Paper presented at the VIII Geostatistical Congress, Santiago, Chili December 1-5, 2008. To appear in the proceedings.

CONDITIONING FACIES SIMULATIONS WITH CONNECTIVITY DATA

PHILIPPE RENARD (1) and JEF CAERS (2)

(1) Centre for Hydrogeology, University of Neuchâtel, Switzerland

(2) Stanford Center for Reservoir Forecasting, University of Stanford, USA

ABSTRACT

When characterizing and simulating underground reservoirs for flow simulations, it is frequent that field observations allow identifying specific points in space that are hydraulically connected. 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-points philosophy but does not imply necessarily the use of multiple-point simulation algorithm. The aim is to generate realizations such that the connectivity information is honoured as well as any prior structural information (e.g. as modelled through a training image). The algorithm consists in 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 locations of point 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.

INTRODUCTION

An important feature when characterizing underground reservoirs with respect to their flow properties is the connectivity of the high or low permeable structures (Zinn and Harvey 2003). Failing to capture subsurface connectivity may bias the forecasts of any underground fluid flow and transport modelling project (Journel and Alabert 1990; Gómez-Hernández and Wen 1998, Knudby and Carrera 2005). However, little effort has yet been devoted to simulate directly parameter fields that honour a connectivity constraint such has the one imposed by a tracer test. Usually the connectivity information (breakthrough of a tracer for example) is used in the framework of an inverse problem (e.g.

Hyndman et al. 1994). The only existing direct method to our knowledge was proposed by Allard (1994), it is general and has been applied in the context of a truncated Gaussian model. The principle is to run an iterative Gibbs sampler that modifies an initial field in which the connectivity constraints has been imposed. The Gibbs sampler allows obtaining after a number of iterations a field that respects both the structure and the connectivity (or disconnectivity) information. One issue is that the iterative procedure may be slow, and that it may not converge if the covariance function is incompatible with the connectivity constraint. Note that this is not a limitation of the method but a problem related to the selection of a covariance model consistent with the connectivity constraints.

In this paper, we propose an alternative direct algorithm that is based on the idea of borrowing a connected path from a training image instead of iteratively building a simulation that satisfy 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.

DEFINITION OF TWO-POINT CONNECTIVITY

Assume for simplicity that the domain Ω of interest is partitioned by a regular Cartesian grid. Each cell represents an elementary volume (Δx Δy Δz). In each cell, there are only two types of porous media represented by an indicator variable $I(\boldsymbol{x})$ that can either take the value 1 (the cell is permeable) or 0 (the cell is impermeable). $I(\boldsymbol{x})$ can be deterministic, or can be one realization of a random function. We define a point or a cell in the domain by its location $\boldsymbol{x}$ which is a vector of three spatial coordinates. Two cells $\boldsymbol{x}$ and $\boldsymbol{y}$ in Ω are said to be connected (notation: $\boldsymbol{x} \leftrightarrow \boldsymbol{y}$) if it exists at least one path of adjacent grid cells that allows to go from $\boldsymbol{x}$ to $\boldsymbol{y}$ 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 3D) or corners (2D and 3D).

When the function $I(\boldsymbol{x})$ is known, a computationally efficient way of checking that any two cells are connected is to construct a cluster identification function $C(\boldsymbol{x})$ that identifies with a unique value every group of adjacent cells. The concept of 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 $C(\boldsymbol{x})$ is equal to zero for all the non permeable cells and is equal to a constant integer value for each cluster. The value $C(\boldsymbol{x})$ is different for each cluster; it is the identifier of this cluster. The details of an efficient computer implementation of the calculation of $C(\boldsymbol{x})$ from $I(\boldsymbol{x})$ are given in Hoshen and Kopelman (1976). Once the function $C(\boldsymbol{x})$ is defined, testing the connectivity between two cells $\boldsymbol{x}$

and y is equivalent to test if $C(\boldsymbol{x})$ is equal to $C(\boldsymbol{y})$. This approach is extremely efficient to test the connectivity between many cells.

