Spatial Statistics 16 (2016) 53–71

Contents lists available at ScienceDirect

Spatial Statistics

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

Conditioning multiple-point statistics simulations to block data

Julien Straubhaar [a,*], Philippe Renard [a], Grégoire Mariethoz [b]

[a] The Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel, rue Emile-Argand 11, CH-2000 Neuchâtel, Switzerland

[b] Institute of Earth Surface Dynamics (IDYST), University of Lausanne, UNIL-Mouline, Geopolis, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 17 August 2015
Accepted 6 February 2016
Available online 17 February 2016

Keywords:
Geostatistical simulation
Multiple-point statistics
Block data
Downscaling

ABSTRACT

Multiple-points statistics (MPS) allows to generate random fields reproducing spatial statistics derived from a training image. MPS methods consist in borrowing patterns from the training set. Therefore, the simulation domain is assumed to be at the same resolution as the conceptual model, although geometrical deformations can be handled by such techniques. Whereas punctual conditioning data corresponding to the scale of the grid node can be easily integrated, accounting for data available at larger scales is challenging. In this paper, we propose an extension of MPS able to deal with block data, *i.e.* target mean values over subsets of the simulation domain. Our extension is based on the direct sampling algorithm and consists to add a criterion for the acceptance of the candidate node scanned in the training image to constrain the simulation to block data. Likelihood ratios are used to compare the averages of the simulated variable taken on the informed nodes in the blocks and the target mean values. Moreover, the block data may overlap and their support can be of any shape and size. Illustrative examples show the potential of the presented algorithm for practical applications.

* Corresponding author.
E-mail addresses: julien.straubhaar@unine.ch (J. Straubhaar), philippe.renard@unine.ch (P. Renard), gregoire.mariethoz@unil.ch (G. Mariethoz).

http://dx.doi.org/10.1016/j.spasta.2016.02.005
2211-6753/

1. Introduction

The multiple-point statistics (MPS) methods have become very popular in earth sciences, because they allows to generate highly heterogeneous random fields reproducing the spatial statistics of a conceptual geological model, the training image, given by the user. These methods overcome some limitations of classical geostatistical simulation techniques based on two-point statistics: variogram-based methods such as sequential Gaussian simulation, sequential indicator simulation (*sisim*) (Deutsch and Journel, 1998), transition probability based approaches such as *TProgs* (Carle, 1996), or Markovian-type categorical prediction (MCP) based on a maximum entropy principle (Allard et al., 2011). Among the existing MPS simulation algorithms, *snesim* (Strebelle, 2002) and *impala* (Straubhaar et al., 2011, 2013) successively populate each node of the simulation grid by randomly drawing a facies category according to a probability distribution conditioned to the data event centered at the simulated node, computed from a catalog of patterns found in the training image. A storage based on a tree structure is used in *snesim* ensuring computational time efficiency, and a list-based catalog employed in *impala* guarantees low memory requirements. Using a catalog implies to consider only categorical variables and patterns of fixed geometry. A multiple grid approach (Tran, 1994) is employed in theses algorithms to capture large scale structures while keeping data events of reduced size. On the other hand, the direct sampling algorithm (Mariethoz et al., 2010) is a distance-based MPS algorithm. To simulate a node, the method consists in randomly scanning the training image until the pattern in the training image is compatible with the pattern retrieved from the simulation grid and centered at the simulated node. Then, the central node value is copied and pasted from the training image to the simulation grid. The compatibility between two patterns is related to a distance. This basic simulation principle leads to a very flexible method. Indeed, not using any catalog of patterns, categorical as well as continuous variables can be considered by defining an appropriate distance between data events, and the geometry of the patterns can vary during the simulation allowing to reproduce large scale structures without using a multiple grid approach. In particular, punctual conditioning data can be simply assigned in the simulation grid at the beginning of the simulation, whereas methods based on a multiple grid approach implies some precautions to properly address punctual data (Straubhaar and Malinverni, 2014). Distance-based MPS algorithms include also techniques consisting in pasting patches of the training image in the simulation grid at a time instead of only one pixel value, such as *filtersim* (Zhang and Journel, 2006) or *simpat* (Arpat and Caers, 2007). These latter methods use patterns database built from the training image and are also based on multiple grid approaches. Other patch-based MPS algorithms not using multiple grids nor databases consist in pasting overlapping boxes of pixels along a raster path, by minimizing a cross-correlation function over the overlapping region in the algorithm *ccsim* (Tahmasebi et al., 2012), or by minimizing an error between the common area followed by an optimal cut through this area in the algorithm *conditional image quilting (CIQ)* (Mahmud et al., 2014). Theses methods allow to better model the connectivity of the structures, but make the conditioning difficult. Therefore, the direct sampling method is appealing due to its simplicity and its flexibility. In particular it can easily be extended to the simulation of multivariate fields (Mariethoz et al., 2010, 2012), providing an intuitive tool to manage various types of nonstationarities.

No matter what MPS technique is considered, the simulation domain is filled by borrowing patterns from the training image, which is assumed to have the same resolution as the simulation grid. Hence, whereas punctual conditioning data (corresponding to the scale of the simulation domain) can be straightforwardly handled, conditioning a simulation at local scale with data defined at a larger scale is quite challenging. Classical parametric methods based on covariance models can be used to integrate data with different support sizes (Liu and Journel, 2009; Journel, 1999). Essentially based on cokriging theory, such techniques nevertheless require point-to-point, point-to-block and block-to-block (cross-) covariance models and imply Gaussian assumptions.

In this paper, we propose an extension of the direct sampling algorithm able to deal with block data, *i.e.* target values for the average of the simulated variable on subsets of the simulation domain. The principle is to use the block data as a criterion for accepting a candidate location in addition to the comparison of the patterns. The current averages accounting for the already simulated pixels in the blocks plus the candidate value are compared to the target mean values and then a related mismatch for each block data is computed. Indeed, we follow a similar strategy as for multivariate simulations

(Mariethoz et al., 2010), where a mismatch (distance) between the training and simulated patterns is computed for each variable. As several mismatches are computed, the condition for accepting a candidate node has to be defined. One can define a global misfit as a weighted average of the considered mismatches and check if it is below a certain threshold, as in the original direct sampling algorithm. This requires to specify a weight per mismatch and a global threshold. Another option is to specify a threshold per compared quantity, and accept the candidate value when each mismatch is less than its corresponding threshold. To develop our extension, we use an implementation of the direct sampling, called *DeeSse* (Straubhaar, 2015), based on this latter alternative.

It is important to highlight the differences between block data and local probability constraints. Local probability constraints are classically given by probability maps, which gives at any point of the simulation domain the probability of having a given category (facies) in a certain neighborhood. These maps are related to an underlying fixed support size, and represent moving local probabilities or proportions. Such constraints are designed for categorical variables and are handled in classical techniques (Liu, 2006; Krishnan, 2008) by using probability aggregation methods (Allard et al., 2012). Mariethoz et al. (2015) proposed a method to constrain distance-based MPS methods, such as the direct sampling algorithm, to such local probability constraints. On the other hand, although block data for binary variables can be viewed as proportion constraints, block data are defined on fixed groups of nodes of any shape, on which target mean values are given. Moreover, for block data conditioning to make sense, the simulated variable can be either continuous or discrete, provided that it does not represent arbitrary category codes.

Downscaling or increasing the resolution of an image is a classical example where the support of the data (input coarse scale image) is larger than the resolution of the output image. Mariethoz et al. (2011) developed a super-resolution method which consists in using the direct sampling algorithm to generate small-scale structures from the patterns found in the coarse scale (training) image. This tools assumes that the spatial structures have a property of scale invariance also called fractal property. Tang et al. (2015) propose to downscale remotely sensed images accounting for available images at different resolutions, using *filtersim* as MPS algorithm at coarse scale, combined with an area-to-point cokriging method to integrate the fine scale information.

More generally, the algorithm presented in this paper for MPS simulation accounting for block data can be used in a range of applications, not restricted to downscaling, where a conceptual model for the fine scale is known and the available conditioning data are defined on support of varying size and shape, larger than the resolution of the simulation domain. In its current implementation, the proposed method assumes that the block data are defined as the arithmetic mean of the values located within a block of known geometry. In practice and in certain situations, the block values are defined in a more complex manner, especially for non-additive variables such as permeability. Nevertheless, our method can still be useful to generate simulations that offer an approximation of the fine-scale structures.

This paper is organized as follows. In Section 2, some background information on the direct sampling algorithm is given. Then, in Section 3, block data are defined and we present how the algorithm is extended to account for block data. The manner the block data constraints are treated is presented in detail in Section 4. In Section 5, a multiGaussian test case is presented. MultiGaussian simulation offers a well-known framework, where the random fields are entirely described by an analytical covariance model which is used for the simulation. Hence, expected results are known and provide a point of comparison to study the performances of the proposed method. Then, application examples displaying more complex structures and justifying the use of MPS are given in Section 6. As illustrations, we apply the proposed method to simulate log-permeability fields in a downscaling context and in a situation where conditioning data are available at three different scales. In another synthetic example, we propose to model the subsurface geology conditionally to geophysical data. Finally, the method is discussed in Section 7.

