

Conditioning Facies Simulations with Connectivity Data

Philippe Renard · Julien Straubhaar · Jef Caers ·
Grégoire Mariethoz

Received: 23 December 2010 / Accepted: 1 August 2011
© International Association for Mathematical Geosciences 2011

Abstract When characterizing and simulating underground reservoirs for flow simulations, one of the key characteristics that needs to be reproduced accurately is its connectivity. More precisely, field observations frequently allow the identification of specific points in space that are connected. For example, in hydrogeology, tracer tests are frequently conducted that show which springs are connected to which sink-hole. Similarly well tests often allow connectivity information in a petroleum reservoir to be provided.

To account for this type of information, we propose a new algorithm to condition stochastic simulations of lithofacies to connectivity information. The algorithm is based on the multiple-point philosophy but does not imply necessarily the use of multiple-point simulation. However, the challenge lies in generating realizations, for example of a binary medium, such that the connectivity information is honored as well as any prior structural information (e.g. as modeled through a training image). The algorithm consists of using a training image to build a set of replicates of connected paths that are consistent with the prior model. This is done by scanning the training image to find point locations that satisfy the constraints. Any path (a string of connected cells) between these points is therefore consistent with the prior model. For each simulation, one sample from this set of connected paths is sampled to generate hard conditioning data prior to running the simulation algorithm. The paper presents

P. Renard (✉) · J. Straubhaar · G. Mariethoz
Centre of Hydrogeology and Geothermics (CHYN), University of Neuchâtel, rue Emile–Argand 11,
2000 Neuchâtel, Switzerland
e-mail: philippe.renard@unine.ch

Present address:
G. Mariethoz
National Centre for Groundwater Research and Training, The University of New South Wales,
Sydney, Australia

J. Caers
Stanford University, Stanford, USA

in detail the algorithm and some examples of two-dimensional and three-dimensional applications with multiple-point simulations.

Keywords Geostatistical simulation · Multiple-point statistics · Connectivity data

1 Introduction

One of the most important features to determine when characterizing an underground reservoir with respect to its flow properties is the connectivity of the high permeable or low permeable structures. It has long been recognized that failing to capture the connectivity may bias the results of any underground fluid flow and transport modeling project (Journel and Alabert 1990; Gómez-Hernández and Wen 1998). Missing the fact that two locations are connected (a sink-hole and a spring, or an injection well and a producer) will have a large impact when modeling the response of the system around those locations. Also, an inadequate characterization of the overall connectivity structure of a subsurface domain will drastically change its large scale equivalent properties (Gómez-Hernández and Wen 1998; Zinn and Harvey 2003; Neuweiler and Cirpka 2005; Nurafza et al. 2006; Kerrou et al. 2008) and have an important impact on the global behavior of the system. These ideas have been the base for a continuous struggle to improve the characterization of heterogeneous reservoirs. They were the main driving forces in the 1980s to start developing indicator or boolean stochastic simulation techniques (see de Marsily et al. 2005 for a review). Consequently, many practical applications of stochastic techniques today are based on a hierarchical procedure in which the lithofacies are simulated first and the continuous properties, such as the porosity or the permeability, are modeled within the facies in a second step, allowing large contrast and discontinuities to be accounted for. This allows one to avoid systematically connecting the intermediate values as with the standard multi-Gaussian techniques for continuous variables (Journel and Alabert 1990; Journel and Deutsch 1993).

While the intuitive concept of connectivity is straightforward, its numerical definition is not unique. Different measures are of different relevance depending on the physical processes (flow, multiphase flow, transport) that occur in the medium (Knudby and Carrera 2005). If we focus only on geometrical measures, it is known that the variogram and the covariance functions are not sufficient to distinguish patterns having different connectivity structures (Western et al. 1998; Krishnan and Journel 2003). Two alternatives have been proposed in the literature. One considers connectivity functions as defined in percolation theory (Allard and Heresim Group 1994; Stauffer and Aharony 1994; Western et al. 2001; Knudby and Carrera 2005; Neuweiler and Cirpka 2005), while the other defines a directional multiple-point connectivity function (Krishnan and Journel 2003). To summarize the information contained in the connectivity function (the one defined in percolation theory), several authors use its integral scale (Western et al. 2001; Knudby and Carrera 2005).

Since the physics of flow and transport is significantly influenced by connectivity, it is important to infer a connectivity measure from field data and then to

simulate fields that honor it. Connectivity can be inferred from tracer tests. The first item of information extracted from tracer breakthrough is of a binary type: either some tracer has been recovered and there is a connection between the injection point and the observation point, or no tracer has been recovered, and it indicates that under the present flow conditions there was no set of streamlines connecting the two points at the time of the experiment or that the travel time between the two points was longer than the duration of the experiment. Therefore, the absence of recovery does not necessarily imply the absence of a connected path of permeable medium between the two points. In addition to such binary connectivity information, the tracer test provides the rate of recovery, the distribution of travel times, and the peak concentration; this information is usually processed as data for an inverse problem that allows the numerical model to be further constrained.

To our knowledge, only one technique has been proposed to simulate parameter fields that honor a connectivity constraint such as the one imposed by a tracer test. The method, which was proposed by Allard (1994), is general but has been applied only in the context of a truncated Gaussian model. The principle is to run an iterative Gibbs sampler. The initial field is generated without accounting for the structural information, but it ensures that the connectivity constraints are respected. The Gibbs sampler is then used to obtain, after a certain number of iterations, a field that respects both the structure and the connectivity (or disconnectivity) information. One difficulty is that the iterative procedure may be slow and that it may not converge if the stochastic model is incompatible with the connectivity constraint. Note that this lack of convergence is not a limitation of the method but a problem related to selection of the model.

In this paper, we restrict ourselves to the simplest type of model: a binary medium under stationarity assumption. One phase is supposed to be highly permeable, while the other phase has low permeability (a typical sand–shale deposit). We then present and discuss several tools to measure and define the geometrical connectivity of these media. In a second step, we present a fast technique that allows the imposition of connectivity data when generating stochastic realizations of a binary medium. Even though the method is presented in the framework of multiple-point statistics, it is general and can be used with a wide range of simulation techniques, such as sequential indicator simulations, truncated plurigaussian simulations, or any other method that allows conditioning with hard data.

The overall motivation is to improve the characterization techniques to account for connectivity constraints. The proposed tool should facilitate and accelerate inverse problem solving (such as history matching) by providing a reduced space in which the inverse procedure will search for solutions reproducing observed physical quantities (pressures, breakthrough curves, etc.). By providing an initial field reproducing connectivity, and by providing tools to perturb these fields while preserving the connectivity, the inversion can be made easier. This has been shown in the work of Alcolea and Renard (2010), who compared the results of an inversion method using a prior distribution that either accounted or did not account for the connectivity information.

2 Definitions

Assume for simplicity that the domain Ω of interest is partitioned by a regular Cartesian grid. Each cell represents an elementary volume ($\Delta x \Delta y \Delta z$). In each cell, there are only two types of porous medium, represented by an indicator variable I that can either take the value 1 (the cell is highly permeable) or 0 (the cell is almost impermeable). We define a point or a cell in the domain by its location x which is a vector of three spatial coordinates.

2.1 Two-Point Connectivity

Two cells x and y in Ω are said to be connected (notation: $x \leftrightarrow y$) if there exists at least one path of adjacent grid cells that allows the path to go from x to y always remaining in the permeable phase. Two cells are defined as adjacent if they are in contact through one of their faces. We do not consider in this work cells adjacent by their edges (in three-dimensional) or corners (two-dimensional and three-dimensional).

When the function $I(x)$ is known, a computationally efficient way of checking that any two cells are connected is to construct a cluster identification function $C(x)$ that identifies with a unique value every group of adjacent cells. The concept of a group of adjacent cells is well known in percolation theory under the terminology of a cluster (Stauffer and Aharony 1994). The same concept is known in petroleum engineering literature as a geobody (Deutsch 1998). In both cases, it is a set of cells connected together by their faces. The function $C(x)$ is equal to zero for all the non permeable cells and is equal to a constant integer value for each cluster. The value $C(x)$ is different for each cluster; it is the identifier of this cluster. The function $C(x)$ is chosen such that it increases regularly from 1 to the total number of clusters within Ω (Fig. 1). The details of an efficient computer implementation of the calculation of $C(x)$ from $I(x)$ are given in Hoshen and Kopelman (1976). Once the function $C(x)$ is defined, testing the connectivity between two cells x and y is equivalent to testing if $C(x)$ is equal to $C(y)$ (and positive). This approach is extremely efficient in testing the connectivity between many cells.