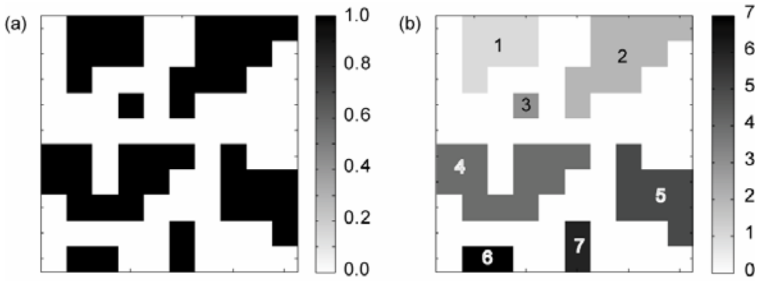

Figure 1*:* Example of a binary medium. (a) the indicator function I(x) representing the two phases. (b) the cluster identification function C(x) that allows to identify the different connected parts.

SIMULATION ALGORITHM

Basic Algorithm

The input data are:

- a pair of connected points ($\boldsymbol{x^1}, \boldsymbol{x^2}$) located on the simulation grid G_s,
- a spatial model of the random function $I(\boldsymbol{x})$

The spatial model is defined here in a broad sense. It includes either a theoretical model of a random function with all the necessary parameters (covariance for example) but includes also non parametric models such as those used in the multiple point statistics framework with again all the relevant information (a training image for example). The aim of the proposed algorithm is then to simulate an indicator variable $I(\boldsymbol{x})$ over G_s with the prescribed model and such that $\boldsymbol{x^1}$ and $\boldsymbol{x^2}$ are connected ($\boldsymbol{x^1} \leftrightarrow \boldsymbol{x^2}$).

The first steps consist in initializing the algorithm:

- creates with the prescribed model a simulation of the indicator function $I(\boldsymbol{x})$ over a large grid G_t. This image is used as a training image (Fig. 2a) in the next steps.
- compute the lag vector $\boldsymbol{d} = \boldsymbol{x^1} - \boldsymbol{x^2}$
- compute the cluster function $C(\boldsymbol{x})$ on G_t. It allows defining all the connected bodies existing in the training image.
- scan the training image grid G_t to find all the grid cells $\boldsymbol{z}_i$ such that $C(\boldsymbol{z}_i)=C(\boldsymbol{z}_i+\boldsymbol{d})$. In other words, we look for all the locations in the training image which feature $\boldsymbol{x^1} \leftrightarrow \boldsymbol{x^2}$. The number n of such replicates is stored, as well as the location of all the values $\boldsymbol{z}_i$ (i=1 ... n).
- 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).

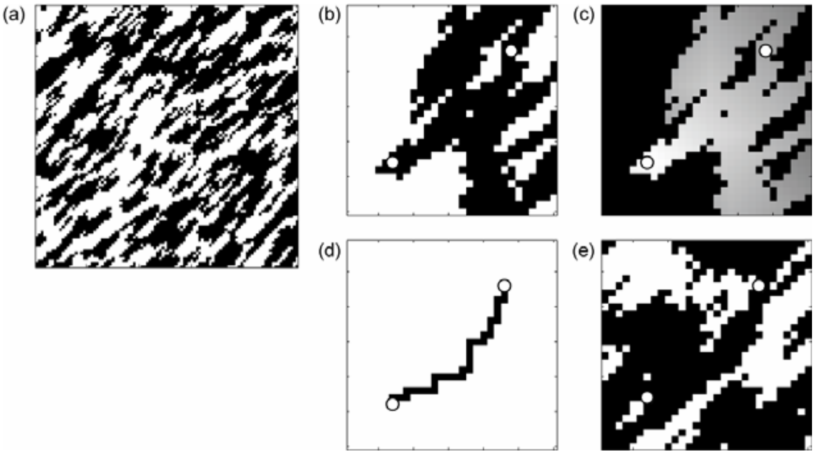

Figure 2*:* Step by step description of the base algorithm. (a) is a training image (100 x 100 cells), it has been generated by sequential indicator simulation with a spherical variogram, an anisotropy oriented at 45 degrees, with variogram ranges equal to 10 and 3. (b) is an image of the simulation grid (30 x 30 cells) after one replicate of a connected body has been translated and pasted in the simulation. The two white disks represent the points that must be connected. (c) represents the distance function which is calculated inside the geobody to draw the random paths. (d) shows one possible path. (e) is the final results of the simulation.

