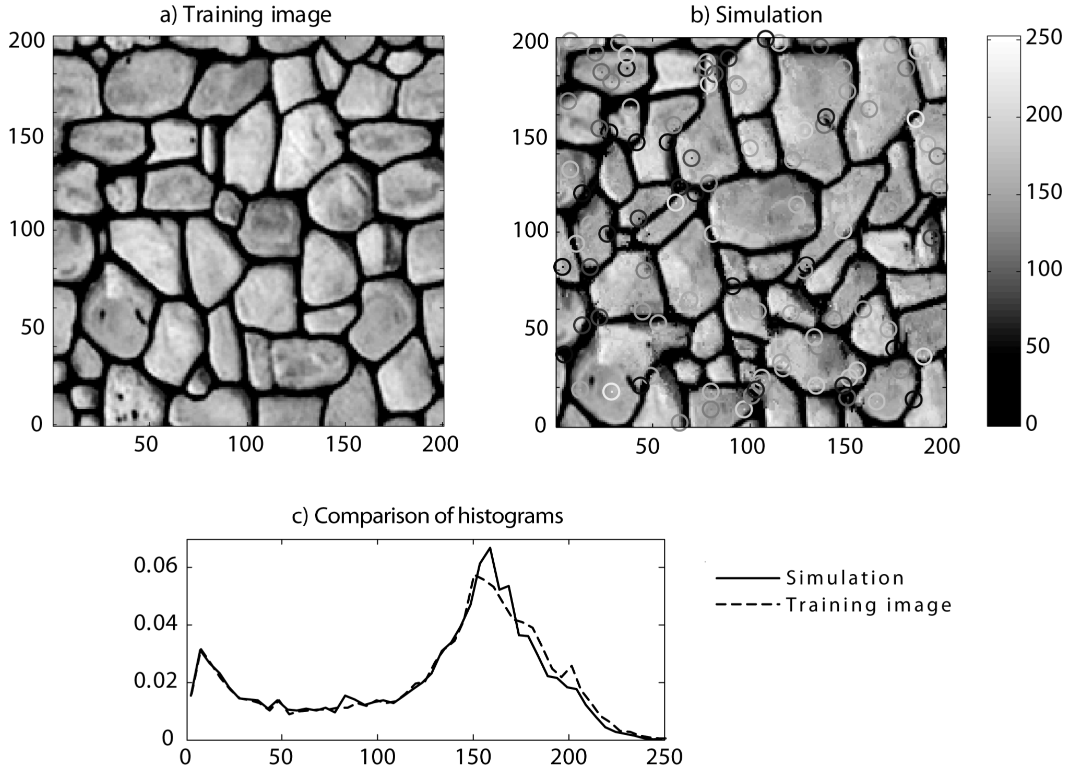

Figure 3: Illustration of the method using a continuous variable. a) Training image with continuous variable. b) One simulation using the unilateral path with 100 randomly located conditioning data (n=80, t=0.01). Positions of conditioning data are marked by circles whose colors indicate the values of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

This example shows that the DS method is able to simulate complex fields of continuous variables while constraining properties such as the statistical distribution and the connectivity patterns. Therefore, the method allows determining complex types of heterogeneity that control the flow and transport behavior of the model.

Multivariate case

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. This allows defining the data events through a set of variables that can be simulated jointly or used for conditioning following the same principle as co-simulation (collocated or not). The training image is a multivariate field comprising m variables $Z_1(\mathbf{x}),\ldots, Z_m(\mathbf{x})$. 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

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

Each variable Z_k involved in the multivariate analysis can have a different neighborhood and a specific data event $\mathbf{d}^k_{n_k}(\mathbf{x},\mathbf{L}^k)=\{Z_k(\mathbf{x}+\mathbf{h}^k_1),\cdots,Z_k(\mathbf{x}+\mathbf{h}^k_{n_k})\}$. This involves that 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 taken for each variable, as defined previously:

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

The weights w_k are defined by the user. This allows accounting 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.

Figure 4 shows an example of a joint simulation of two variables that are spatially related by some unknown function. For this synthetic example, the TI for variable 1 (Figure 4a) is a binary image representing a channel system. 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 relationship. Figure 4c and Figure 4d show one unconditional bivariate simulation using the TI described above. The categorical variable 1 uses distance (3) 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-points correlation. Note that addressing correlations 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 scatter-plot (e.g. cross-variograms).

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** (Chugunova and Hu 2008; Straubhaar*, et al.* submitted), DS accounts for complex spatial patterns of the secondary variable because multiple-point statistics are considered for both primary and secondary variables.