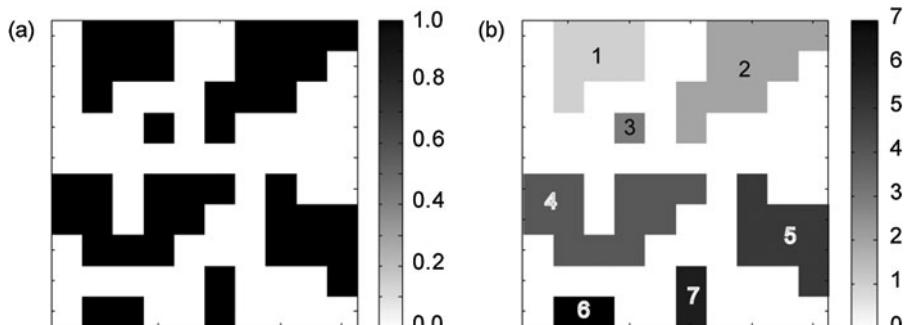


Fig. 1 Example of a binary medium. (a) The indicator function $I(x)$ representing the two phases. (b) The cluster identification function $C(x)$ that allows to identify the different connected parts

2.2 The Connectivity Function

To measure how the connectivity evolves as a function of a lag vector h between two points ($h = y - x$), we can use the connectivity function or pair connectivity $g(h)$. The connectivity function is defined, in the framework of percolation theory, as the probability that any permeable cell x taken uniformly in Ω is connected with (i.e. belongs to the same cluster as) another cell located at $x + h$ (Stauffer and Aharony 1994).

$$g(h) = P(x \leftrightarrow x + h \mid I(x) = 1) = P(C(x) = C(x + h) \mid I(x) = 1). \quad (1)$$

By applying the conditional probability rule, $g(h)$ can be written as the ratio

$$g(h) = \frac{P(x \leftrightarrow x + h, I(x) = 1)}{P(I(x) = 1)} = \frac{P(x \leftrightarrow x + h)}{P(I(x) = 1)} = \frac{1}{p} P(x \leftrightarrow x + h), \quad (2)$$

where $p = P(I(x) = 1)$ is the fraction of permeable cells. To write the second equality we use the fact that the event $x \leftrightarrow y$ is a subset of the event $I(x) = 1$, and therefore the probability of the intersection of the two events is equal to the probability of the first.

3 Connectivity Conditioning Algorithm

We consider now the problem of imposing connectivity when simulating a binary medium. For that purpose, we propose an algorithm that is based on the idea of borrowing connected paths from a training image instead of iteratively building a simulation that satisfies both the connectivity and the structural constraints. The advantage of the approach is that it is numerically efficient and that it ensures consistency between the geometry of the paths and the random function model. The consistency is imposed by using as the training image (to search for connected paths) either the one used as input in a multiple-point algorithm such as impala (Straubhaar et al. 2011), snesim (Strebelle 2002), filtersim (Zhang et al. 2006), the direct sampling method (Mariethoz et al. 2010), or an unconditional simulation constructed with the method that will subsequently be used to generate the realizations.

The connectivity function (1) given in the previous section allows us to check the compatibility between the connectivity data and the training image (from which a path is borrowed). This verification can be included in the initialization phase of the algorithm (see Sect. 3.1 below). Hence, the method will be applied only if the connectivity constraints are sufficiently coherent with the conceptual model. First, we propose a base algorithm for the situation where connectivity data involve only two points. Next, we extend this algorithm to account for multiple-point connectivity data.

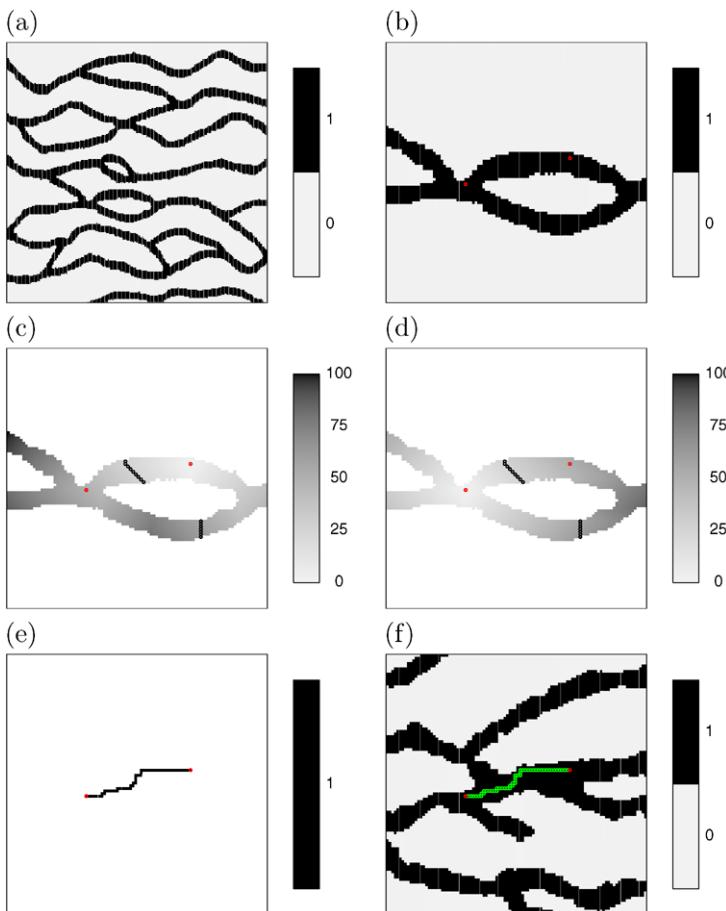


Fig. 2 Step by step description of the base algorithm. (a) Training image (250×250 cells) (Strebelle 2002). (b) Image of the simulation grid (100×100 cells) after one replicate of a connected geobody has been translated and pasted in the simulation. The two *red circles* mark the points that must be connected. (c, d) Distance function from each conditioning point, which is calculated inside the geobody to draw the random paths. The two *red circles* mark the points that must be connected. The *black circles* show the location of the points in the geobody for which the distances from each of the conditioning point are the same (or differ of 1 pixel). (e) One of the path binding the two conditioning points. (f) Final result of the simulation. The *green circles* highlight the connecting path

3.1 Base Algorithm for 2-Point Connectivity

The base algorithm can be described as follows (Fig. 2).

- The input data are
 - A pair of connected points (x, x') located on the simulation grid G_s .
 - A binary training image (Fig. 2a), i.e. an indicator function $I(x)$ defined over a grid G_t (typically larger than G_s).

The aim is to simulate an indicator variable $I(x)$ over G_s such that x and x' are connected ($x \leftrightarrow x'$). We start by initializing the algorithm.

1. We define the lag vector $d = x' - x$.
2. We check the compatibility of the training image and the constraint of connectivity between the two points x and x' . For this, we use the connectivity function (1) to compute the probability of connection in the training image for a pair of points distant from a lag d : if $g(d)$ is below a given threshold, we consider that the connectivity data are inconsistent with the training image and then the algorithm stops; otherwise, the algorithm continues to the next step.
3. We calculate the cluster function $C(z)$ on G_t . It allows all the connected (geo)bodies existing in the training image to be defined.
4. We then scan the training image G_t to find all the grid cells z_i such that $C(z_i) = C(z_i + d) > 0$. In other words, we look for all the locations in the training image which feature $x \leftrightarrow x'$. The number n of such replicates is stored, as well as the location of all the z_i values ($i = 1, \dots, n$).
5. If $n = 0$, then the algorithm stops because this means that the training image (stochastic model) is not compatible with the constraint. The problem can be solved by considering a larger training image or changing the stochastic model (providing a different training image, or changing the ranges of the model).

At this point, all the preprocessing is done and we enter a loop that is applied for each simulation.

- (i) One index i is chosen randomly between 1 and n . It corresponds to the random selection of one replicate of a connected pattern (all the replicates have the same probability of being chosen).
- (ii) The whole cluster is identified by $C(z) = C(z_i)$ and copied from G_t to G_s as long as its location is such that z_i in G_t corresponds to x in G_s (Fig. 2b).
- (iii) Because G_t is generally larger than G_s , it is possible that a large cluster connected in G_t is no longer connected in G_s . Therefore, we must check if x and x' are really connected. If such is not the case, we need to start again with another replicate.
- (iv) Because the whole replicate may occupy a large portion of the simulated image after having been copied, keeping the whole replicate in the simulation would restrict very significantly the possible variability within G_s . Instead, we build a path within the replicate and set all the cells along the path to 1 (Fig. 2e). Because we consider that the connected paths go through the faces of the grid cells, there are many paths that might go from x to x' while remaining within the connected geobody. To consider a maximum number of paths while minimizing their length (to ensure the maximum of variability later on), we proceed as follows: The idea for building the path is first to define a point x_m that is within the geobody G_s and at equal distance from x and x' , and then to build the path by starting from this midpoint and creeping in both directions toward x and x' . For any pair of grid cells (u, v) in the geobody G_s , we compute a distance $d(u, v)$ defined as the minimal number of faces of grid cells to be crossed to bind u and v while remaining within the geobody. Any path connecting x and x' contains a