2. Basic direct sampling algorithm

The basic principle of the direct sampling technique (Mariethoz et al., 2010) for simulating a node x in the simulation grid is to compare the pattern (or data event) $d(x)$ centered at x with patterns of

same geometry $d(y)$ centered at random locations in the training image (TI), until $d(x)$ and $d(y)$ are sufficiently similar. For univariate simulation, the direct sampling algorithm requires in input a TI with a unique variable Z, a simulation grid (SG), a normalized distance D used for comparing two patterns, and the three following parameters:

- n: maximal size of the patterns;
- t: threshold: two patterns are considered similar if the distance between them is below t;
- f: maximal scan fraction f of the TI for the simulation of each node.

The algorithm consists in simulating the variable Z in the SG as follows. First, conditioning punctual data (if present) are assigned in the SG and a random path visiting all non informed nodes in the SG is defined. Then, each node x along this path is simulated by applying the steps (a–d) below.

(a) Retrieve the pattern

$$d(x) = \{Z(x + h_1), \ldots, Z(x + h_n)\} \tag{1}$$

in the SG, made up of the maximal n closest informed neighbors of x.

(b) Set $E_{cur} = \infty$ (best current error), $y_{cur} = NA$ (best current candidate), and $f_{cur} = 0$ (current scanned fraction of the TI).

(c) While $E_{cur} > 0$ and $f_{cur} < f$ do:

(i) Sample randomly a location y in the TI (not already visited during the while loop).

(ii) Retrieve the pattern $d(y) = \{Z(y + h_1), \ldots, Z(y + h_n)\}$ in the TI, and compute the error

$$E = \max\left(0, \frac{D(d(x), d(y)) - t}{t}\right). \tag{2}$$

(iii) If $E < E_{cur}$, then set $E_{cur} = E$ and $y_{cur} = y$.

(iv) If $E_{cur} = 0$, exit the while loop.

(v) Update the current scanned fraction f_{cur} (by adding the inverse of the number of nodes in the TI).

(d) Assign $Z(x) = Z(y_{cur})$.

According to the definition (2) of the error E, we have $E = 0$ if and only if $D(d(x), d(y)) \leqslant t$. In that case, the scan of the TI is interrupted (step (iv)). If this condition is not reached, the best candidate y_{cur} (*i.e.* giving the smallest error) met so far is retained (step (iii)). Moreover, note that decreasing the maximal scan fraction f allows to save computational time, and specifying $f < 1$ (*i.e.* excluding an exhaustive scan of the TI) can be useful to avoid "verbatim copy", *i.e.* exact copies of part of the TI (Meerschman et al., 2013).

The distance D is normalized such that $D(d(x), d(y))$ falls in the interval $[0, 1]$ for any pair of patterns, so that the threshold t required in input should be in $]0, 1]$ in any situation. The definition of the distance depends on the type of the simulated variable. Typically, if Z is a categorical variable, the distance can be defined as the proportion of mismatching nodes,

$$D(d(x), d(y)) = \frac{1}{n}\sum_{i=1}^{n} a_i, \quad \text{with } a_i = \begin{cases} 0 & \text{if } Z(x + h_i) = Z(y + h_i), \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

For a continuous variable Z, one can use the Manhattan distance

$$D(d(x), d(y)) = \frac{1}{\left(\max\limits_{y \in TI} Z(y) - \min\limits_{y \in TI} Z(y)\right)} \cdot \frac{1}{n}\sum_{i=1}^{n} |Z(x + h_i) - Z(y + h_i)|. \tag{4}$$

Note that conditional punctual data, if present, are supposed to be within the range of values present in the TI.

3. Algorithm accounting for block data

Initially designed for continuous variables, block data can be considered for categorical variables as well. In this situation, a unique numerical value is assigned to each category. These values should have a physical signification, so that they can be arranged in order. Under these conditions, the distance (3) can still be used for comparing patterns.

Considering a variable Z in the SG, a block data is defined by a triplet (B, μ_B, t_B) where:

- B is a set of nodes (pixels) in the SG,
- μ_B is a target value for the mean of the variable Z over the nodes in B,
- t_B is a tolerance corresponding to the half length of a target interval $[a_B, b_B]$ containing μ_B: the block data is considered as honored if the mean of Z on B is within this interval.

The target interval $[a_B, b_B]$ is defined in Section 4.1. Defining the tolerance at left and the tolerance at right respectively as

$$t_{B,left} = \mu_B - a_B \geqslant 0, \tag{5}$$

$$t_{B,right} = b_B - \mu_B \geqslant 0, \tag{6}$$

the target interval can be written $[\mu_B - t_{B,left}, \mu_B + t_{B,right}]$. Note that by definition, $t_{B,left} + t_{B,right} = 2t_B$.

Considering a set of block data, the aim is to simulate the variable Z in the SG such that the spatial structures present in the TI are reproduced and each block data respected. The idea is to adapt the direct sampling algorithm above, by adding a condition related to the block data constraints to interrupt the scan of the TI, *i.e.* exit the while loop (step iv). During the scan of the TI for the simulation of a node x in the SG, the current mean over each block containing x is computed accounting for the informed nodes which are already simulated (or punctual data) within each block and for the candidate value $Z(y)$ at x. For each block B containing x, the current mean is compared to the target mean value given in input and an error E_B according to the specified tolerance is computed. The error E (Eq. (2)) in step (ii) of the algorithm presented in the previous section is then updated by adding each of these errors. More precisely, the following step (ii') is inserted after the step (ii).

(ii') For each block B containing the node x:

- Compute the current mean on the block B accounting for the candidate node y,

$$\mu_B^*(y) = \frac{1}{k}\left(\sum_{i=1}^{k-1} Z\left(x_B^{(i)}\right) + Z(y)\right), \tag{7}$$

  where $\left\{x_B^{(1)}, \ldots, x_B^{(k-1)}\right\}$ are the set of informed nodes in block B.
- Compute an error for the constraint on block B,

$$E_B = E_B\left(\mu_B, \mu_B^*(y), t_{B,left}, t_{B,right}\right) \tag{8}$$

  depending on the target mean value μ_B and the tolerances at left $t_{B,left}$ and at right $t_{B,right}$ derived from t_B.
- Then, update the error E:

$$E = E + E_B. \tag{9}$$

The error E_B is defined (Section 4.2) such that it is a positive value and vanishes if the current mean $\mu_B^*(y)$ is within the target interval $[\mu_B - t_{B,left}, \mu_B + t_{B,right}]$. Thus, the scan of the TI is stopped in step (iv) once $D(d(x), d(y)) \leqslant t$ and $E_B = 0$ for each block containing x. Note that during a simulation, the means on every blocks are stored and simply updated after the simulation of each node. Hence the computation of $\mu_B^*(y)$ is very fast.

It is worth to notice that following this principle allows to deal with several specific constraints: a global error expressed as a sum of specific errors is computed, reflecting the desired conditions. For example, for multivariate simulation, an error related to the comparison of the patterns for each variable is defined similarly to (2), each variable having their own distance type, data event and threshold. Block data can then be considered in multivariate simulation, and each variable can be constrained by block data. The tolerance related to a block data and the threshold applied to the comparison of patterns play a similar role and allow to give more or less importance to the

corresponding constraints. This strategy for dealing with multiple constraints (*e.g.* patterns similarity or block data) and consisting in computing an error for each of them is implemented in a version of the direct sampling technique called *DeeSse* (Straubhaar, 2015). Then, the condition for accepting a candidate scanned node in the TI is that the global error, defined as the sum of the specific errors, becomes zero, which means that all constraints are honored. The original version of the direct sampling algorithm (Mariethoz et al., 2010) differs from *DeeSse* in that a unique threshold is applied to a global misfit expressed as a weighted average of the misfit (or distance) related to each constraint. In any case, the principle of the block conditioning methodology remains identical with both types of implementations.

4. Defining the target interval and the error for a block data constraint

In this section, the error E_B (8) required by the method is defined in detail.

4.1. Target interval for block data

First, the target interval $[a_B, b_B] = [\mu_B - t_{B,left}, \mu_B + t_{B,right}]$ has to be defined. The rationale to not take a symmetric interval around μ_B (*i.e.* $t_{B,left} = t_{B,right} = t_B$) is that it would result in a bias regarding the difference between the *a posteriori* average value (in the simulation) and the target mean value μ_B: the expectation of this difference would not be equal to zero.