Once the initialization is finished, the following steps are applied to generate a realization conditioned to the connectivity data:

- A random integer value is chosen between 1 and n. This value corresponds to the random selection of one replicate of a connected pattern (all the replicates have the same probability to be chosen).
- The entire geobody/cluster is identified by $C(\mathbf{z})=C(\mathbf{z}_i)$ and copied from G_t to G_s (Fig 2b). Its location is defined such that $\mathbf{z}_i$ in G_t corresponds to $\mathbf{x}$ in G_s.
- Because G_t is generally larger than G_s, it is possible that a large cluster connected in G_t is not anymore connected in G_s. Therefore, we must check if $\mathbf{x}^1$ and $\mathbf{x}^2$ are really connected. If it is not the case, we need to start over with another replicate. To avoid this situation later in the process, we remove $\mathbf{z}_i$ from the list of replicates and reduce the number of replicates to n-1.
- Because the copied geobody may occupy a large portion of the simulated image (see Fig 2b), keeping it in its entirety in the simulation grid would significantly limit spatial variability within G_s. Instead, we build a random path within the replicate and set all the cells along the path to 1 (Fig 2d). Because we consider that the paths go through the faces of grid cells, there are many paths of exactly the same length allowing to go from

$\boldsymbol{x^1}$ to $\boldsymbol{x^2}$ while remaining within the connected body. Because we want to minimize the number of pixels that are imposed during that stage of the simulation (to allow the maximum of variability later on) we select paths whose length is the minimum distance between $\boldsymbol{x^1}$ and $\boldsymbol{x^2}$. This is done with a random walker that starts at point $\boldsymbol{x^1}$ and walks down along a slope whose altitude is the distance to point $\boldsymbol{x^2}$ (Fig 2c).

- All these cells are taken as hard conditioning data for the simulation algorithm which is applied as usual, whatever the technique and the model (Fig 9e).

Accounting for Hard Data

The data set may contain hard data locations that must be taken into account to condition locally the simulations in addition to the connectivity constraints. We must ensure that the simulation satisfies the usual conditioning constraints plus the connectivity constraints. Two approaches are proposed to extend the algorithm to this situation. The first approach, which ensures strictly the consistency between the model and the simulation, is to scan the training image for replicates of the complete set of constraints, namely those that satisfy both the connectivity conditions and the local values. We then keep only the replicates that satisfy this criterion and proceed as it has been described above. One problem is that the number of replicates that satisfy all conditions decreases rapidly when the number of conditioning points increases. We then need a very large training image to be able to apply this technique which is not practical.

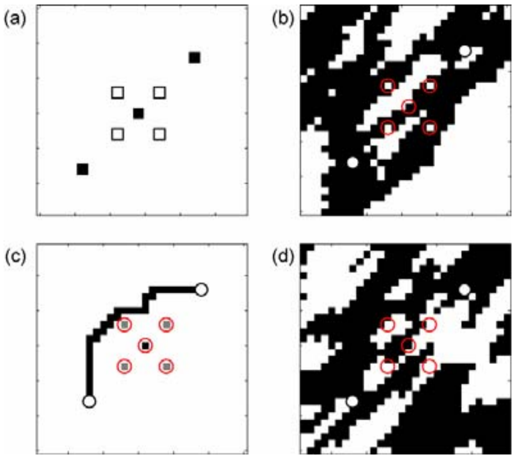

Figure 3*:* Simplified example showing the procedure to insure local conditioning and connectivity conditioning. (a) shows the location of the local conditioning data. The black squares are the permeable phase. The lower left and upper right points are the points that must be connected. The black point in the center is not informed in terms of connectivity. (b) one replicates of a connected cluster has been pasted in the simulation grid and « holes » have been drilled at the location of the non permeable conditioning data. The position of the data is indicated by the red circles. (c) the selected path and (d) one simulation conditional to local and connectivity data.

The second approach (Figure 3) consists of modifying the values of the replicate (connected geobody) locally where conditioning data are available. This is done before selecting the random path. Figure 3a shows, as an example, a set of conditioning data. It contains 3 permeable cells and 4 non permeable ones. The aim is to connect the two extreme ones on the diagonal. After having scanned the training image, one replicate of a connected geobody is copied on the simulation grid, but without modifying the values of the conditioning data (Fig 3b) that were already pasted on the grid. This ensures that the non-permeable cells within the conditioning dataset remain non permeable. The path naturally avoids these cells (Fig 3c) and allows creating a simulation (Fig 3d) 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 corresponds to a situation similar to the replicates that did connect on G_t but not in G_s. In those cases, the replicate needs to be removed from the list, and the procedure has to start over with a new replicate. To conclude that part, one could argue that such modification may create local inconsistencies in the simulation (e.g. placing a permeable path in the middle of impermeable cells). This cannot be avoided but considering that usually the amount of conditioning data is small as compared to the amount of cells to be simulated, this phenomenon should not be too problematic.