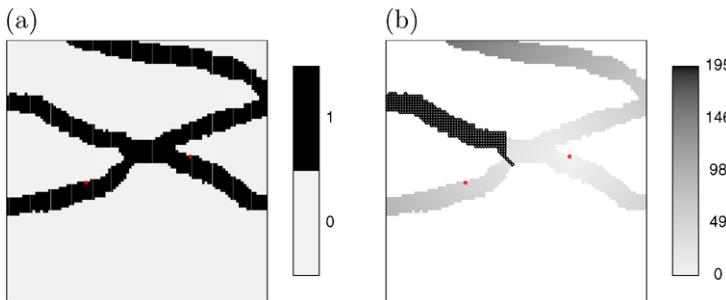


Fig. 3 Special case for building a connecting path. The same training image and same conditioning points (marked in red) that must be connected are used as in the example of Fig. 2. (a) Image of the simulation grid after one replicate of a connected geobody has been translated and pasted in the simulation. (b) Distance function from the right conditioning point. The black circles show the possible starting points for the random walkers used for building the path: many of them are in a “dead arm” of the connected geobody

grid cell x_m such that the distance from x and the distance from x' differ of at most 1, i.e.

$$|d(x_m, x) - d(x_m, x')| \leq 1. \quad (3)$$

The path binding x and x' is built by randomly selecting a pixel x_m in the connected geobody satisfying (3) (Figs. 2c and 2d). From this location, a walker goes down from x_m to x (resp. x') along a slope whose altitude is the distance to the point x (resp. x'). At each step a point having an altitude inferior to the current location is randomly selected.

- (v) All these cells (Fig. 2e) are taken as conditioning data for the simulation algorithm which is applied as usual, whatever the technique and the model (Fig. 2f).

The path binding x and x' , built as described in point (iv) above, can contain closed loops. This can occur when the random walkers from x_m to x and from x_m to x' go through the same cell. In such a situation, all the pixels upstream in the path from the cells common to the two random walks are not retained, because these pixels are superfluous to ensure the connectivity between x and x' . In other words, if x_c is the last common node in the two random walks going from x_m , then the nodes from x_m to x_c (not included) along these two paths are removed and x_c is considered as the “new middle point” of the path binding x and x' . Figure 3 shows such a case where many of the potential starting points for the random walkers are located in a “dead arm” of the connected geobody. Note that there are two levels of uncertainty for building the path: first, a connected pattern is chosen from the training image randomly (point (i) above), then the choice of a path inside the selected geobody as described in point (iv) above again ensures some variability.

3.2 Accounting for Hard Conditioning Data

The data set may contain hard data locations (without connectivity constraints) that must be taken into account to locally condition the simulations in addition to the

connectivity constraints. We must ensure that the simulation will satisfy the usual conditioning constraints as well as the connectivity constraints. Two approaches are proposed to extend the algorithm to this situation.

The first approach, which strictly ensures the consistency between the model and the simulation, is to scan the training image with the whole set of constraints and to look for replicates that satisfy the connectivity conditions and the local values. We then keep only the replicates that satisfy these criteria and proceed as described above. The problem is that the number of replicates that satisfy all conditions reduces very rapidly when the number of conditioning points increases. Very large training images would be needed, and this technique is therefore not practical.

The second approach, which consists of modifying the values of the replicate (connected geobody) locally where we have conditioning data, is done before selecting the random path. Practically, we simply put all the hard conditioning data into the simulation grid, after having pasted the replicate (geobody) that connects the two points: this constitutes an additional step, between step (ii) and step (iii) in the base algorithm described above. This ensures that the non permeable cells within the conditioning dataset remain-non permeable. The path binding the two points naturally avoids these cells and allows the creation of a simulation that satisfies both the connectivity and the local conditioning. There may be cases where the conditioning data disconnect the geobody. This needs to be tested, and the conditioning data must correspond to a situation similar to the replicates that did connect on G_t but not on G_s .

In those cases, the replicate needs to be removed from the list, and the procedure has to be started again with a new replicate. Since we put the hard conditioning data independently of the choice of the pasted geobody, this second approach does not strictly ensure the consistency between the model and the set of all conditioning data (including the additional ones given by the algorithm). However, we have observed that the method performs adequately, at least when the amount of hard conditioning data is not too large.

3.3 Accounting for Multiple-Point Connectivity

In some situations, more than two points must be connected together. As for the hard conditioning data, an approach that strictly ensures the consistency between the model and the simulation would start by pasting a geobody from G_t into G_s that contains all the points that must be connected. But again, the chance of finding such geobodies, even in a large training image, rapidly decreases when the number of connected points increases, making this technique unusable.

We propose a simple way to generalize the base algorithm for the cases where the connectivity constraints involve more than two points. First, we consider the situation where three points x, x', x'' must be connected. To honor such a constraint, we proceed as follows: we start by building a path $\{x = u_1, u_2, \dots, u_N = x'\}$ binding x and x' , following the base algorithm (steps (i–iv)), and then we build a path that connects x'' to the path $\{x = u_1, u_2, \dots, u_N = x'\}$ before running the simulation algorithm.

This involves building a path that connects a point x'' to a set of points $\{x = u_1, u_2, \dots, u_N = x'\}$. For this, the base algorithm needs to be adapted.

1. We choose a cluster in G_t that features one of the connections $x'' \leftrightarrow u_j$, $j = 1, \dots, N$, i.e a cluster that contains a point z such that $C(z) = C(z + u_j - x'')$ for at least one j . Then, we paste this geobody entirely in G_s such that the location of z in G_t corresponds to the location of x'' in G_s .
2. We put the path $\{u_1, \dots, u_N\}$ (and the eventual hard conditioning data) into G_s , and, as in the base algorithm, we check if x'' and one of the points u_j is really connected.
3. For each cell v in the connected geobody, we compute the distance $d(v, x'')$ defined as in the base algorithm and the distance $d(v, \{u_1, \dots, u_N\})$ defined as the minimal number of faces of grid cells to be crossed to bind v and one of the points in $\{u_1, \dots, u_N\}$ while remaining within the geobody. Then, we randomly choose in the geobody a point x_m that verifies

$$|d(x_m, x'') - d(x_m, \{u_1, \dots, u_N\})| \leq 1, \quad (4)$$

and then, using a random walker, we go down from x_m to x'' and from x_m to one of the points in $\{u_1, \dots, u_N\}$ along a slope whose altitude is given by the distances $d(., x'')$ and $d(., \{u_1, \dots, u_N\})$, respectively.

The points u_2, \dots, u_{N-1} are not known in advance because these locations are determined using a method having a random component (base algorithm, steps (i–iv)). Then, in order to consider all the clusters in G_t that feature one of the connections $x'' \leftrightarrow u_j$, $j = 1, \dots, N$, we have to scan the training image for each new simulation. To avoid repeating the scan of the training image, we consider only the clusters in G_t that feature at least one of the connections $x'' \leftrightarrow x$ and $x'' \leftrightarrow x'$. Then, we can include the following computation in the preprocessing treatment preceding the loop applied to all simulations:

1. We define the lag vectors $d_1 = x - x''$ and $d_2 = x' - x''$.
2. We scan the training image G_t to find all the grid cells z_i such that $C(z_i) = C(z_i + d_1) > 0$ or $C(z_i) = C(z_i + d_2) > 0$, and we store the number l of such replicates as well as the location of all the z_i values ($i = 1, \dots, l$).
3. If $l = 0$, then the algorithm stops (for the same reasons as in the base algorithm).

A sketch of the proposed algorithm with three points x, x', x'' that must be connected is presented in Fig. 4. We choose to first connect the two closest point x, x' for the L_1 distance, and then to connect the third point x'' to the path binding x and x' . An alternative could be to first select the pair of points having the highest probability of being connected according to the connectivity function.

The generalization to n -point connectivity is obvious. If x_1, \dots, x_n are n points that must be connected together, we first build a path connecting x_1 and x_2 , and then, for $i = 3, \dots, n$ successively, we build a path connecting x_i to one point of one of the previous paths. Then, the union of all these paths ensures the connectivity between the n points, and constitutes additional conditioning data for the simulation algorithm. Some computation in the preprocessing phase of the algorithm is required: to connect the point x_i , all the grid cells z in G_t such that $C(z) = C(z + x_j - x_i) > 0$ for at least one j in $\{1, \dots, i-1\}$ are stored. For the following examples, we first connect points that are closer to each other.