Following a rejection scheme, suppose that simulations of a variable Z are performed without accounting for the block data constraint (B, μ_B, t_B), and then only those for which the average of Z on the block B is within the target interval $[a_B, b_B] = [a_B, a_B+2t_B]$ are retained. The idea is to determine a_B such that the mean of the average values on B considering the retained simulations is equal to μ_B. For that, we estimate the distribution of the mean value M on the block B (for simulation not conditioned to block data) by the normal law

$$M \sim \mathcal{N}\left(m, s^2\right), \tag{10}$$

where the "block mean" m and the "block variance" s^2 are estimated as

$$m = \frac{1}{N_{TI,B}} \sum_{i=1}^{N_{TI,B}} \mu_i \quad \text{and} \quad s^2 = \frac{1}{N_{TI,B} - 1} \sum_{i=1}^{N_{TI,B}} (\mu_i - m)^2, \tag{11}$$

μ_i being the mean on the block $B_{TI,i}$ and $\left\{B_{TI,1}, \ldots, B_{TI,N_{TI,B}}\right\}$ the set of all blocks of same geometry as B and included in the TI. Let X_a be the random variable constructed as M restricted on the interval $[a, a + 2t_B]$; its density function is defined by

$$f_{X_a}(x) = \frac{f_M(x)}{F_M(a + 2t_B) - F_M(a)}, \quad a \leqslant x \leqslant a + 2t_B, \tag{12}$$

where f_M and F_M are respectively the density and the cumulative distribution function of the Gaussian random variable M. The left bound a_B of the target interval is then chosen such that the mean of X_{a_B} is equal to μ_B (Fig. 1), *i.e.* a_B is the zero of the function

$$\begin{aligned} h(a) &= \mathbb{E}\left(X_a\right) - \mu_B \\ &= \left(F_M(a + 2t_B) - F_M(a)\right)^{-1} \cdot \int_a^{a+2t_B} x f_{X_a}(x) \mathrm{d}x - \mu_B \\ &= m - \mu_B + \frac{s\left(\exp\left(-\frac{1}{2}\left(\frac{a-m}{s}\right)^2\right) - \exp\left(-\frac{1}{2}\left(\frac{a+2t_B-m}{s}\right)^2\right)\right)}{\sqrt{2\pi}\left(F_M(a + 2t_B) - F_M(a)\right)}. \end{aligned} \tag{13}$$

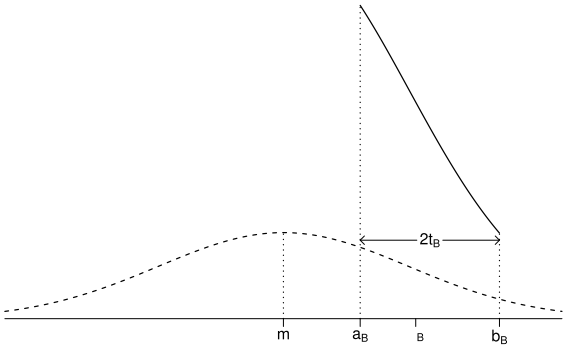


Fig. 1. Illustration of the target interval for a block data; the dashed line is the density function f_M of $M \sim \mathcal{N}\left(m, s^2\right)$, with m the block mean, and s^2 the block variance (estimated from the TI); the solid line is the density function f_X of X defined on $[a_B, b_B]$, f_X is proportional to f_M; the bounds a_B and b_B are set such that $b_B - a_B = 2t_B$ and $\mathbb{E}(X) = \mu_B$, where μ_B and t_B are the target mean and the tolerance for the block B.

Note that the cumulative distribution function of a Gaussian law can be expressed with the error function erf,

$$F_M(x) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{x-m}{\sqrt{2}s}\right)\right), \qquad \mathrm{erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^t e^{-u^2}\mathrm{d}u. \tag{14}$$

As h is increasing and $h(\mu_B - 2t_B) < 0 < h(\mu_B)$, the zero of h can be easily computed by using the bisection method.

This manner to define the target interval supposes that the mean value on blocks of same geometry as B in the TI follows a Gaussian distribution, which is a reasonable assumption for stationary TI. When the target mean μ_B for a block B in the simulation domain is far from the block mean value m, the target interval is strongly asymmetric around μ_B and this allows to avoid significant biases which would appear in such cases by taking a symmetric interval around μ_B.

4.2. Error for block data

Once the target interval is known, we have to define the error E_B (8) quantifying how far from the target interval is the current mean on the block B. By the central limit theorem, we know that the mean $\overline{Z}_k$ of k independent and identically distributed random variables of mean μ and variance σ^2 approximately follows a normal law of same mean and variance σ^2/k, whose probability distribution function is proportional to

$$L_{\mu,\sigma^2,k}(z) = \exp\left(-\frac{k}{2\sigma^2}(z-\mu)^2\right). \tag{15}$$

This likelihood function can be used to evaluate the probability that a mean z of k values is equal to μ. Assuming that a value $z \in [\mu - t_{left}, \mu + t_{right}]$ is acceptable, the likelihood ratio

$$R_{\mu,\sigma^2,k,t_{left},t_{right}}(z) = \begin{cases} L_{\mu,\sigma^2,k}(\mu - t_{left})/L_{\mu,\sigma^2,k}(z), & \text{if } z \le \mu \\ L_{\mu,\sigma^2,k}(\mu + t_{right})/L_{\mu,\sigma^2,k}(z), & \text{if } z > \mu \end{cases} \tag{16}$$

i.e.

$$R_{\mu,\sigma^2,k,t_{left},t_{right}}(z) = \begin{cases} \exp\left(-\frac{k}{2\sigma^2}\left(t_{left}^2 - (z-\mu)^2\right)\right), & \text{if } z \le \mu \\ \exp\left(-\frac{k}{2\sigma^2}\left(t_{right}^2 - (z-\mu)^2\right)\right), & \text{if } z > \mu \end{cases} \tag{17}$$

gives a rate less than or equal to 1 for acceptable cases, and a rate greater than 1 otherwise, which can be used to quantify the "misfit" between z and μ.

For a given block data (B, μ_B, t_B), the tolerance at left $t_{B,left}$ and the tolerance at right $t_{B,right}$ are computed (Section 4.1), and the idea is to use this ratio to express the error E_B. In Eq. (17), μ is set to μ_B, t_{left} and t_{right} are set to $t_{B,left}$ and $t_{B,right}$ respectively, and k to the number of nodes used for computing the current mean on the block. It remains to define the variance involved in the likelihood. A local variance $\sigma^2_{TI,B}$ is computed such that it reflects the variance of the values within the blocks in the TI having the same geometry as B and a mean close to μ_B. Considering all blocks $B_{TI,1}, \ldots, B_{TI,N_{TI,B}}$ of same geometry as B and included in the TI, the mean μ_i and the standard deviation σ_i of the values within the block $B_{TI,i}$ is computed for each i. Indeed, these pairs (μ_i, σ_i) form a sample of an unknown bivariate distribution (M, S), for which we want to get an evaluation $\sigma_{TI,B}$ of the conditional expectation $\mathbb{E}(S|M = \mu_b)$. An approximation is obtained using the Nadayara–Watson kernel estimator (Demir and Toktamis, 2010)

$$\sigma_{TI,B} = \left(\sum_{i=1}^{N_{TI,B}} K((\mu_b - \mu_i)/h) \right)^{-1} \cdot \sum_{i=1}^{N_{TI,B}} K((\mu_b - \mu_i)/h)\sigma_i, \tag{18}$$

with the Gaussian kernel $K(t) = 1/\sqrt{2\pi} \cdot e^{-t^2}$, and the fixed bandwidth, automatically computed according to the "Silverman's rule of thumb" (Sheather, 2004),

$$h = 0.9 \min \left(\sqrt{\mathrm{Var}\left(\{\mu_i\}_{i=1}^{N_{TI,B}}\right)}, \mathrm{IQR}\left(\{\mu_i\}_{i=1}^{N_{TI,B}}\right) \Big/ 1.34 \right) \cdot N_{TI,B}^{-1/5}, \tag{19}$$

where Var is the variance, IQR is the interquartile range (*i.e.* the difference between the 75% and 25% quantiles), and $N_{TI,B}$ is the size of the sample. Note that an adaptive bandwidth could be used by considering $h(\mu_i)$ instead of h in (18) for a more accurate estimation, as proposed by Demir and Toktamis (2010) for example, but this would imply additional computation to determine the bandwidth at each point of the sample. The curve for $\sigma_{TI,B}$ as a function of μ_B is illustrated on an example in Fig. 8. Note that the computation of the block mean m and the block variance s^2 (11), and the computation of the within block standard deviation $\sigma_{TI,B}$ (18) require only one scan of the TI with a block of same geometry as B to retrieve the local means μ_i and local standard deviation σ_i. This is done once in a pre-processing step, and to save computational time, one can sample blocks within the TI to achieve these computations instead of considering the exhaustive list of blocks $B_{TI,1}, \ldots, B_{TI,N_{TI,B}}$.

Then, the error E_B in Eq. (8) is defined as

$$E_B = \max\left(0, R_{\mu_B,\sigma^2_{TI,B},k,t_{B,left},t_{B,right}}\left(\mu^*_B(y)\right) - 1\right), \tag{20}$$

where k is the number of nodes (including the candidate node y) used to compute the current mean $\mu^*_B(y)$.

Although the values of the simulated variable on a block are clearly not independent, because they have to respect the spatial statistics within the TI, the local standard deviation $\sigma_{TI,B}$ defined above allows to account for the variability of the values at the scale of the considered block in the TI, and conditionally to the target block mean value. Indeed, considering the variable Z on local areas in the TI where the mean is close to μ_B, the stronger the spatial correlation is, the smaller the local variance will be. Then, in the case where the target interval containing μ_B is not reached, the likelihood ratio (17) and then the error E_B (20) will also be larger. Therefore, when the local variability in the conceptual model is small, the penalization of candidates y (not satisfying the constraint) is large.

Furthermore, using this likelihood ratio approach has the advantage that the error related to block data constraints depends not only on the specified tolerance, but also on the number k of values contributing to the current mean. The error increases when k increases, therefore the simulation of the first nodes in a block will be less penalized, which is appropriate for block data conditioning because the aim is not to force the value of the variable at punctual locations. Note also that a block data constraint is enabled only when the number of informed nodes in the considered block has reached a minimal proportion of the total number of nodes in the block, *i.e.* ignoring the constraint below this threshold. In each example presented in this paper, all block data constraints are always enabled.