EXAMPLE APPLICATION

Several tests have been made to investigate the performances of the algorithm in 2D and 3D. One such example is shown in figure 4. In that case, the simulations are made with snesim using the classical training image consisting of a set of hand-drawn channels made by Strebelle (2002) and two conditioning data points located on the middle of two opposite faces of the simulation grid (white circles in Fig 4b). Two simulations obtained by snesim without imposing the connectivity constraints are displayed in Figs 4b and 4c. One is connected, the other not. In average, most of those simulations are not connected. The E-type (map of local mean) of 100 such simulations is shown in Fig 4a. Figs 4e and 4f show two simulations obtained with the same conditioning data plus the connectivity constraint. All those simulations are connected as expected. The corresponding E-type (Fig 4d) shows how the connectivity constraints has reduced the uncertainty between the two conditioning data points and reveals now the presence of several possible channels. In that case, connectivity is extremely interesting because it provides not only local information. Imposing the connectivity between two rather distant points provides a strong control in between those points without knowing the details of the actual geometry.

To illustrate the impact of the connectivity constraints on flow and solute transport, the fields described in the previous section were used as input data for a finite element model. The permeability contrast between the permeable and impermeable phase is fixed to a moderate value of 100. The porosity is homogeneous and equal to 0.1, the longitudinal dispersivity is equal to 1m, and the lateral dispersivity is set to 0.1m. Constant heads are prescribed on the right and left sides of the grid in order to impose a uniform flow through the domain.

The mean and standard deviations of the computed head fields are shown in Figure 5. One can see on that figure, that the mean head fields are identical whether one imposes or not a connectivity constraint. This is not surprising considering the moderate permeability contrast and the small sensitivity of the pressure equation to heterogeneity parallel to the main flow direction. Still, we observe a slight difference in the central part of the standard deviation maps indicating a slight reduction of uncertainty in that zone when imposing the connectivity constraint (Fig. 5d).

Then, a pulse of solute (dimensionless concentration=1) is released on the right side of the domain in the conditioning point. The advective dispersive transient transport equation is solved using finite elements in the Laplace domain (Cornaton, 2006). The solute concentration at the outlet is monitored in the conditioning point located on the left side of the domain. Because the matrix is not considered purely impermeable, the solute can leave the channels, flow through the matrix and reach the outlet point even if the inlet and outlet points are not connected by a continuous channel. The average breakthrough curves for the 100 simulations with or without connectivity constraints are displayed in Figure 6. As expected, one can see that even if the head fields were extremely similar in terms of heads, the breakthrough curves of the solute are completely different for the simulations with or without connectivity. When the permeability contrast is increased the difference becomes even larger.

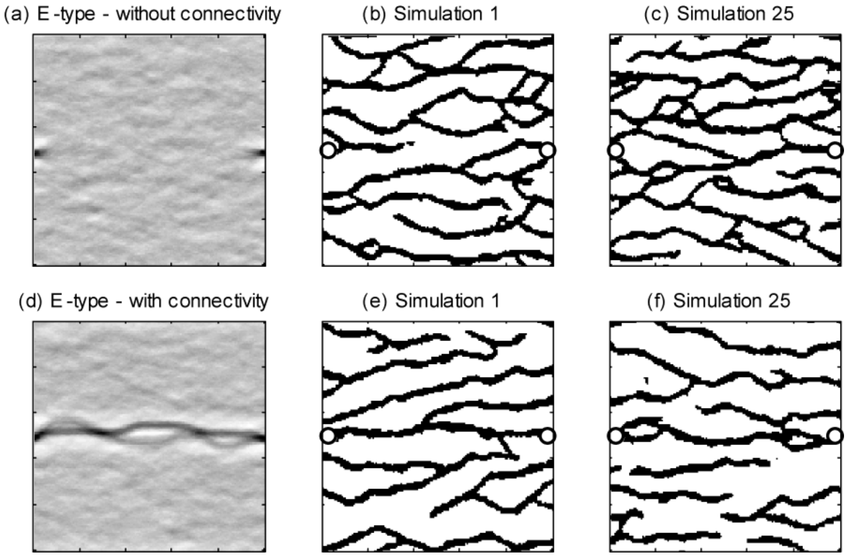

Figure 4*:* Comparative example of snesim simulations accounting (a,b,c) only for conditioning data at the wells, or (d,e,f) accounting both from conditioning data and connectivity constraints. The two white circles represent the location of the conditioning data. The E-type is computed on 100 simulations, the black corresponds to a probability of one and the white to a probability of zero.