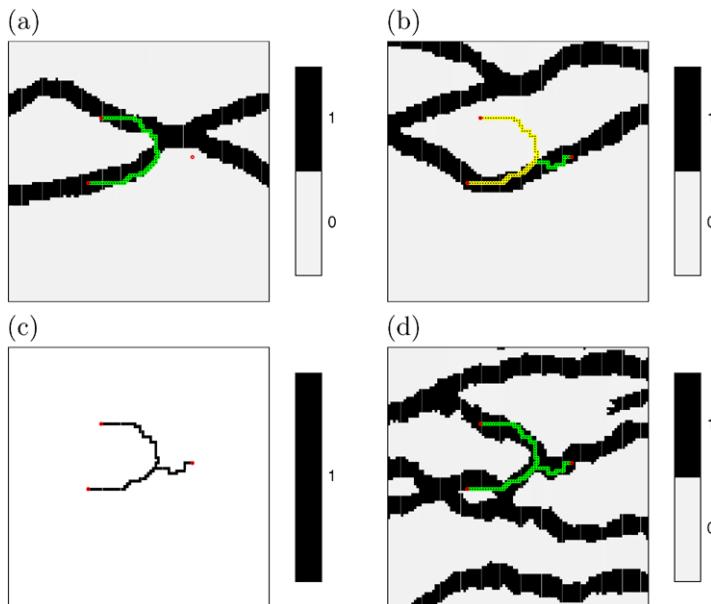


Fig. 4 Sketch of the algorithm for connecting three points. The same training image (Fig. 2a) as in previous examples is used. The three *red circles* mark the points that must be connected. (a) Image of the simulation grid (100×100 cells) after one replicate of a geobody connecting the two closest points has been pasted in the simulation, and a path in *green* binding these two points inside this geobody built using the base algorithm. (b) Image of the simulation grid after one replicate of a geobody connecting the third point and one of the two others has been pasted in the simulation, the path built in (a) in *yellow*, and a path in *green* binding the third point to the *yellow* path. (c) The additional conditioning points that have been added to ensure the connectivity of the three points. (d) Final simulation. The *green circles* highlight the points shown in (c)

4 Tests

By construction, the algorithm described in the previous sections will always ensure that the connectivity constraints are honored. The randomization of both the choice of the connected component and of the paths between the cells have been designed to minimize the risk of creating systematic bias in the statistics of the realizations. The main objectives of the following tests are therefore (1) to illustrate the behaviour of the algorithm in different situations, (2) to check that the method does not create artifacts, and (3) to evaluate the computational efficiency of the method. The algorithm proposed in this paper is implemented in the parallel multiple-point simulation code *impala* (Straubhaar et al. 2011), which is used for all cases presented.

4.1 Two-Dimensional Tests

In the first test, we use the channel image (250×250) of Fig. 2a as a training image and we consider a simulation grid of size 100×100 cells. Three levels of multigrids are used, and the search templates have a disc shape and contain 100, 60 and 20 nodes for the coarse, intermediate and fine multigrids, respectively. We consider the

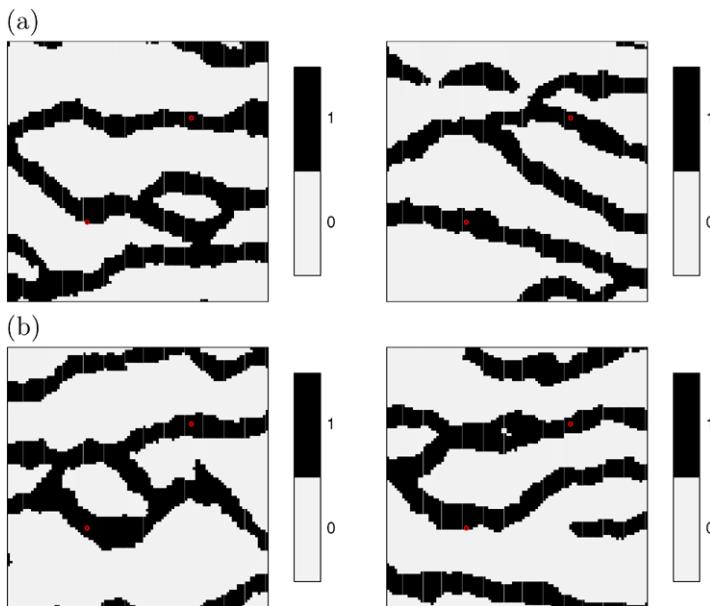


Fig. 5 Comparative example for 2-point connectivity; multiple-point simulations with *impala*: **(a)** accounting only for conditioning data, **(b)** accounting both for conditioning data and for connectivity constraints (proposed algorithm). In both cases, two realizations are shown. The red circles represent the location of the conditioning data. The grid size is 100×100 cells. The training image (250×250) shown in Fig. 2a is used

two conditioning data, located at the coordinates (30, 30) and (70, 70), with a value of 1 (permeable cell). We generate 100 realizations for the three following situations:

- (1) Conditional only to the data (without taking into account for the connectivity).
- (2) Conditional both to the data and to the connectivity constraints obtained by the rejection method: it consists simply of creating a realization conditional to the local values and keeping it if and only if it satisfies the connectivity constraints.
- (3) Conditional both to the data and to the connectivity constraints obtained by the proposed algorithm.

Two realizations in cases 1 and 3 are shown in Fig. 5. When connectivity constraints are not imposed (case 1), the two points can be connected (Fig. 5a left) or not (Fig. 5a right). For the three cases, we compute the E-type and the variance map of the 100 simulations, i.e. the mean $p(x, y)$ and the variance $v(x, y) = p(x, y)(1 - p(x, y))$ at each grid cell of the 100 simulations. These maps are shown in Fig. 6.

For a second test, we add 100 conditioning data located at the coordinates (50, .), extracted from an unconditional simulation. No connectivity constraint is considered for these data. With identical connectivity constraints as in the previous example, we generate 100 realization with each method. The results are presented in Fig. 7 (simulations) and Fig. 8 (E-type and variance maps). Finally, a third test is done with three permeable cells that must be connected, located at the coordinates (30, 30),

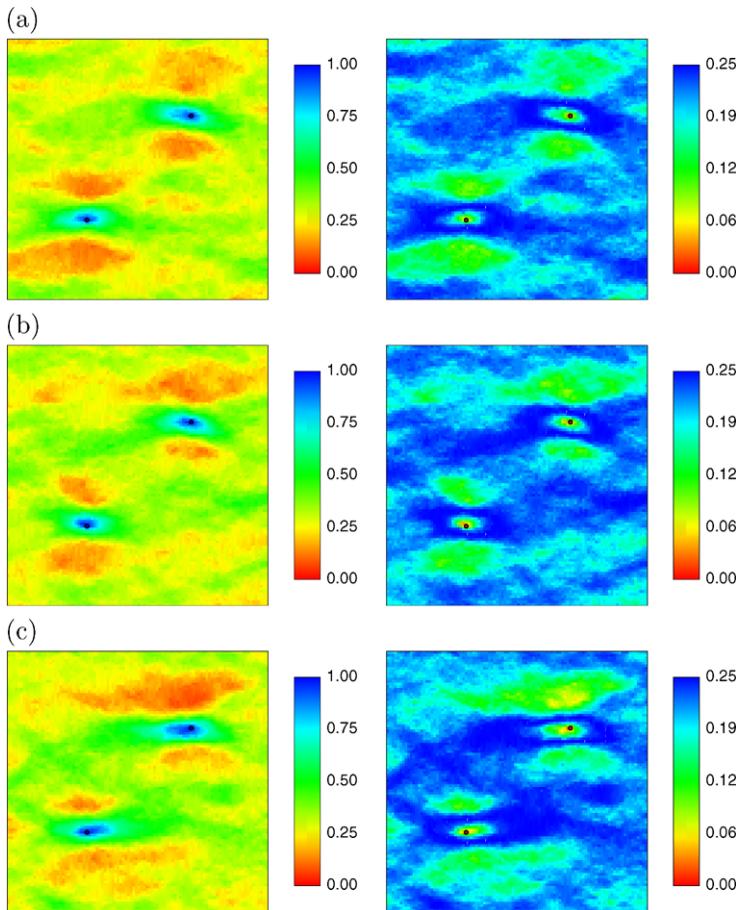


Fig. 6 Comparative example for 2-point connectivity; multiple-point simulations with *impala*; E-type (left) and variance maps (right) of 100 simulations: (a) accounting only for conditioning data, (b) obtained by the rejection method, (c) obtained by the proposed algorithm. The black circles represent the location of the conditioning data. The grid size is 100×100 cells. The training image (250×250) shown in Fig. 2a is used

(70, 70) and (20, 60) (no other conditioning data are taken into account). Then, we consider 3-point connectivity constraints. The results are presented in Figs. 9–10.