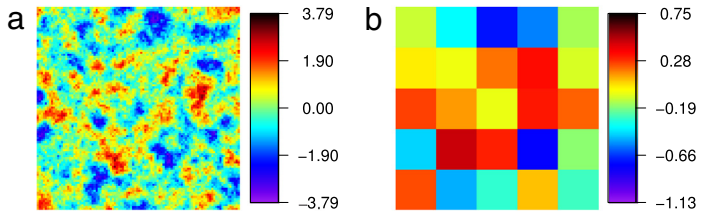


Fig. 2. (a) MultiGaussian simulation of size 100×100 with a mean of 0 and an isotropic spherical covariance with a range of 10 pixels and a variance of 1.0; (b) mean values on blocks of 20×20 pixels computed from field (a) and considered as target block data for the multiGaussian test case.

5. MultiGaussian test case

In this section, we use our method for generating multiGaussian fields conditioned to block data. The aim is to better understand how the method performs in a situation where the theoretical solution is known because it can be expressed analytically. As a point of comparison, we will also use multiGaussian simulations based on all the point-to-point, point-to-block, and block-to-block covariances. The MPS technique is expected to work less efficiently in this situation because it does not use all this information but only the patterns borrowed from an initial TI having a finite size.

For this test, the covariance model is set to the isotropic spherical model with a range of 10 (pixels) and a variance of 1.0, *i.e.* the covariance between two pixels with a distance h is given by $C(h) = 1 - \left(3/2 \cdot h/10 - 1/2 \cdot (h/10)^3\right)$ if $h < 10$ and is equal to 0 for $h \geqslant 10$. The mean of the multiGaussian fields is set to 0. A 100×100 unconditional multiGaussian simulation is displayed in Fig. 2(a), and the corresponding means on blocks of 20×20 pixels covering the entire field shown in Fig. 2(b) are considered as target block data for this test case.

On the one hand, we generate 500 multiGaussian simulations conditioned to the block mean values of Fig. 2(b) based on cokriging. Let $N_P = 10'000$ be the number of points in the simulation grid, $N_B = 25$ the number of blocks, and C_{PP} the $N_P \times N_P$ point-to-point covariance matrix, C_{PB} the $N_P \times N_B$ point-to-block covariance matrix, and C_{BB} the $N_B \times N_B$ block-to-block covariance matrix derived by the covariance model, *i.e.*

$$C_{PP}(i,j) = C(\|x_i - x_j\|), \tag{21}$$

$$C_{PB}(i,l) = \frac{1}{N_B} \sum_{x_j \in B_l} C(\|x_i - x_j\|), \tag{22}$$

$$C_{BB}(k,l) = \frac{1}{N_B^2} \sum_{x_i \in B_k} \sum_{x_j \in B_l} C(\|x_i - x_j\|). \tag{23}$$

To obtain a multiGaussian simulation $Y = \left(Y(x_1), \ldots, Y(x_{N_P})\right)$ conditioned to the block data values $\overline{Y} = \left(\overline{Y}(B_1), \ldots, \overline{Y}(B_{N_B})\right)$, we proceed as follows:

(1) generate an unconditional simulation $Z \sim \mathcal{N}(0, C_{PP})$,
(2) compute the mean on each block $\overline{Z} = \left(\overline{Z}(B_1), \ldots, \overline{Z}(B_{N_B})\right)$, and
(3) update the field Z by setting $Y = Z + C_{PB} \cdot C_{BB}^{-1} \cdot \left(\overline{Y} - \overline{Z}\right)$.

Step (1) is done using circulant embedding of the covariance matrix C_{PP} and Fast Fourier Transform (Wood and Chan, 1994), which provides exact simulations, and step (3) consists in a simple kriging of the residuals on blocks (Dietrich and Newsam, 1996). One simulation resulting from this procedure, and the pixel-wise mean over 500 realizations are displayed in the first column of Fig. 3.

On the other hand, we use the proposed method. First, a $1'000 \times 1'000$ unconditional multiGaussian simulation is generated and is taken as the TI (not shown). The main parameters of the direct sampling algorithm are set to $n = 12$, $t = 0.01$ and $f = 0.1$, with the distance (4) (for continuous variable) (see Section 2), and 500 realizations of size 100×100 conditioned to the block mean values of Fig. 2(b) are generated for the tolerance $t_B = 0.5$, $t_B = 1.0$, and $t_B = 2.0$ (on each block data) respectively. One

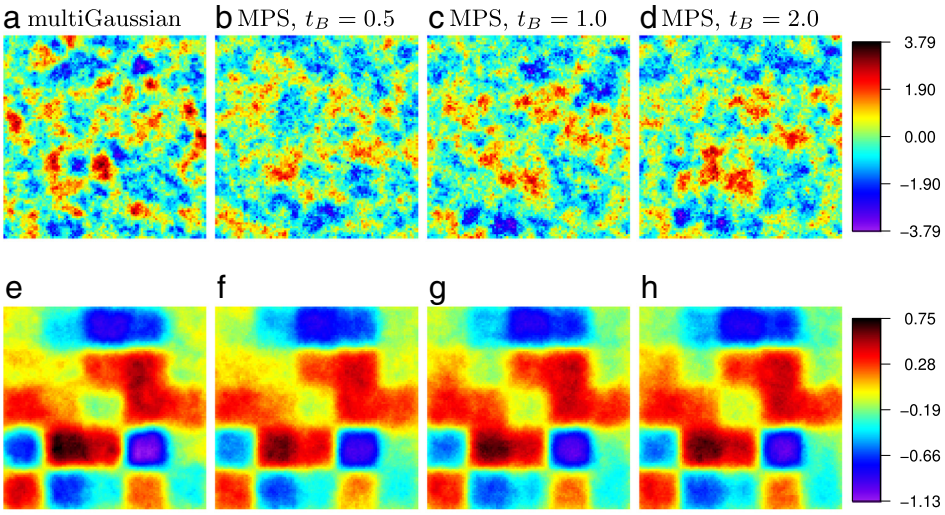


Fig. 3. Results for the multiGaussian test case, using multiGaussian simulation and the proposed MPS method; (1st row) one 100×100 realization conditioned to block data of Fig. 2(b); (2nd row) the pixel-wise mean over 500 realizations; (1st col.) (a,e) multiGaussian simulation; (2nd col.)(b,f) MPS with $t_B = 0.5$; (3rd col.)(c,g) MPS with $t_B = 1.0$; (4th col.)(d,h) MPS with $t_B = 2.0$.

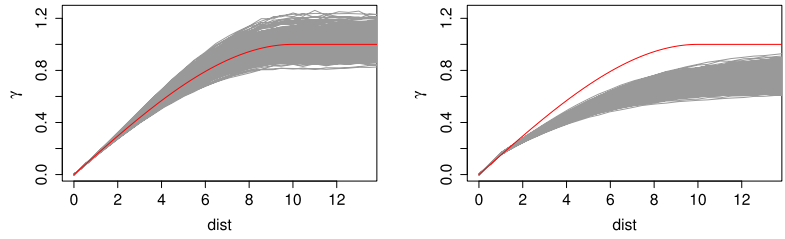
simulation generated with the proposed method, and the pixel-wise mean over 500 realizations are displayed for each tolerance value in the second to fourth columns of Fig. 3.

An experimental (omnidirectional) variogram is computed for each realization of each setup (multiGaussian, MPS with $t_B = 0.5$, $t_B = 1.0$, $t_B = 2.0$), and drawn in Fig. 4 (gray curves). The variogram model (red curve) is given by $\gamma(h) = C(0) - C(h)$, $C(h)$ being the covariance model. The mean value on a block for unconditional multiGaussian simulations follows the normal distribution $\mathcal{N}\left(0, \sigma^2_{BB}\right)$, where σ^2_{BB} is equal to the diagonal coefficients of the block-to-block covariance matrix C_{BB}. Then, in this situation, we can compute the "theoretical" target interval as a function of the target mean, according to Section 4.1 with $m = 0$ and $s^2 = \sigma^2_{BB}$. The differences between the average values on the input blocks for every realization of each setup and the target values (called errors) are displayed in Fig. 5 and we can observe that all these differences are within the theoretical interval $[-t_{B,left}, t_{B,right}]$.

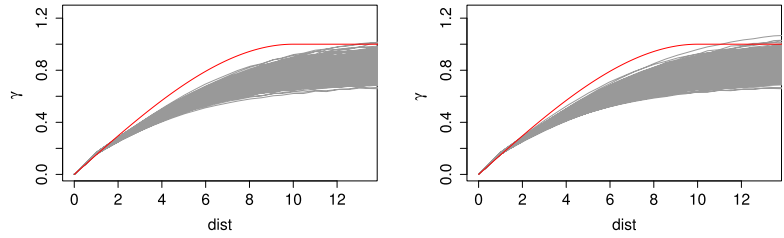
Figs. 3 and 4 show that the proposed method allows to reproduce the structures of the model, but with a reduced point variance (sill of the variogram). The pixel-wise mean maps are all very similar. Increasing the specified tolerance t_B has a small impact on the results. Indeed, for "large" (resp. "small") target mean value μ_B compared to the block mean $m = 0$, increasing t_B results in an increase of $t_{B,right}$ (resp. $t_{B,left}$) whereas only a little change will be observed on $t_{B,left}$ (resp. $t_{B,right}$). Hence, the simulations remain conditioned to block data. Although the sill of the variograms increases a little bit if a larger tolerance is specified, the sill of the covariance model is still greater, which means that conditioning MPS simulations to block data implies a loss of variability at the pixel scale. Note also that a little change of slope at $dist = 1$ can be observed on the variograms of the MPS simulations, which corresponds to a small nugget effect (less than 0.1) not present in the model, and that the slope of these curves seems to stabilize to zero a little further than the range of the model ($dist = 10$). However, each experimental variogram presents a relatively similar shape, depicting the same type of structures. Finally, recall that the proposed MPS method is not designed for a multiGaussian framework, for which an analytical approach is more efficient.