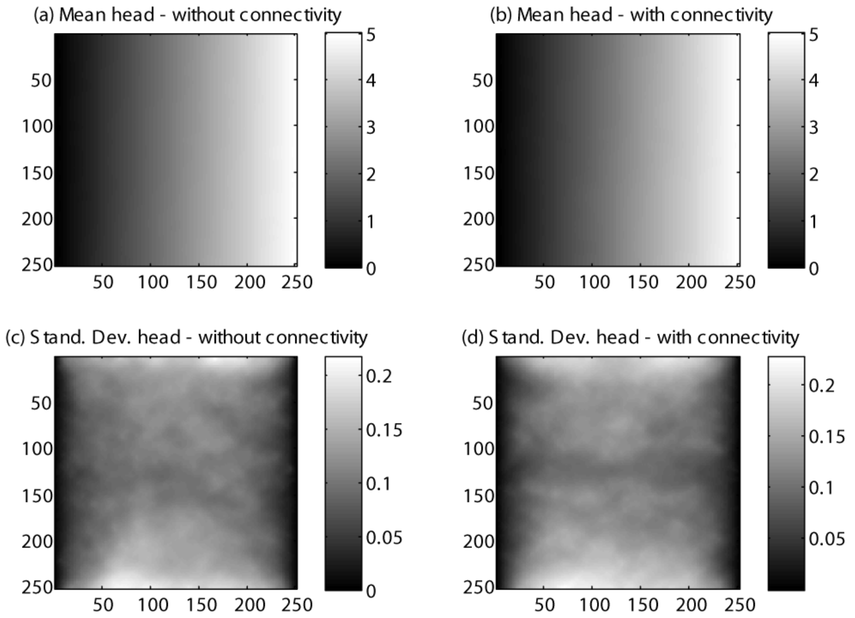

Figure 5: Results of the flow simulations (a) mean head map for the simulation without connectivity constraints, (b) mean head map with connectivity constraints, (c) standard deviation of the heads without connectivity constraints, (d) standard deviation of heads with connectivity constraints.

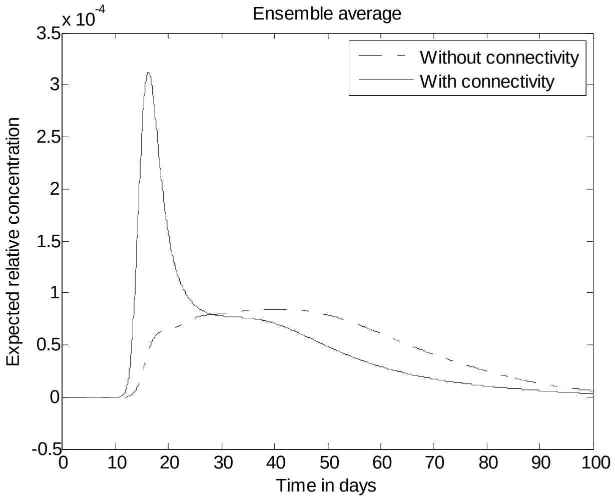

Figure 6: Mean concentration at the outlet point computed with or without connectivity constraints.

DISCUSSION AND CONCLUSION

The work presented in this paper shows that it is possible to generate rapidly a wide range of stochastic simulations that honour the two points connectivity constraints. The algorithm is simple and fast.

An issue that remains to be solved is that the selection of a short path creates a bias. By comparing several sets of connected simulations obtained by a rejection algorithm and several sets of simulations obtained by the proposed algorithm, we observe that there are cases in which the E-type are clearly different. The proposed algorithm over-samples the shortest paths. To avoid this problem, we propose to use the fast marching approach as described in Hovadik and Larue (2007) to compute an ensemble of possible paths.