For these three tests, the means of the global proportion of permeable cells are presented in Table 1. This table shows that for the three tests, the connectivity constraints give realizations with a few more permeable cells, and this becomes more marked with the proposed algorithm. The slight difference for the presented method is due to the additional conditioning data (permeable cells) that are used to ensure the connectivity, and the fact that we do not use a servo–system to correct the proportions in order to match a user target (Strebelle 2002). By comparing the E-type in Figs. 6, 8 and 10, we observe that the proposed algorithm slightly favors certain locations for the connecting channels. The variance maps in these figures show that the variance is slightly smaller with the proposed algorithm than with the

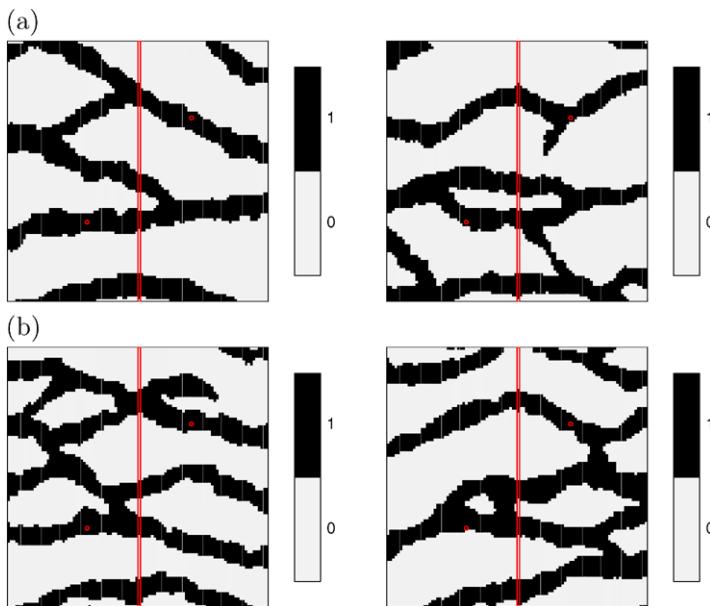


Fig. 7 Comparative example for 2-point connectivity with 100 usual hard conditioning data; multiple-point simulations with *impala*: **(a)** accounting only for conditioning data, **(b)** accounting both for conditioning data and for connectivity constraints (proposed algorithm). In both cases, two realizations are shown. The two isolated red circles represent the location of the conditioning data that must be connected and 100 usual hard conditioning data (without connectivity constraints) having same x -coordinate (middle of x -axis) are highlighted in red. The grid size is 100×100 cells. The training image (250×250) shown in Fig. 2a is used

rejection method in the neighborhood of the point that must be connected. However, we observe that the variance can locally be greater with the proposed algorithm than with the rejection method. This counterintuitive result is explained as follows: By adding the path, the probability of having a permeable cell can locally increase as compared to the rejection algorithm. If such is the case and this probability p for the proposed algorithm is lower than 0.5, the variance $p(1 - p)$ increases.

For each test, the locations of the binding paths built by the proposed algorithm are shown in Fig. 11, using the E-type of the 100 simulation grids containing all conditioning data (initial + additional) only. This shows the variability of the paths. In particular, the figure shows that the selected paths cover a wide area not restricted to the rectangle delimited by the points to be connected. We also observe in this figure that the binding path frequently ends with a horizontal line at the extremities (close to the points that must be connected). This is due to the fact that we want to add a minimum number of conditioning data points when building the path, see point (iv) of the base algorithm (Sect. 3.1). However, it is important to emphasize that these straight lines exist within the geobody picked from the training image (point (ii) of the base algorithm).

For these three tests, the rejection method required 291, 331 and 357 realizations, respectively, to obtain 100 of them that respect the connectivity constraints (2-point

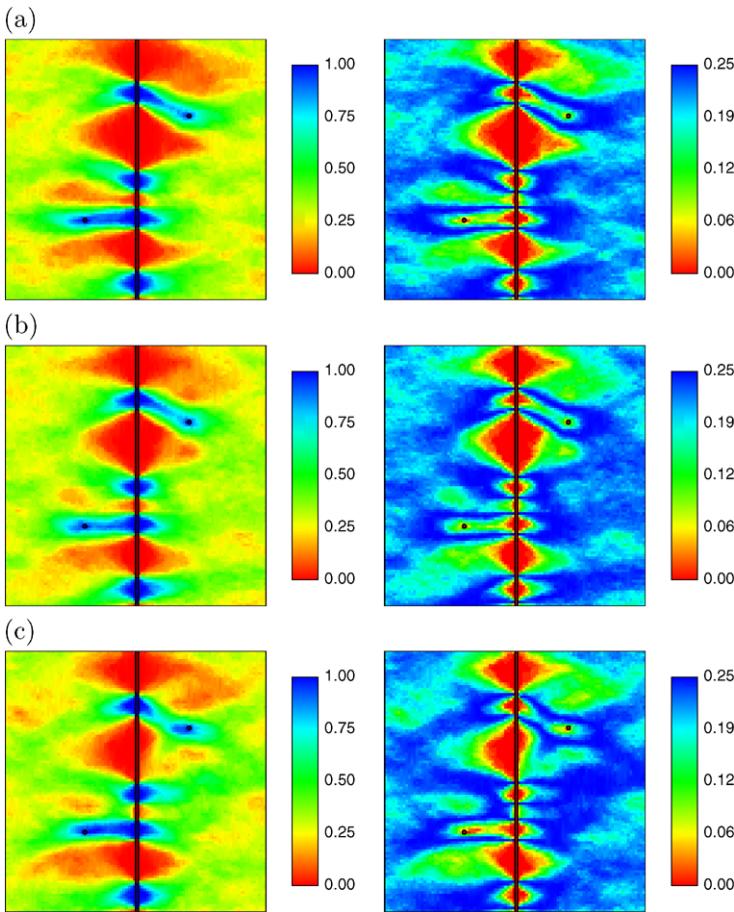


Fig. 8 Comparative example for 2-point connectivity with 100 usual hard conditioning data; multiple-point simulations with *impala*; E-type (left) and variance maps (right) of 100 simulations: (a) accounting only for conditioning data, (b) obtained by the rejection method, (c) obtained by the proposed algorithm. The two isolated black circles represent the location of the conditioning data that must be connected and 100 usual hard conditioning data (without connectivity constraints) having same x -coordinate (middle of x -axis) are highlighted in black. The grid size is 100×100 cells. The training image (250×250) shown in Fig. 2a is used

connectivity, 2-point connectivity with additional usual conditioning data and 3-point connectivity, respectively). The computation time necessary for one realization is practically the same when imposing connectivity constraints with our method, indicating that the overhead cost related to finding the path is negligible. Thus, for these tests, the rejection method is slower by a factor of about 3 compared to the proposed algorithm.

4.2 Three-Dimensional Tests

We consider the $200 \times 400 \times 140$ training image representing thin channels shown in Fig. 12 (value 1 for the permeable cells, and 0 for the non permeable cells). As for the

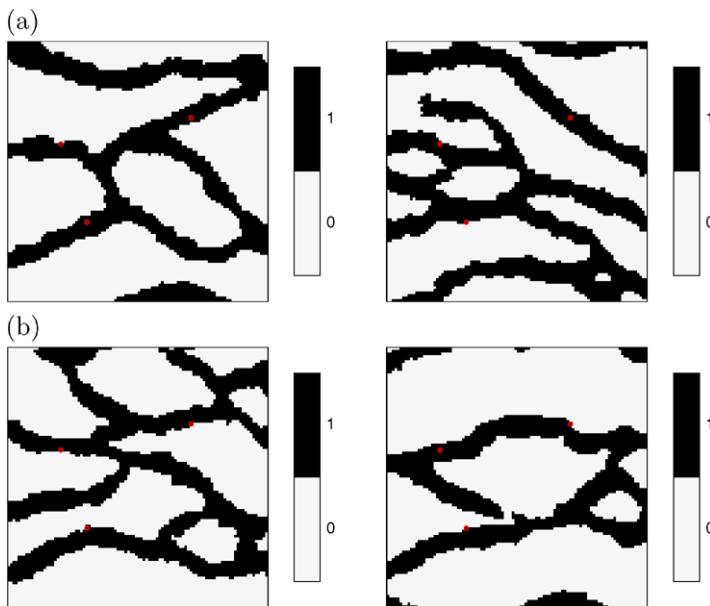


Fig. 9 Comparative example for 3-point connectivity; multiple-point simulations with *impala*: **(a)** accounting only for conditioning data, **(b)** accounting both for conditioning data and for connectivity constraints (proposed algorithm). In both cases, two realizations are shown (the red circles represent the location of the conditioning data). The grid size is 100×100 cells. The training image (250×250) shown in Fig. 2a is used

2D examples above, we use *impala* to generate 100 realizations: (1) conditional only to the data; conditional both to the data and to the connectivity constraints, (2) by rejection method, (3) with the proposed algorithm. We consider a simulation grid of size $50 \times 100 \times 50$. Three levels of multigrids are used, and spherical search templates containing 250, 56 and 26 nodes for the coarse, intermediate and fine multigrids, respectively. We test 2 and 5-point connectivity, considering the following conditioning data with a value of 1 located at the coordinates:

- (20, 10, 30), (30, 90, 20): Figs. 13–14,
- (20, 10, 30), (30, 90, 20), (15, 40, 35), (25, 25, 25), (35, 70, 15): Figs. 15–16.

For both tests, the means of the global proportion of permeable cells are presented in Table 2. In general, we can make the same observations as for the previous 2D tests. The differences in the central region of the E-type and variance maps is more marked here, probably because the points that must be connected are farther apart in these 3D tests. The number of realizations required by the rejection method to have 100 of them connected were 196 and 200 for the cases, with 2 and 5 data, respectively. Note that additional connectivity constraints do not imply that the connected realizations are more difficult to obtain. On the contrary, additional conditioning data can facilitate the connection.