6. Application examples

In this section, the method is applied on synthetic cases, and the computation of the local standard deviation (18) required to handle error on a block data constraint is illustrated.

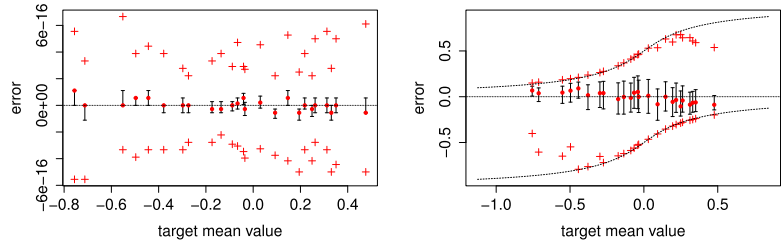


(a) MultiGaussian. (b) MPS, $t_B = 0.5$.

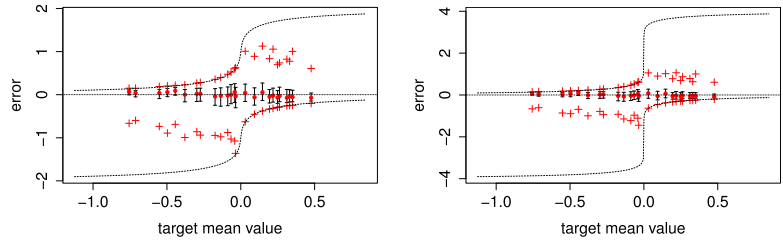


(c) MPS, $t_B = 1.0$. (d) MPS, $t_B = 2.0$.

Fig. 4. Experimental variograms of the 500 realizations (gray curves), for each case of Fig. 3; (a) multiGaussian simulation; (b) MPS with $t_B = 0.5$; (c) MPS with $t_B = 1.0$; (d) MPS with $t_B = 2.0$. In each plot the red curve is the variogram model $\gamma(h) = C(0) - C(h)$ (spherical with a range of 10 pixels and a sill of 1.0). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) MultiGaussian. (b) MPS, $t_B = 0.5$.



(c) MPS, $t_B = 1.0$. (d) MPS, $t_B = 2.0$.

Fig. 5. Differences between observed and target values (Fig. 2(b)) for the mean on each block over 500 realizations, for each case of Fig. 3; (a) multiGaussian simulation; (b) MPS with $t_B = 0.5$; (c) MPS with $t_B = 1.0$; (d) MPS with $t_B = 2.0$. In each plot the target mean values on blocks are in abscissa and the difference "block mean value on simulation minus target mean value" in ordinate; for each block, the median (red point), the interquartile range (black line), and the minimum and maximum (red crosses) over 500 realizations are displayed; the two dashed curves correspond to $-t_{B,left}$ and $t_{B,right}$, the tolerances at left and at right of the target mean value computed from theoretical block mean $m = 0$ and block variance σ^2_{BB} according to Section 4.1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

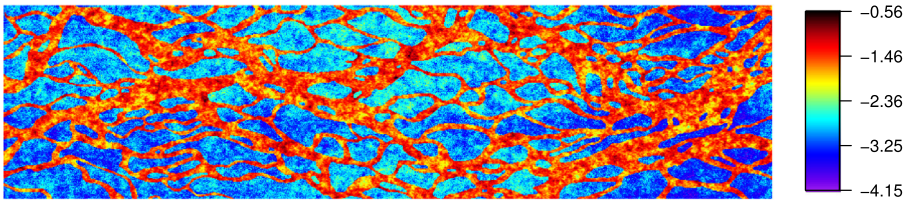


Fig. 6. Training image, 764 × 239 pixels of units 1.95312 × 1.5625 m, representing a log-permeability field of a braided system.

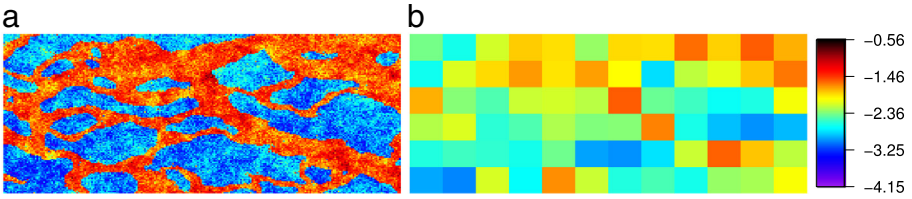


Fig. 7. Log-permeability reference field: (a) unconditional simulation (using the TI of Fig. 6), 240 × 120 pixels of units 1.95312 × 1.5625 m (same as for the TI); (b) mean values on blocks of 20 × 20 pixels computed from field (a). The color bar is identical to that of Fig. 6. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.1. Simulating log-permeability fields conditioned to block data

Consider the TI displayed in Fig. 6 and representing a log-permeability field in a braided river environment. The variable values follow a bimodal distribution and present sharp spatial transitions typical of a channel structure.

First, an unconditional simulation is generated using the distance (4) (for continuous variable) with the main simulation parameters $n = 24$, $t = 0.05$ and $f = 0.25$ (see Section 2). On this field, considered as a reference, we compute the average values on blocks of 20 × 20 pixels covering the entire simulation domain (Fig. 7). As a first application, we propose to use our method to downscale the field of Fig. 7(b). More precisely, we consider in input the TI of Fig. 6 which depicts the fine scale structures supposed to be known, and the 72 block data given in Fig. 7(b) (blocks and target mean values).

Let us illustrate on this example how the local standard deviation $\sigma_{TI,B}$ (18) involved in the block data error (20) is computed. Scanning the TI with a block of 20 × 20 pixels, the local mean and standard deviation (μ_i, σ_i) for every block is retrieved. Then, the value of $\sigma_{TI,B}$ for a given target local mean μ_B is computed with Eqs. (18) and (19). The joint density distribution (M, S) provided by the sample (μ_i, σ_i), and the curve of the approximations $\sigma_{TI,B}$ of $\mathbb{E}(S|M = \mu_b)$ as a function of μ_B are displayed in Fig. 8. In this example, the value of the bandwidth given by Eq. (19) is about $8.9 \cdot 10^{-3}$.

One hundred realizations are generated using a tolerance $t_B = 0.5$ for each block data, whereas the simulation parameters are set to $n = 24$, $t = 0.05$ and $f = 0.25$ as before. For each realization, one computes the *a posteriori* average value on each block given in input. Results are displayed in Fig. 9. Two realizations are shown in Fig. 9(a) and (c), and the mean values on the blocks in Fig. 9(b) and (d) respectively, which can be compared to the input block data (Fig. 7(b)). The pixel-wise mean (Fig. 9(e)) and standard deviation (Fig. 9(f)) over 100 realizations show the central tendency and the variability of the simulations. The differences between the *a posteriori* average values on the blocks and the corresponding target values are displayed for the set of realizations in Fig. 9(g). The tolerance at left $t_{B,left}$ and the tolerance at right $t_{B,right}$ is computed as a function of the target mean value μ_B as explained in Section 4.1 and the curves $-t_{B,left}$ and $t_{B,right}$ are shown in Fig. 9(g). The plot of this figure shows that the block data constraints are honored, since the differences are within the interval $[-t_{B,left}, t_{B,right}]$ of length $2 \cdot t_B = 1.0$. One can observe that for "extreme" target values μ_B, the differences approximately remain centered on 0 which results from the asymmetry of the target interval $[\mu_B - t_{B,left}, \mu_B + t_{B,right}]$: small $t_{B,right}$ (resp. $t_{B,left}$) for small (resp. large) μ_B.

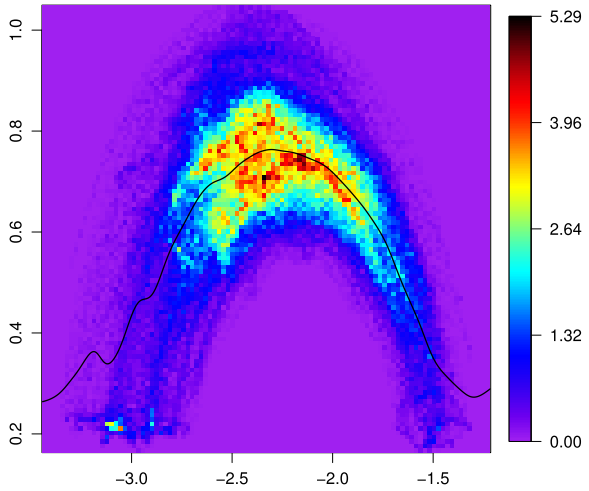


Fig. 8. Joint distribution density of mean (μ_i along abscissa) and standard deviation (σ_i, along ordinate) on every 20×20 pixels block included in the TI of Fig. 6. The black curve shows the values of local standard deviation $\sigma_{TI,B}$ as a function of target mean μ_B resulting from Eqs. (18) and (19).