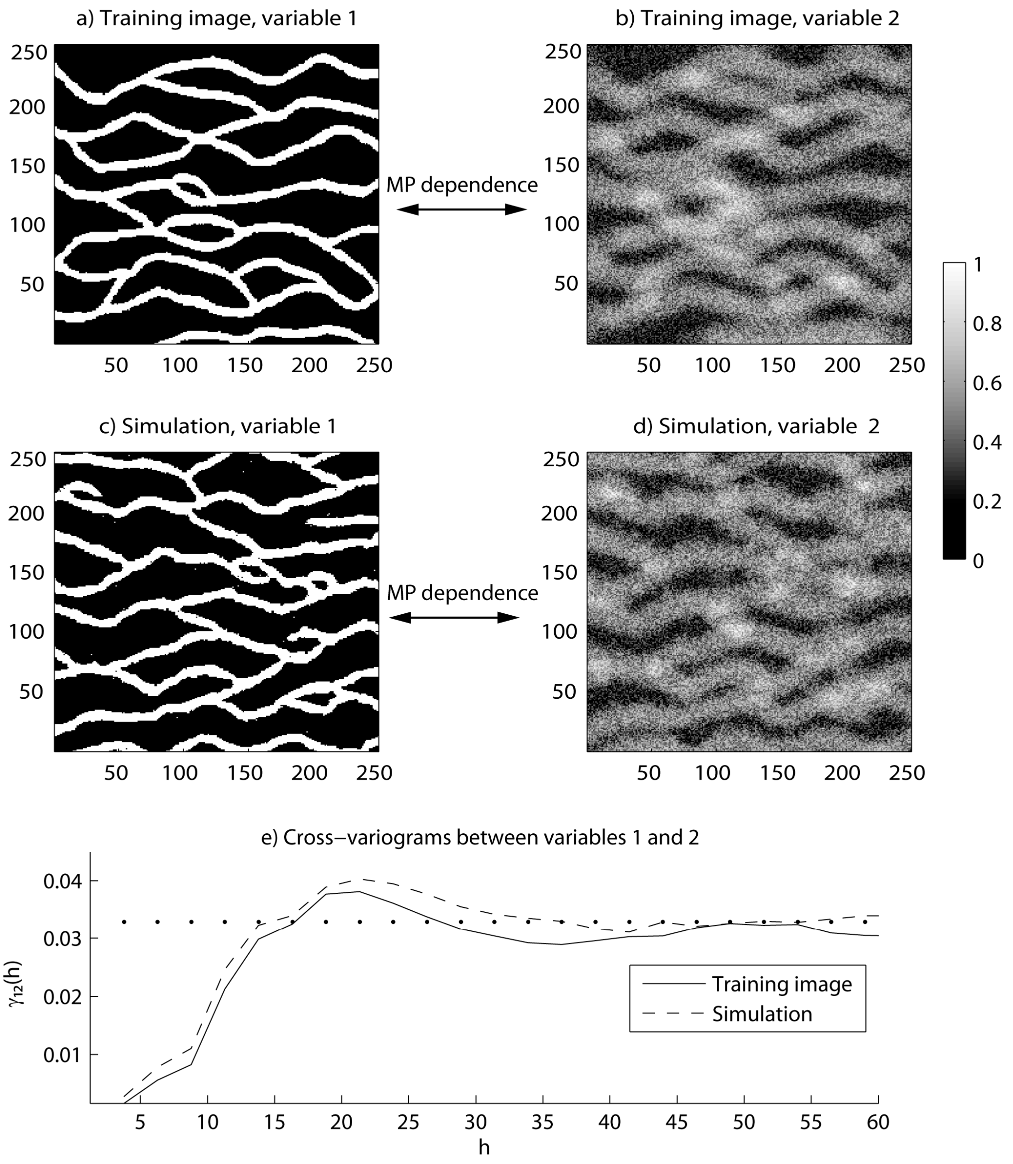

Figure 4: Joint simulation of two variables (n_1=30, n_2=30, t=0.01, w_1=0.5, w_2=0.5). a) and b) The bivariate training image, with a complex multiple-point dependence. c) and d) One resulting bivariate simulation, where the MP dependence is reproduced. e) Cross-variograms reproduction. Note that no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

When one (or several) of the joint variables is already known, DS uses this information as conditioning data on the secondary variable, guiding the simulation of the primary variable and then reducing uncertainty. The relation between both variables, possibly complex and not necessary linear, is given via the bivariate TI. 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 primary variable and then use a forward model to simulate the secondary variable.

Discussion and conclusion

In this paper, we present the Direct Sampling (DS) simulation method. As compared to traditional multiple point techniques such as *snesim*, 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 multi-grids. 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 fails 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 problem 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 discrepancies between the TI and the simulation. Carrying this idea further, one can argue that such discrepancies are necessary in a modeling context. Perfectly honoring all patterns of the TI at all scales would involve that the simulation algorithm reproduces every single pixel in the TI. This is certainly not a modeling objective.