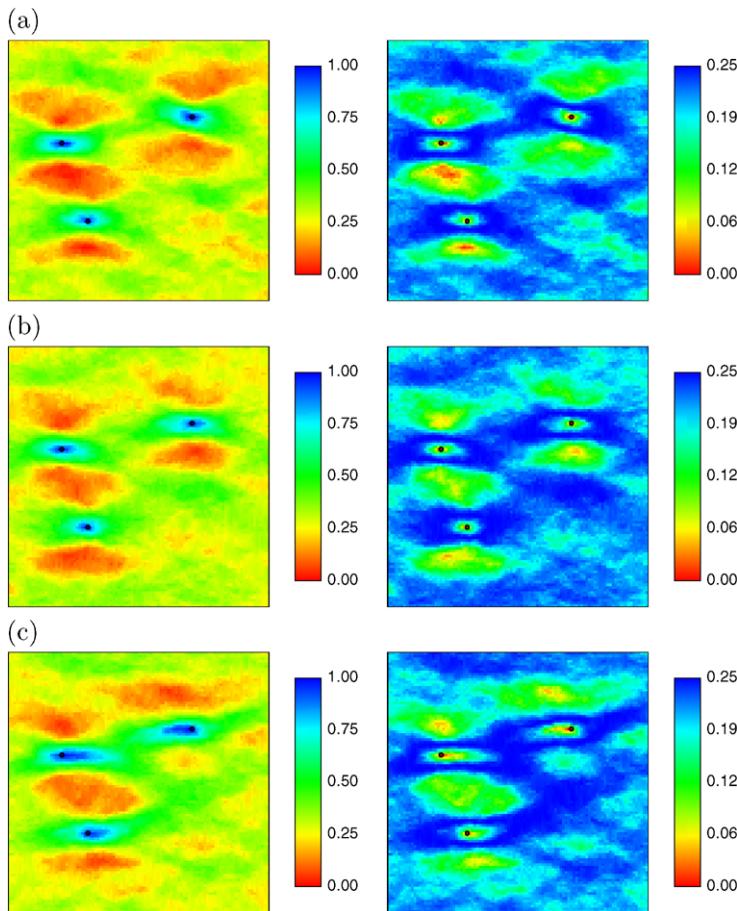


Fig. 10 Comparative example for 3-point connectivity; multiple-point simulations with *impala*; E-type (left) and variance maps (right) of 100 simulations: (a) accounting only for conditioning data, (b) obtained by the rejection method, (c) obtained by the proposed algorithm. The *black circles* represent the location of the conditioning data. The grid size is 100×100 cells. The training image (250×250) shown in Fig. 2a is used

Table 1 Mean over 100 simulations of the global proportion of permeable cells in percent for the 2D examples. Header line: number of data with connectivity constraints for rejection method and proposed algorithm, hd means hard conditioning data without connectivity constraint

	2 data	2 data + 100 hd	3 data
Standard algorithm	30.47	31.13	30.38
Rejection method	31.33	32.10	31.43
Proposed algorithm	31.95	32.96	32.57

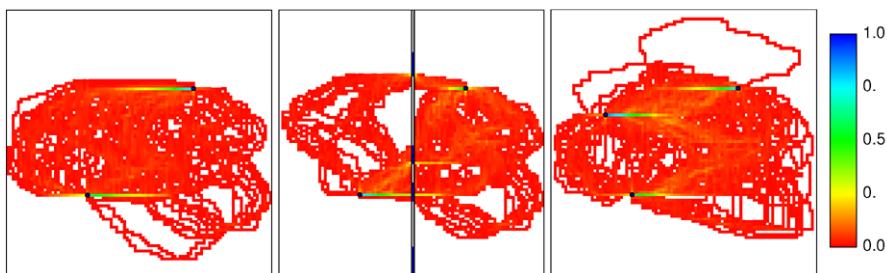


Fig. 11 Location of the binding paths (additional conditioning data) built by the proposed algorithm; E-type of 100 simulations with all conditioning data (initial + additional) only: (*left*) for 2-point connectivity example of Figs. 5–6, (*middle*) for 2-point connectivity with 100 usual hard conditioning data example of Figs. 7–8, (*right*) for 3-point connectivity example of Figs. 9–10

Fig. 12 3D binary training image of size $200 \times 400 \times 140$

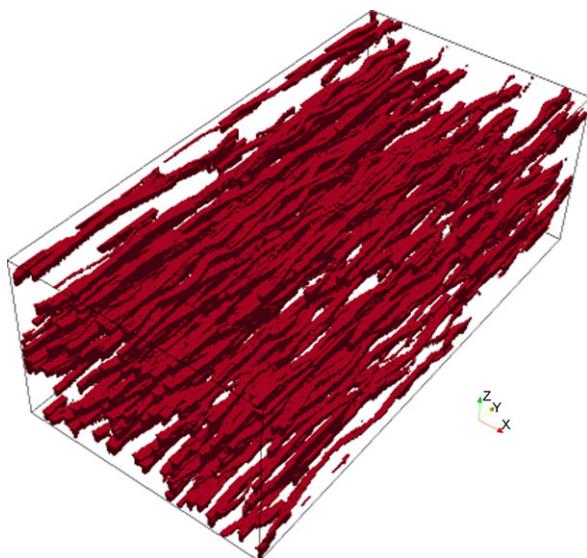


Table 2 Mean over 100 simulations of the global proportion of permeable cells in percent for the 3D examples. Header line: number of data with connectivity constraints for rejection method and proposed algorithm

	2 data	3 data	5 data
Standard algorithm	15.23	15.60	16.35
Rejection method	16.09	16.61	16.87
Proposed algorithm	17.04	17.82	20.34

5 Extensions of the Method and Outlook

Even though the method was described only for a binary variable in this paper, it can be directly applied to categorical variables. The most important limitation for the

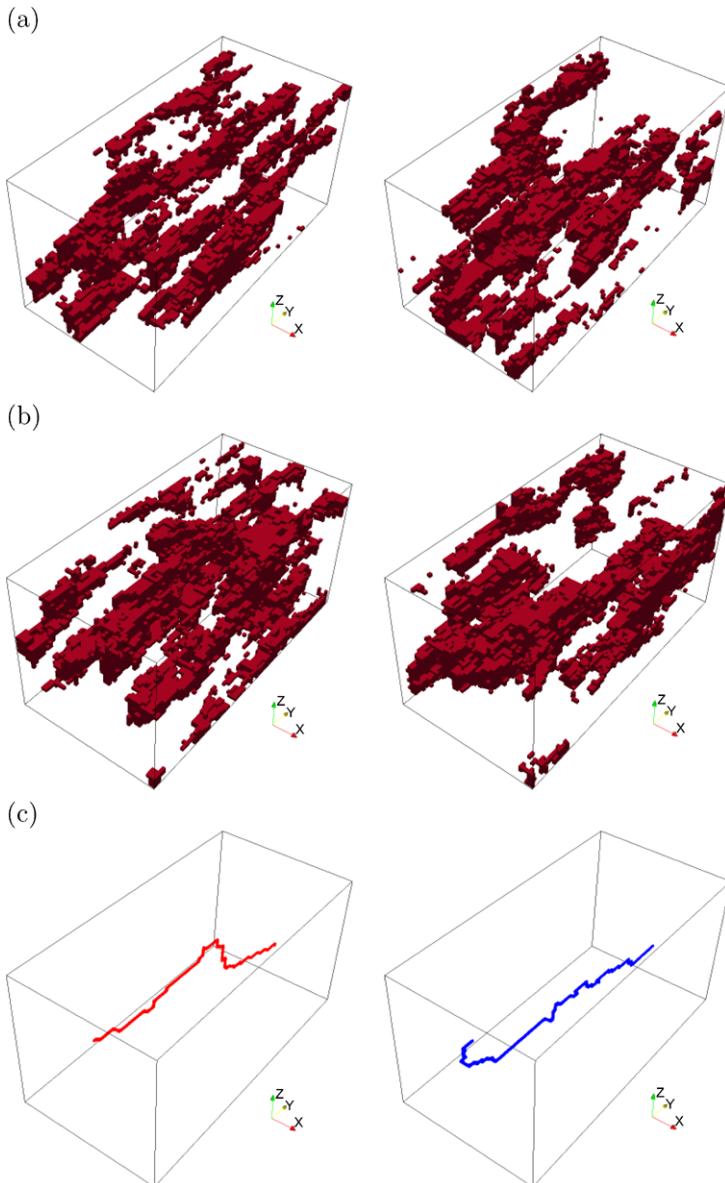


Fig. 13 Comparative 3D example for 2-point connectivity; multiple-point simulations with *impala*: (a) two realizations accounting only for conditioning data, (b) two realizations accounting both for conditioning data and for connectivity constraints (proposed algorithm). (c) Connecting paths built by the proposed algorithm (in red (left) for the simulation (b) left, in blue (right) for the simulation (b) right). Two conditioning data located at coordinates (20, 10, 30), (30, 90, 20) (permeable cells) are considered in the simulation grid of size $50 \times 100 \times 50$. The training image ($200 \times 400 \times 140$) shown in Fig. 12 is used. The connectivity between conditioning data is respected in simulation (a) left and not respected in simulation (a) right