Tests with the same parameters as above are carried out, with a block data tolerance set to $t_B = 0.1$ and $t_B = 1.0$. Results are shown in Fig. 10. Compared to the results obtained with a tolerance of $t_B = 0.5$ (Fig. 9), a smaller value ($t_B = 0.1$, left column of Fig. 10) results in realizations with means on blocks closer to the target values and smaller pixel-wise variability. The pixel-wise standard deviation map shows smaller values and the pixel-wise mean map is more contrasted with the blocks given in input clearly visible. Note that using overlapping blocks, mimicking "mobile averages", could attenuate the sharp limits around the blocks. On the contrary, one observes very few changes if the tolerance value is relaxed ($t_B = 1.0$, right column of Fig. 10). As discussed in Section 5, changing the tolerance value has a small impact for block data with target mean value far from the block mean m (computed from blocks in the TI) which corresponds to the abscissa value on the plots of Fig. 9(g) or 10(g)–(h) where the interval formed by the dashed lines is symmetric around 0 (*i.e.* $t_{B,left} = t_{B,right}$).

As the method has to deal with multiple constraints, *i.e.* pattern reproduction and honoring block data, a trade-off has to be found between the acceptance threshold value applied to the pattern comparison and the block data tolerance. To illustrate the potential and the flexibility of the method, we propose a second example using the same TI (Fig. 6) and overlapping input block data of various sizes. A simulation grid of the same size and same pixel units as in the previous example is considered, and six disk-shape block data are given in input (Fig. 11(i)), at three different scales: block A containing 5′067 pixels with target mean value (μ_B) of -2.0, block B with 993 pixels included in block A and a target mean of -1.7, and blocks C to F constituted of 559 pixels each with target mean values set to -2.8, -1.5, -1.6 and -2.9 respectively. The aim of this synthetic case is to simulate (log-) hydraulic conductivity fields accounting for data collected at three different scales provided by *e.g.* pumping and slug tests. The same parameters as for the previous example are chosen ($n = 24$, $t = 0.05$ and $f = 0.25$). Results with a tolerance on each block data of $t_B = 0.02$ and $t_B = 0.5$ respectively are displayed in Fig. 11. The pixel-wise mean maps are very similar, which shows that the method is stable. However, the standard deviation map presents smaller value (*i.e.* less variability at pixel scale) for the smaller tolerance.

6.2. Application accounting for geophysical data

In this section, we propose to use our method to face a more challenging (synthetic) case. The aim is to reconstruct bidimensional rock facies maps based on geophysical data. A similar context as in Lochbühler et al. (2014) is set up. Ground Penetrating Radar (GPR) measurements consisting in travel

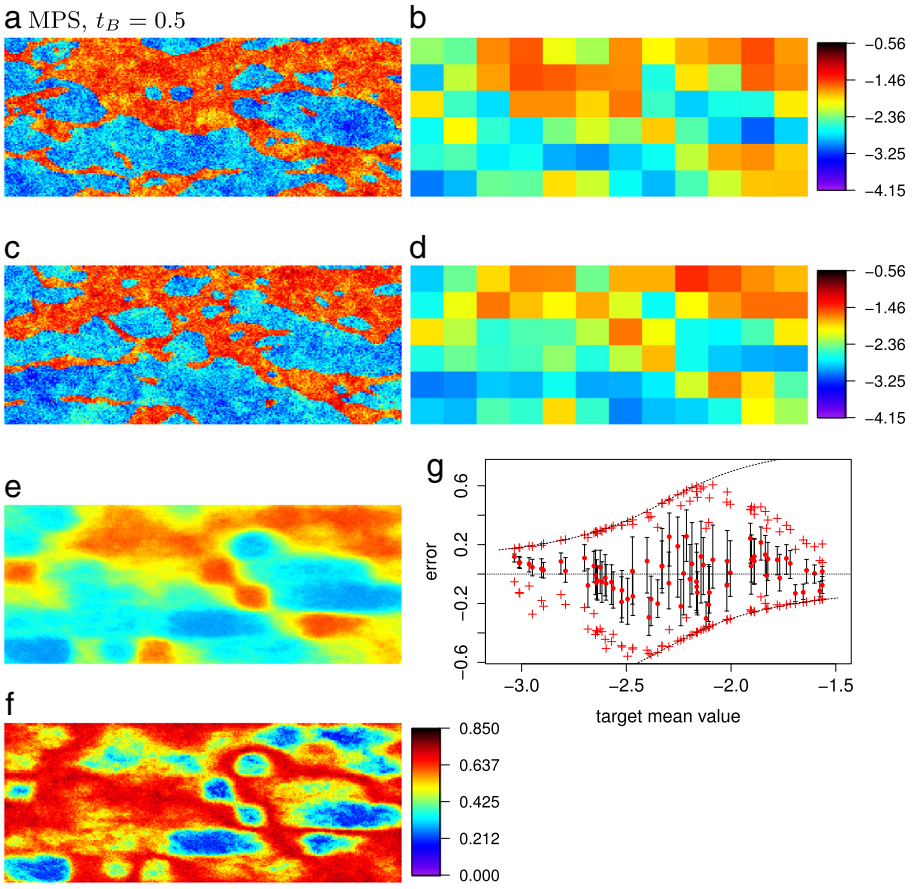


Fig. 9. Downscaling log-permeability data: (a) and (c) one realization conditioned to the block data of Fig. 7(b); (b) and (d) corresponding mean values on 20 × 20 pixels blocks; (e–f) pixel-wise mean (e) and standard deviation (f) over 100 realizations; (g) differences between observed and target values for mean on each block over 100 realizations: median (red point), interquartile range (black line), and minimum and maximum (red crosses); the two dashed curves correspond to $-t_{B,left}$ and $t_{B,right}$, the tolerances at left and at right of the target mean value computed according to Section 4.1. For plots (a)–(e) the same color scale as in Figs. 6 and 7 is used. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

times between transmitter and receiver antennas are considered. The TI with 5 facies in Fig. 12(a) is used to describe the spatial structures. This image, used by Comunian et al. (2011) for 3D geostatistical modeling, is derived from one digitized section of the fluvio-glacial aquifer environment at the Herten site (Bayer et al., 2015). In the original digitized section, 10 facies were defined. As proposed by Lochbühler et al. (2014), the number of facies is reduced to 5 by regrouping rock types having similar hydraulic conductivity and porosity, and the facies codes are converted into radar wavespeed. More precisely, the facies values used are the inverse of the radar wavespeed as explained below.

The TI used in this application is composed of 6 2D layers derived from 6 digitized sections at the Herten site as explained above. One of these layer is displayed in Fig. 12(a), and each of them is of dimension 320 × 140 pixels and represents an area of 16 × 7 square meters (1 pixel = 5 cm × 5 cm). An unconditional MPS simulation of size 80 × 140 pixels (with same units) is considered as reference (Fig. 12(b)), and 28 transmitter–receiver antennas are placed with regular spacing on both sides. Travel times between each pair of antennas from one side of the domain to the other and describing an angle of less than 50° compared to the horizontal constitutes the input data (694 pairs, white lines