In order to make the algorithm really applicable, there is still a need to extend it to multiple connectivity and disconnectivity constraints. A multiple connectivity constraint consists of a list of several sets of points that belong to different clusters (that can be known to be disconnected or not). To connect a set of points instead of two points, the simplest idea is to decompose the set into a series of pairs of points and simply apply the algorithm that was presented above for every pair. No simple solution has yet been found for the disconnection problem, i.e. imposing that two points do not belong to the same cluster. A possible algorithm is to ensure iteratively that no connection is made during the simulation process.

Along the same line of thought, there is a need to extend the algorithm to the case of non-stationary simulations. One way to account for non-stationarity is to consider different zones of different orientations. Connecting two points belonging to two different zones could be achieved by splitting the problem in each region and connecting successive pair of points. The intermediate points would be placed along the boundary between the regions in such a way as to maximize the probability of allowing a connection.

Concerning the computational efficiency, the time required to run the algorithm is significantly shorter than the time required to run a rejection algorithm if the two points to connect are distant. However the cpu gain is highly dependent on the specific configuration and more precisely on the probability of having a connected path between the two points. If the two points are very close, they have a high probability to be connected, and then the rejection algorithm is efficient. On the opposite, when the points have a low probability of being connected, the rejection algorithm is not efficient. In one of our tests, two points could not get connected with the rejection algorithm after having generated 400 simulations, while all the simulations made with the proposed algorithm were easily connected.

Finally, the 2D example used to illustrate the methodology emphasizes that connectivity has a high impact on solute transport even for moderate permeability contrasts. In terms of history matching, or inversion, the connectivity constraints should be used whenever possible to reduce the space in which optimal solutions are searched.

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. Special thanks to Alexandre Boucher, Nicolas Remy, and Ting Li for their support in coding the Plug-In within SGeMS.

REFERENCES

Allard, D. (1994) *Simulating a geological lithofacies with respect to connectivity information using the truncated Gaussan mode,* in Geostatistical simulations: Proceedings of the geostatistical simulation workshop, Fontainebleau, France, 27-28 May 1993. (eds. M. Armstrong, and P.A. Dowd), Kluwer, p. 197-211.

Allard, D., and Heresim Group. (1994) *On the connectivity of two random set models: The truncated Gaussian and the Boolea,* in Geostatistics Tróia '92. (ed. A. Soares), Kluwer, p. 467-478.

Cornaton, F. (2006) *GroundWater, A 3-D Ground water flow and transport finite element simulator. Technical Report*, Technical report, University of Neuchâtel, Switzerland.

Deutsch, C.V. (1998) *Fortran programs for calculating connectivity of three-dimensional numerical models and for ranking multiple realizations*, Computers and Geosciences, vol. 24, no. 1, p. 69-76.

Gómez-Hernández, J.J. and Wen, X.-H. (1998) *To be or not to be multi-gaussian? A reflection on stochastic hydrogeology*, Advances in Water Resources, vol. 21, no. 1, p. 47-61.

Hoshen, J., and Kopelman, R. (1976) *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm*, Physical Review B, vol. 14, no. 8, p. 3438-3445.

Hovadik, J.M. and Larue, D.K. (2007) *Static characterizations of reservoirs: refining the concepts of connectivity and continuity,* Petroleum Geoscience, vol. 13, no. 3, p. 195-211

Hyndman, D.W., Harris, J.M. and Gorelick, S.M. (1994) *Coupled seismic and tracer test inversion for aquifer property characterization*, Water Resources Research, vol. 30, no. 7, p. 1965-1977.

Journel, A.G. and Alabert, F.G. (1990) *New method for reservoir mapping*, Journal of Petroleum Technology, vol. 42, no. 2, p. 212-218.

Knudby, C. and Carrera, J. (2005) *On the relationship between indicators of geostatistical, flow and transport connectivity*, Advances in Water Resources, vol. 28, no. 4, p. 405-421.

Stauffer, D. and Aharony, A. (1994) *Introduction to percolation theory*, *Second ed.*, Taylor and Francis.

Strebelle, S. (2002*) Conditional simulation of complex geological structures using multiple-point statistics*, Mathematical Geology, vol. 34, no. 1, p. 1-21.

Zinn, B. and Harvey, C.F. (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 Resources Research, vol. 39, no. 3, p. 1051.