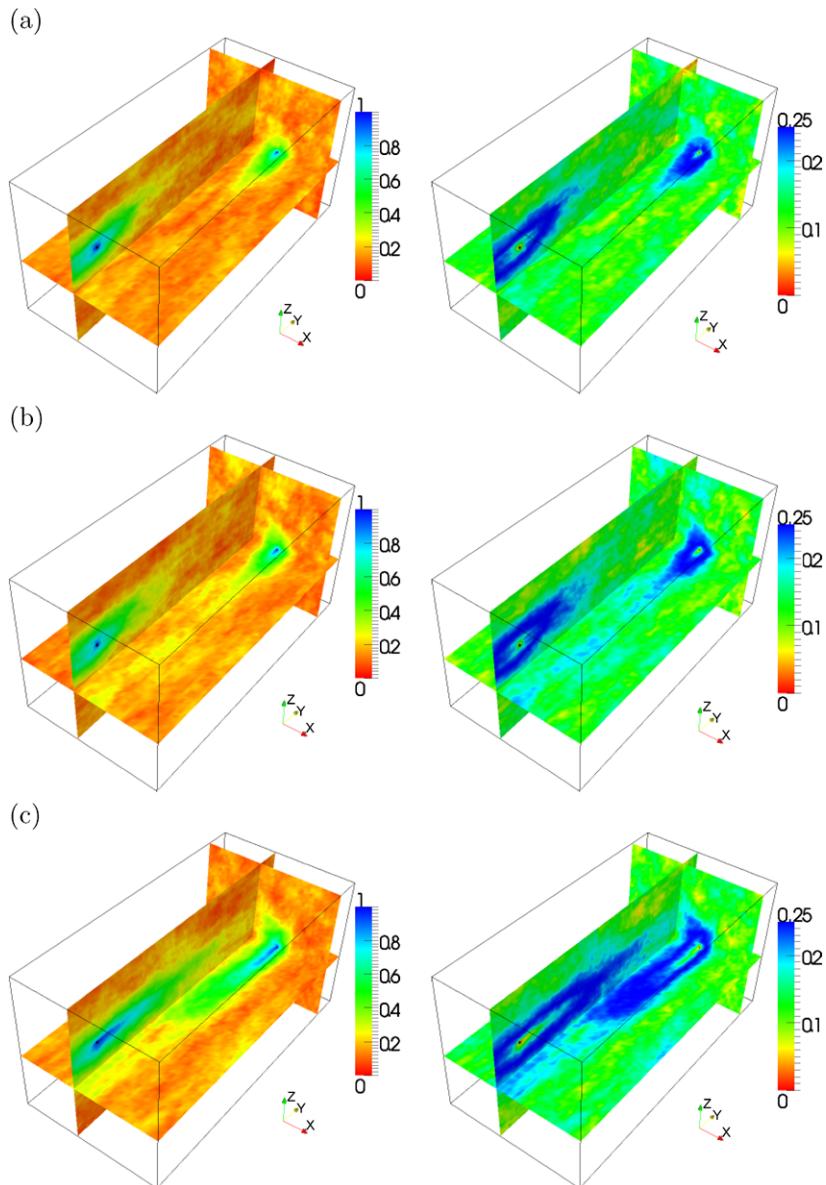


Fig. 14 Comparative 3D example for 2-point connectivity; multiple-point simulations with *impala*; E-type (left) and variance maps (right) of 100 simulations: (a) accounting only for conditioning data, (b) obtained by the rejection method, (c) obtained by the proposed algorithm. The *black points* represent the location of the conditioning data. The grid size is $50 \times 100 \times 50$ cells. The training image ($200 \times 400 \times 140$) shown in Fig. 12 is used

moment is that the method is designed for stationary fields. If the simulation must be carried out in a field with strong non stationarity (for example in orientations or proportions), it needs to be extended and adapted, either by splitting the problem into

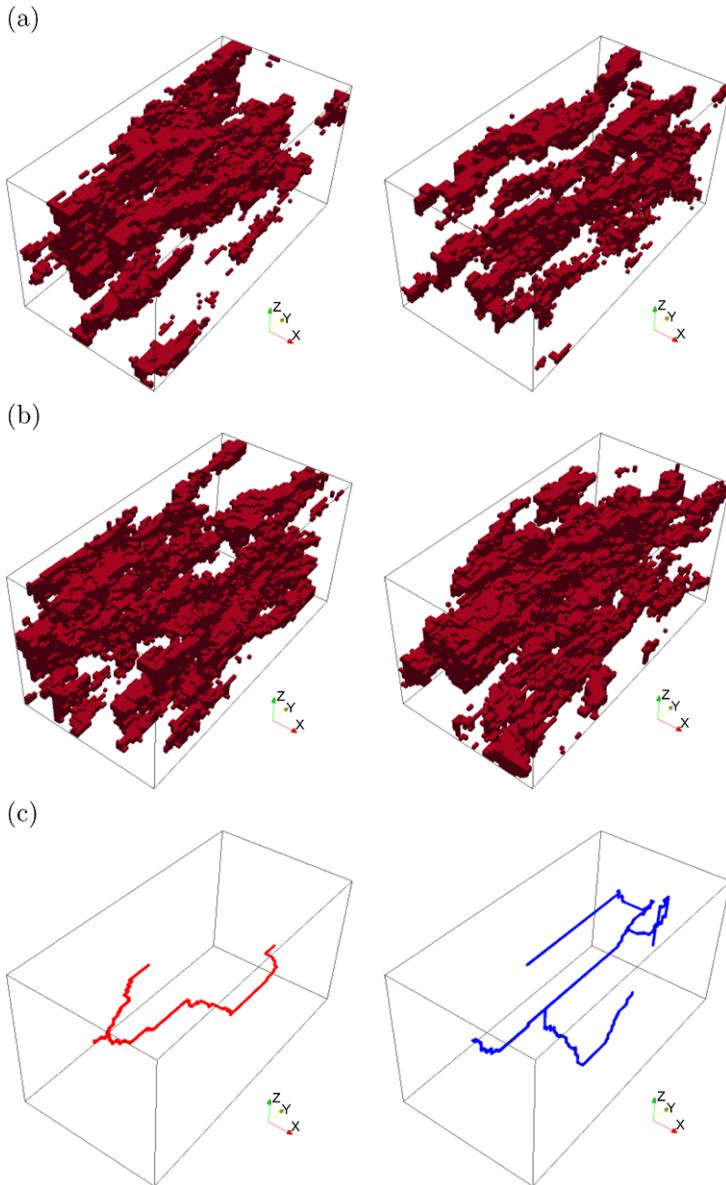


Fig. 15 Comparative 3D example for 5-point connectivity; multiple-point simulations with *impala*: (a) two realizations accounting only for conditioning data, (b) two realizations accounting both for conditioning data and for connectivity constraints (proposed algorithm). (c) Connecting paths built by the proposed algorithm (in red (left) for the simulation (b) left, in blue (right) for the simulation (b) right). Five conditioning data located at coordinates (20, 10, 30), (30, 90, 20), (15, 40, 35), (25, 25, 25), (35, 70, 15) (permeable cells) are considered in the simulation grid of size $50 \times 100 \times 50$. The training image ($200 \times 400 \times 140$) shown in Fig. 12 is used. The connectivity between conditioning data is respected in simulation (a) left and not respected in simulation (a) right