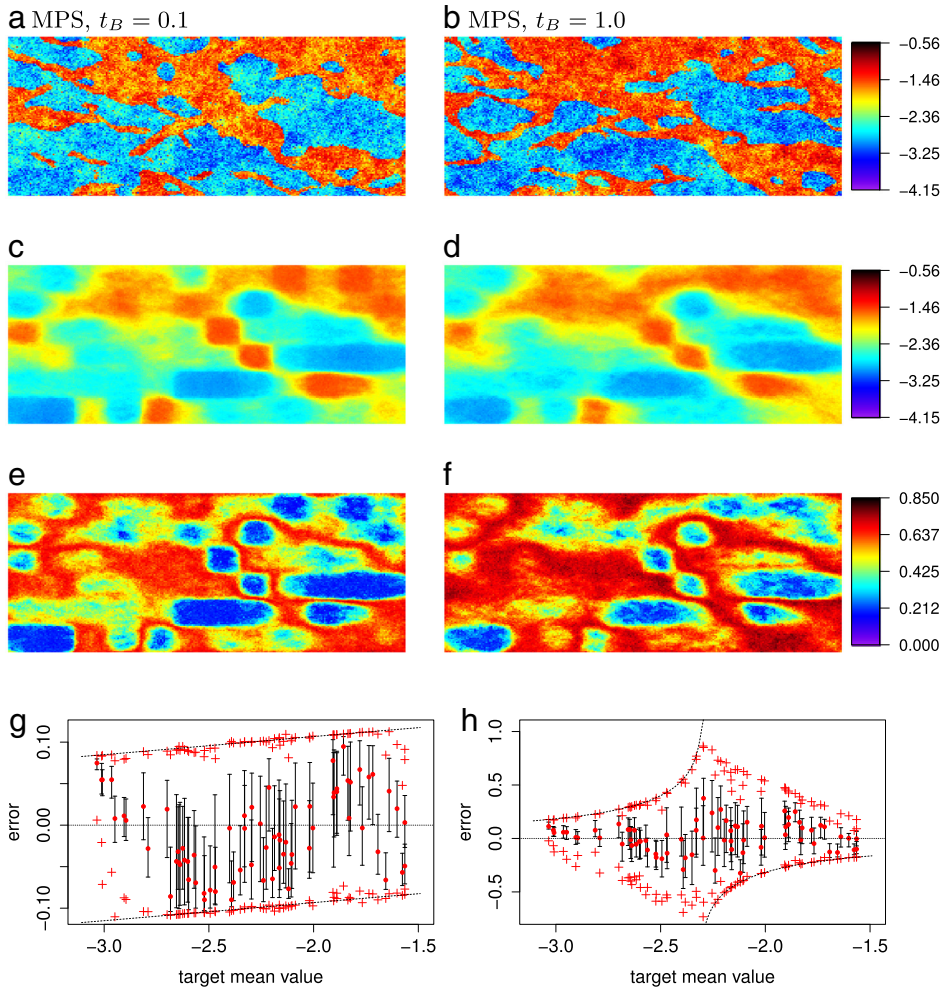


Fig. 10. Results obtained by varying the block data tolerance: (left column) results with tolerance $t_B = 0.1$; (right column) results with tolerance $t_B = 1.0$; (a–b) one realization; (c–f) pixel-wise mean (c–d) and standard deviation (e–f) over 100 realizations; (g–h) differences between observed and target values for mean on each block over 100 realizations: median (red point), interquartile range (black line), and minimum and maximum (red crosses); the two dashed curves correspond to $-t_{B,left}$ and $t_{B,right}$, the tolerances at left and at right of the target mean value computed according to Section 4.1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in Fig. 12(b)). In this synthetic case, we assume straight paths between two antennas, and compute a mean radar wavespeed along each path. As the average velocity along a path is computed as a (weighted) harmonic mean of the radar wavespeed in each rock type crossed by the path, inverse radar wavespeed values are used so that harmonic means are replaced by arithmetic means. Thus, the facies codes correspond to inverse radar wavespeed values and the input data are arithmetic means between each considered pair of antennas. In our method, the support of a block data is a subset of pixels corresponding to one path. Then, for each considered path, the computed mean is attached to the set of pixels traversed by the corresponding straight line from one antenna to the other. Thus, we obtain 694 intertwining block data made up of 80–169 pixels depending on the slope of the path.

Ignoring the reference field, 100 realizations are generated using our method. The distance (3) designed for categorical variables is used to compare two patterns and the simulation parameters of

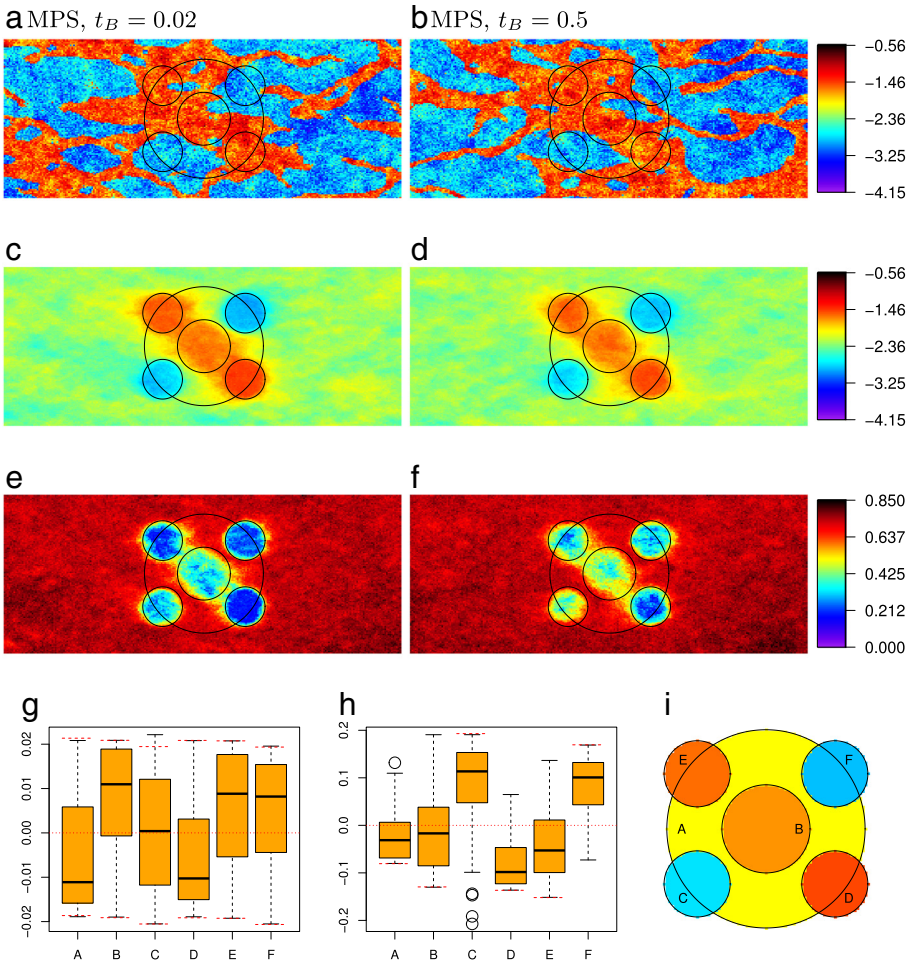


Fig. 11. Example with overlapping blocks of various sizes. Six input disk-shape block data is considered (i) in input and the TI of Fig. 6 is used. Results with tolerance $t_B = 0.02$ and $t_B = 0.5$ are shown in the first and second columns respectively: (a–b) one realization; (c–f) pixel-wise mean (c–d) and standard deviation (e–f) over 100 realizations; (g–h) differences between observed and target values for mean on each block over 100 realizations (boxplot); the red dashed lines correspond to the tolerances at left or at right of the target mean value computed according to Section 4.1. Note that same color scale is used in (a–d) and (i) and for the TI. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the direct sampling algorithm are set to $n = 36$ $t = 0.05$ and $f = 0.25$ (see Section 2). A tolerance of $2 \cdot 10^{-3}$ is used for each of the 694 block data. The results are displayed in Fig. 13. First, spatial structures of the TI (Fig. 12(a)) are well reproduced in the realizations (Fig. 13(b) and (c)), which are slightly noisy. Indeed, when a pixel is simulated, the intertwined blocks implies numerous constraints additionally to the pattern reproduction, and the method has to face several conditions, which are honored as well as possible. Moreover, the pixel-wise mean (Fig. 13(d)) and the occurrence proportion maps (Fig. 13(e)–(i)) over a set of realizations show the most likely locations for each facies.

We are aware that assuming straight paths between transmitter–receiver antennas is not realistic. Indeed, to get the propagation of the radar wave, eikonal equation should be solved as done in Lochbühler et al. (2014). However, to apply our method, we have to define the support (geometry) of the block data. Because the simulation grid is empty before starting the simulation, we cannot know

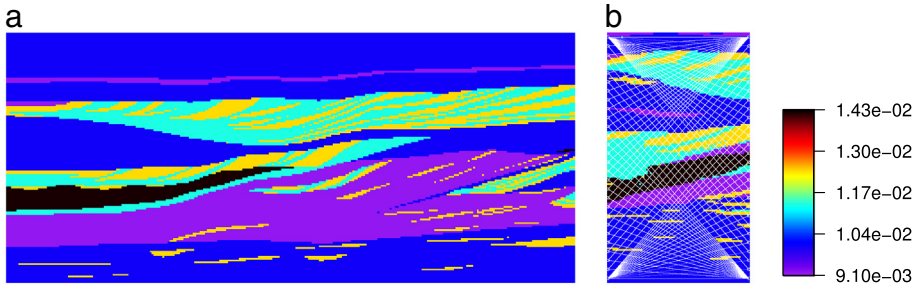


Fig. 12. TI and reference for the application accounting geophysical data: (a) one of the 6 2D layers of the TI, simplified model of a section at the Herten site; (b) one unconditional MPS simulation with transmitter–receiver antenna marked by white circles on both sides, and with straight paths superimposed in white representing the block data (among the 694 paths at total, only those with the minimal or maximal slope considering each starting and ending point are displayed). In (a) and (b) the same color scale is used, and the 5 facies correspond to inverse wavespeed values reflecting the hydraulic properties of the rock type. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

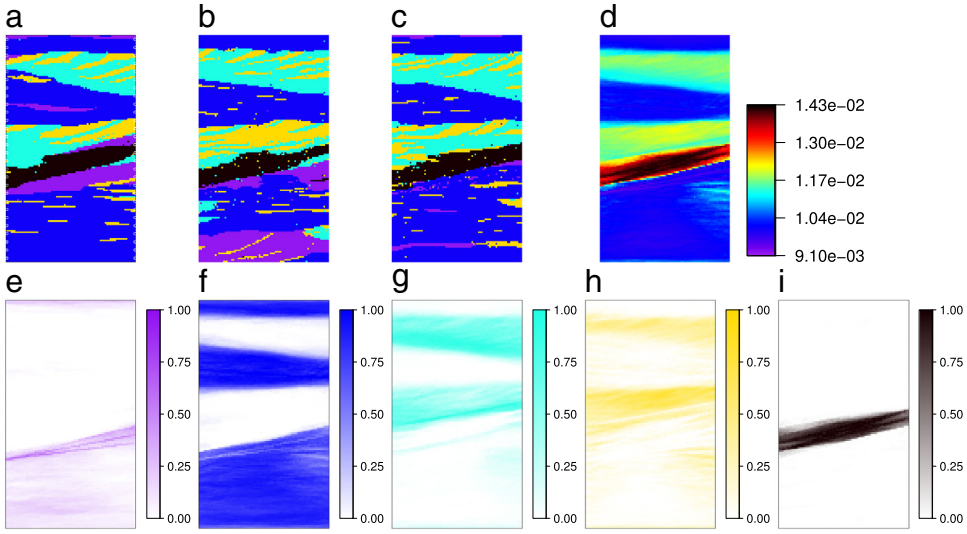


Fig. 13. Results of the application accounting for geophysical data: (a) reference field; (b–c) two realizations generated using our method ($t_B = 2 \cdot 10^{-3}$, see text for details); (d) pixel-wise mean over 100 realizations; (e–i) occurrence proportion for each facies (by color) over 100 realizations. For images (a–d) the same color scale as that of Fig. 12 is used. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the propagation paths of the radar wave. Thus, without any information, straight paths are assumed. This approach allows to guide the simulation of the rock type facies and it could be used to accelerate procedures for the inversion of geophysical data.