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 non-stationary. This can be extremely powerful when realistic geological structures must be modeled, as is commonly the case for transport problems that are strongly influenced by heterogeneity and connectivity of the geological structures. 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 non-parametric 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.

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. Therefore the size of the neighborhood is not limited by memory considerations. 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 of nodes *n*. In certain cases, it can be useful to limit the radius of the data events, for example, when considering non-stationary variables, to avoid capturing non-stationarity 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 towards reproducing the statistical properties of a small central portion of the TI.

A related advantage of the DS approach is that multi-grids (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 multi-grids avoids problems related to the migration of conditioning data on coarse multi-grid levels. By avoiding multi-grids, DS is easy to implement, easy to parameterize and has no problems accommodating large datasets.

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. Moreover, using parallelization allows easily increasing the performance of DS.

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

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.

References

Aitokhuehi, I., and L. J. Durlofsky [2005], Optimizing the performance of smart wells in complex reservoirs using continuously updated geological models, *Journal of Petroleum Science and Engineering*, **48**(3-4), 254-264.

Arpat, B., and J. Caers [2007], Conditional Simulations with Patterns, *Mathematical Geology*, **39**(2), 177-203.

Caers, J. [2005], *Petroleum Geostatistics*, Society of Petroleum Engineers, Richardson.

Caers, J., et al. [2003], Stochastic integration of seismic data and geologic scenarios: a West Africa submarine channel saga, *The Leading Edge*, **22**(3), 192-196.

Chugunova, T., and L. Hu [2008], Multiple-Point Simulations Constrained by Continuous Auxiliary Data, *Mathematical Geosciences*, **40**(2), 133-146.

Daly, C. [2004], Higher order models using entropy, Markov random fields and sequential simulation, paper presented at *Geostatistics Banff 2004*, Kluwer Academic Publisher, Banff, Alberta.

El Ouassini, A., et al. [2008], A patchwork approach to stochastic simulation: A route towards the analysis of morphology in multiphase systems, *Chaos, Solitons and Fractals*, **36**(2008), 418-436.

Feyen, L., and J. Caers [2006], Quantifying geological uncertainty for flow and transport modelling in multi-modal heterogeneous formations, *Advances in Water Resources* **29**(6), 912-929.

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

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, *Journal of Petroleum Science and Engineering*, **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).

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

Liu, Y., et al. [2004], Multiple-point simulation integrating wells, three-dimensional seismic data, and geology, *AAPG Bulletin*, **88**(7), 905-921.

Mariethoz, G. (2009), Geological stochastic imaging for aquifer characterization, 232 pp, Ph.D. Dissertation, University of Neuchâtel, Neuchâtel, Switzerland.

Mariethoz, G. [2010], A general parallelization strategy for random path based geostatistical simulation methods, *Computers & Geosciences*.

Mariethoz, G., and P. Renard [2010], Reconstruction of incomplete data sets or images using Direct Sampling, *Mathematical Geosciences*, **42**(3), 245-268.

Mariethoz, G., et al. [2009], Integrating collocated auxiliary parameters in geostatistical simulations using joint probability distributions and probability aggregation, *Water Resour. Res.*, **45**(W08421).

Michael, H., et al. [in press], Combining geologic-process models and geostatistics for conditional simulation of 3-D subsurface heterogeneity, *Water Resour. Res.*

Pickard, D. [1980], Unilateral Markov fields, *Advances in Applied Probability*, **12**655-671.

Renard, P. [2007], Stochastic hydrogeology: what professionals really need?, *Ground Water*, **45**(5), 531-541.

Ronayne, M., et al. [2008], Identifying discrete geologic structures that produce anomalous hydraulic response: An inverse modeling approach, *Water Resour. Res.*, **44**(W08426).

Shannon, C. E. [1948], A mathematical theory of communication, *The Bell system technical journal*(27), 379-423.

Straubhaar, J., et al. [submitted], An improved parallel multiple-point algorithm, *Mathematical Geosciences*.

Straubhaar, J., et al. [2008], Optimization issues in 3D multipoint statistics simulation, paper presented at *Geostats 2008, 1-5 Dec. 2008*, Santiago, Chile.

Strebelle, S. [2002], Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics, *Mathematical Geology*, **34**(1), 1-22.

Strebelle, S., et al. [2003], Modeling of a deepwater turbidite reservoir conditional to seismic data using principal component analysis and multiple-point geostatistics, *SPE Journal*, **8**(3), 227-235.

Zhang, T., et al. [2006], Filter-Based Classification of Training Image Patterns for Spatial Simulation, *Mathematical Geology*, **38**(1), 63-80.