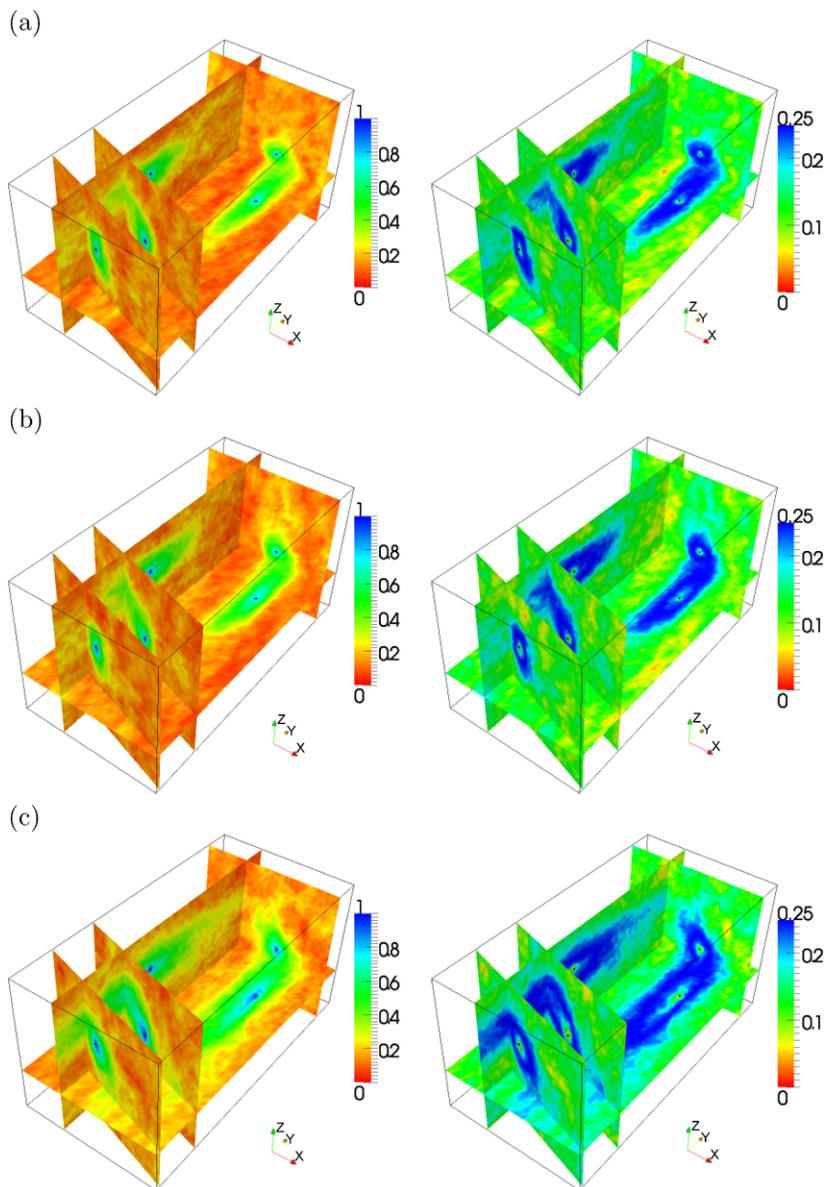


Fig. 16 Comparative 3D example for 5-point connectivity; multiple-point simulations with *impala*; E-type (left) and variance maps (right) of 100 simulations: (a) accounting only for conditioning data, (b) obtained by the rejection method, (c) by the proposed algorithm. The black points represent the location of the conditioning data. The grid size is $50 \times 100 \times 50$ cells. The training image ($200 \times 400 \times 140$) shown in Fig. 12a is used

a series of smaller problems in which stationarity can be assumed locally and by employing intermediate points located along the boundary of the domains in which we assume stationarity, or by generating unconditional simulations with the same non

stationarity, and by picking connected patterns in such a simulation instead of from the training image.

Another possible extension of the algorithm would be to use a concept of fuzzy connectivity. Indeed, for most real media, there is never a fully impervious phase. That means that it may be useful to be able to define a new type of connectivity (fuzzy connectivity) between two points by not stating that these points are within the same clusters but by stating that there is a path between these points which is faster than between other pairs of points located at the same Euclidean distance within the domain. The concept of fuzzy connectivity would allow the algorithm to work even when the constraints of 2-point connectivity are not strictly compatible with the structural model. For example, if we consider a training image representing permeable sand lenses, we would expect this algorithm to place several lenses close to each other in order to allow for a fast path between two points even if the points that should be connected are at a distance much larger than the characteristic size of one single lens. Another use of this new concept would be to extend the algorithm to the case of a continuous variable. One possible approach would be to define some typical travel time for each value of the medium. This would allow for the calculation of the travel time required between any two cells in the domain. We would then sample the fastest path or the path that has a certain duration instead of sampling a path within a geobody.

6 Conclusions

The proposed method allows for the generation of stochastic simulations that honor connectivity constraints. The connectivity is ensured by adding conditioning data prior to running the simulation algorithm. Hence, the proposed algorithm is computationally efficient and very general because it does not depend on the actual simulation technique. It can, for example, be applied without any modification to sequential indicator simulations or truncated plurigaussian methods.

We have shown that the time required to run our algorithm is significantly shorter than the time required to run a rejection algorithm. However, these differences are highly dependent on the specific configuration and more precisely on the number of points that must be connected and their probabilities of being connected. They also depend on the model of spatial continuity adopted. If the points to be connected are very close to each other, they may have a high probability of being connected, and the rejection algorithm is then very efficient (most realizations will be connected). On the contrary, if two points have a low probability of being connected, the rejection algorithm is highly inefficient and cannot yield connected simulations with a reasonable number of realizations. The connectivity function can be used to decide on the best technique to use when 2-point connectivity is considered.

Some slight differences have been observed when comparing the results obtained with our method and those obtained by the rejection method. First, some paths in the simulation grid are favored for connecting the points. Using a larger training image should reduce the differences. Moreover, the variance in the regions close to the points that must be connected is smaller when using the proposed algorithm than with the rejection method. However, it can be the opposite in other regions. Second, the

global proportion of permeable cells (in the simulation grid) is higher when using the proposed technique than with the rejection method. If the number of points that must be connected increases, the proportion of permeable cells increases too (larger bias). This is an indirect consequence of the use of a large number of conditioning data for imposing the connected paths which induces that artifact. Hence, the proposed algorithm should be used with care when the connectivity constraints require a large number of points to be connected.

Acknowledgements This work has been funded by the Swiss National Science Foundation under the contract PP02–106557. It has benefited from numerous and highly productive discussions with André Journel, Alexandre Boucher, David Ginsbourger, and Lin Hu.

References

- Alcolea A, Renard P (2010) Blocking moving window algorithm: conditioning multiple-point simulations to hydrogeological data. *Water Resour Res* 46:W08511
- Allard D (1994) Simulating a geological lithofacies with respect to connectivity information using the truncated gaussian model. In: Armstrong M, Dowd P (eds) *Geostatistical simulations: proceedings of the geostatistical simulation workshop*, Fontainebleau, France, 27–28 May 1993. Kluwer Academic, Norwell, pp 197–211
- Allard D, Heresim Group (1994) On the connectivity of two random set models: the truncated gaussian and the boolean. In: Soares A (ed) *Geostatistics Tróia '92*. Kluwer, Norwell, pp 467–478
- de Marsily G, Delay F, Gonçalvès J, Renard P, Teles V, Violette S (2005) Dealing with spatial heterogeneity. *Hydrogeol J* 13:161–183
- Deutsch C (1998) Fortran programs for calculating connectivity of three-dimensional numerical models and for ranking multiple realizations. *Comput Geosci* 24:69–76
- Gómez-Hernández J, Wen X-H (1998) To be or not to be multi-gaussian? A reflection on stochastic hydrogeology. *Adv Water Resour* 21:47–61
- Hoshen J, Kopelman R (1976) Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Phys Rev B, Condens Matter Mater Phys* 14:3438–3445
- Journel A, Alabert F (1990) New method for reservoir mapping. *J Pet Technol* 42:212–218
- Journel A, Deutsch C (1993) Entropy and spatial disorder. *Math Geol* 25:329–355
- Kerrou J, Renard P, Hendricks Franssen H, Lunati I (2008) Issues in characterizing heterogeneity and connectivity in non-multigaussian media. *Adv Water Resour* 31:147–159
- Knudby C, Carrera J (2005) On the relationship between indicators of geostatistical, flow and transport connectivity. *Adv Water Resour* 28:405–421
- Krishnan S, Journel A (2003) Spatial connectivity: from variograms to multiple-point measures. *Math Geol* 35:915–925
- Mariethoz G, Renard P, Straubhaar J (2010) The direct sampling method to perform multiple-point geostatistical simulations. *Water Resour Res* 46:W11536
- Neuweiler I, Cirpka O (2005) Homogenization of Richards equation in permeability fields with different connectivities. *Water Resour Res* 41
- Nurafza P, King P, Masih M (2006) Facies connectivity modelling: analysis and field study. In: Hanna G (ed) *SPE Europe/EAGE annual conference and exhibition*. Society of Petroleum Engineers, Vienna, p SPE 100333
- Stauffer D, Aharony A (1994) *Introduction to percolation theory*, 2nd edn. Taylor & Francis, London
- 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. doi:[10.1007/s11004-011-9328-7](https://doi.org/10.1007/s11004-011-9328-7)
- Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. *Math Geol* 34:1–21
- Western A, Blöschl G, Grayson R (1998) How well do indicator variograms capture the spatial connectivity of soil moisture? *Hydrol Process* 12:1851–1868

- Western A, Blöschl G, Grayson R (2001) Toward capturing hydrologically significant connectivity in spatial patterns. *Water Resour Res* 37:83–97
- Zhang T, Switzer P, Journel AG (2006) Filter-based classification of training image patterns for spatial simulation. *Math Geol* 38(1):63–80
- Zinn B, Harvey C (2003) When good statistical models of aquifer heterogeneity go bad: a comparison of flow, dispersion, and mass transfer in connected and multivariate gaussian hydraulic conductivity fields. *Water Resour Res* 39:1051