7. Discussion and conclusion

In this paper, a MPS simulation method allowing to account for data at different scales is proposed. The technique is based on the direct sampling algorithm (Mariethoz et al., 2010) which consists in randomly searching in the TI for a pattern compatible to the one centered at the simulated node. Several constraints in addition to the similarity of patterns can be handled. When scanning the TI, a misfit according to each constraint is computed and the scan is stopped as soon as all misfits are below some prescribed values. The proposed method, developed in the *DeeSse* software (Straubhaar, 2015),

accounts for block data, that is target mean values for any subset of nodes included in the simulation grid. The misfit related to a block data is quantified by using likelihood ratios accounting for local standard deviation of the simulated variable in blocks within the TI, conditionally to the target mean value. Moreover, the computation of this misfit also depends on the number of informed (already simulated) nodes in a block. For each block data, a tolerance is specified by the user which corresponds to the half length of a target interval containing the target value for the mean on the block. This interval (not symmetric in general) is automatically defined so that the differences between the *a posteriori* average value and the target mean value has a zero mean (unbiased).

The proposed algorithm allows to benefit from the ability of the MPS techniques to generate complex spatial features, while accounting for data collected at different scales, which is a common practical problem. Data at the scale of one node (punctual data) or data at larger scale (block data) can be handled by our method. In its current implementation, the method is designed for block data expressed as arithmetic mean of the simulated variable on known supports (blocks). The blocks are defined as subsets of pixels, which may overlap and be of any shape and size. Whereas the multiGaussian test case shows that honoring block data using the proposed method comes at the cost of a loss of variability at the pixel scale, the synthetic examples presented in this paper demonstrate the potential of the method in practical applications. Although only 2D cases were chosen for illustration, the proposed method can also be used for 3D simulations. Finally, extending the method to deal with non-additive variables is a perspective which will enlarge the range of applications.

Acknowledgments

We are grateful to Fabio Oriani for helpful discussions, and we also thank Denis Allard and an anonymous reviewer for their constructive criticism giving us the opportunity to improve the quality of this paper.

References

Allard, D., Comunian, A., Renard, P., 2012. Probability aggregation methods in geoscience. Math. Geosci. 44 (5), 545–581. http://dx.doi.org/10.1007/s11004-012-9396-3.

Allard, D., D'Or, D., Froidevaux, R., 2011. An efficient maximum entropy approach for categorical variable prediction. Eur. J. Soil Sci. 62 (3), 381–393. http://dx.doi.org/10.1111/j.1365-2389.2011.01362.x.

Arpat, G., Caers, J., 2007. Conditional simulation with patterns. Math. Geol. 39 (2), 177–203. http://dx.doi.org/10.1007/s11004-006-9075-3.

Bayer, P., Comunian, A., Hönig, D., Mariethoz, G., 2015. High resolution multi-facies realizations of sedimentary reservoir and aquifer analogs. Sci. Data 2 (150033), http://dx.doi.org/10.1038/sdata.2015.33.

Carle, S.F., 1996. A transition probability-based approach to geostatistical characterization of hydrostratigraphic architecture (Ph.D. Thesis), University of California, Davis.

Comunian, A., Renard, P., Straubhaar, J., Bayer, P., 2011. Three-dimensional high resolution fluvio-glacial aquifer analog—Part 2: Geostatistical modeling. J. Hydrol. 405 (1–2), 10–23. http://dx.doi.org/10.1016/j.jhydrol.2011.03.037.

Demir, S., Toktamis, O., 2010. On the adaptive Nadaraya–Watson kernel regression estimators. Hacet. J. Math. Stat. 39 (3), 429–437.

Deutsch, C., Journel, A., 1998. GSLIB: Geostatistical Software Library and User's Guide, second ed. Oxford University Press.

Dietrich, C.R., Newsam, G.N., 1996. A fast and exact method for multidimensional Gaussian stochastic simulations: Extension to realizations conditioned on direct and indirect measurements. Water Resour. Res. 32 (6), 1634–1652. http://dx.doi.org/10.1029/94WR02977.

Journel, A., 1999. Conditioning geostatistical operations to nonlinear volume averages. Math. Geol. 31 (8), 931–953. http://dx.doi.org/10.1023/A:1007551529317.

Krishnan, S., 2008. The Tau model for data redundancy and information combination in earth sciences: Theory and application. Math. Geosci. 40 (6), 705–727. http://dx.doi.org/10.1007/s11004-008-9165-5.

Liu, Y., 2006. Using the snesim program for multiple-point statistical simulation. Comput. Geosci. 32 (10), 1544–1563. http://dx.doi.org/10.1016/j.cageo.2006.02.008.

Liu, Y., Journel, A., 2009. A package for geostatistical integration of coarse and fine scale data. Comput. Geosci. 35 (3), 527–547. http://dx.doi.org/10.1016/j.cageo.2007.12.015.

Lochbühler, T., Pirot, G., Straubhaar, J., Linde, N., 2014. Conditioning of multiple-point statistics facies simulations to tomographic images. Math. Geosci. 46 (5), 625–645. http://dx.doi.org/10.1007/s11004-013-9484-z.

Mahmud, K., Mariethoz, G., Caers, J., Tahmasebi, P., Baker, A., 2014. Simulation of earth textures by conditional image quilting. Water Resour. Res. 50 (4), 3088–3107. http://dx.doi.org/10.1002/2013WR015069.

Mariethoz, G., McCabe, M., Renard, P., 2012. Spatiotemporal reconstruction of gaps in multivariate fields using the direct sampling approach. Water Resour. Res. 48, W10507. http://dx.doi.org/10.1029/2012WR012115.

Mariethoz, G., Renard, P., Straubhaar, J., 2010. The direct sampling method to perform multiple-point geostatistical simulations. Water Resour. Res. 46, W11536. http://dx.doi.org/10.1029/2008WR007621.

Mariethoz, G., Renard, P., Straubhaar, J., 2011. Extrapolating the fractal characteristics of an image using scale-invariant multiple-point statistics. Math. Geosci. 43 (7), 783–797. http://dx.doi.org/10.1007/s11004-011-9362-5.
Mariethoz, G., Straubhaar, J., Renard, P., Chugunova, T., Biver, P., 2015. Constraining distance-based multipoint simulations to proportions and trends. Environ. Modell. Softw. 72, 184–197. http://dx.doi.org/10.1016/j.envsoft.2015.07.007.
Meerschman, E., Pirot, G., Mariethoz, G., Straubhaar, J., Meirvenne, M.V., Renard, P., 2013. A practical guide to performing multiple-point statistical simulations with the direct sampling algorithm. Comput. Geosci. 52, 307–324. http://dx.doi.org/10.1016/j.cageo.2012.09.019.
Sheather, S., 2004. Density estimation. Statist. Sci. 19 (4), 588–597. http://dx.doi.org/10.1214/088342304000000297.
Straubhaar, J., 2015. DeeSse User's Guide. The Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel.
Straubhaar, J., Malinverni, D., 2014. Addressing conditioning data in multiple-point statistics simulation algorithms based on a multiple grid approach. Math. Geosci. 46 (2), 187–204. http://dx.doi.org/10.1007/s11004-013-9479-9.
Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., Besson, O., 2011. An improved parallel multiple-point algorithm using a list approach. Math. Geosci. 43 (3), 305–328. http://dx.doi.org/10.1007/s11004-011-9328-7.
Straubhaar, J., Walgenwitz, A., Renard, P., 2013. Parallel multiple-point statistics algorithm based on list and tree structures. Math. Geosci. 45 (2), 131–147. http://dx.doi.org/10.1007/s11004-012-9437-y.
Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Math. Geol. 34 (1), 1–21. http://dx.doi.org/10.1023/A:1014009426274.
Tahmasebi, P., Hezarkhani, A., Sahimi, M., 2012. Multiple-point geostatistical modeling based on the cross-correlation functions. Comput. Geosci. 16 (3), 779–797. http://dx.doi.org/10.1007/s10596-012-9287-1.
Tang, Y., Atkinson, P., Zhang, J., 2015. Downscaling remotely sensed imagery using area-to-point cokriging and multiple-point geostatistical simulation. ISPRS J. Photogramm. 101, 174–185. http://dx.doi.org/10.1016/j.isprsjprs.2014.12.016.
Tran, T.T., 1994. Improving variogram reproduction on dense simulation grids. Comput. Geosci. 20 (7–8), 1161–1168. http://dx.doi.org/10.1016/0098-3004(94)90069-8.
Wood, A.T.A., Chan, G., 1994. Simulation of stationary Gaussian processes in $[0, 1]^d$. J. Comput. Graph. Statist. 3 (4), 409–432. http://dx.doi.org/10.2307/1390903.
Zhang, T., Journel, P.S.A., 2006. Filter-based classification of training image patterns for spatial simulation. Math. Geol. 38 (1), 63–80. http://dx.doi.org/10.1007/s11004-005-9004-x.