Mathematical Geology, Vol. 34, No. 1, January 2002 (© 2002)

Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics[1]

Sebastien Strebelle[2]

In many earth sciences applications, the geological objects or structures to be reproduced are curvilinear, e.g., sand channels in a clastic reservoir. Their modeling requires multiple-point statistics involving jointly three or more points at a time, much beyond the traditional two-point variogram statistics. Actual data from the field being modeled, particularly if it is subsurface, are rarely enough to allow inference of such multiple-point statistics. The approach proposed in this paper consists of borrowing the required multiple-point statistics from training images depicting the expected patterns of geological heterogeneities. Several training images can be used, reflecting different scales of variability and styles of heterogeneities. The multiple-point statistics inferred from these training image(s) are exported to the geostatistical numerical model where they are anchored to the actual data, both hard and soft, in a sequential simulation mode. The algorithm and code developed are tested for the simulation of a fluvial hydrocarbon reservoir with meandering channels. The methodology proposed appears to be simple (multiple-point statistics are scanned directly from training images), general (any type of random geometry can be considered), and fast enough to handle large 3D simulation grids.

KEY WORDS: geostatistics, stochastic simulation, training image, random geometry.

INTRODUCTION

This section briefly recalls the limitations of traditional simulation approaches based on two-point statistics, and reviews present approaches which account for multiple-point (mp) information. Other introductions to multiple-point (geo)statistics can be found in Farmer (1988), Deutsch (1992), Journel (1997), and Srivastava (1992, 1995).

Curvilinear geometries, such as sinuous channels in a fluvial reservoir or incised valleys over a topography, cannot be modeled using only traditional two-point statistics such as a variogram. Reproduction of such random geometries calls for the parametrization of specific shapes or the consideration of the joint categorical variability at three or more points at a time. This is the reason why

[1]Received 30 March 2000; accepted 2 January 2000.

[2]Department of Geological and Environmental Sciences, Stanford University, Stanford, California 94305-2115; e-mail: strebell@pangea.stanford.edu

specific geometries are poorly reproduced by traditional pixel-based algorithms, such as indicator or Gaussian truncated simulation techniques which succeed at reproducing only proportions and two-point (cross)variograms.

The most straightforward way to reproduce crisp geometries is to parametrize their shapes, e.g. curvilinear "fettucinis" to model meandering channels, then randomize these shape parameters and use Boolean object-based algorithms for dropping these random geometries over the volume to be simulated (Bridge and Leeder, 1979; Haldorsen and Damsleth, 1990; Omre, 1991). There are two limitations to such object-based approach: (1) each class of objects requires its own specific parametrization and not all geomorphological heterogeneities can be summarized by a few geometric parameters, (2) the conditioning to local data of these random objects can be difficult, particularly if the data are dense with regard to the average object size.

Coming back to the pixel-based alternative, Xu (1996) addresses the challenge of curvilinearity by presimulating local directions and ratios of geometric anisotropy. The local kriging systems are then adapted to these locally varying anisotropy characteristics, resulting in the simulation of changing anisotropy directions rather than curvilinear features. The technique being limited to reproduction of two-point statistics fails to yield crisp continuous and curvilinear geometries. Also Xu does not address the theoretically challenging problem of joint simulation of a 3D stochastic direction defined by up to 3 related angles (azimuth, dip, and rake).

Some of the pixel-based methods which use directly mp information to simulate curvilinear structures are

- Simulated annealing, where mp statistics inferred from a training image are incorporated in the objective function (Deutsch, 1992; Farmer, 1988). However, the number of components in the objective function increases with the number of mp statistics imposed, consequently CPU and RAM demand quickly increases and convergence becomes problematic.
- Markov Chain Monte Carlo (MCMC) simulation calls for a prior specification of either the mp probability distribution of the variable to be simulated, or some ratio of conditional probability values given a mp data event (Tjelmeland, 1996). An analytical definition of such mp probability distributions typically requires severe model approximations such as data screening, conditional independence and/or arbitrary Gaussian distributions with their congenial properties. In this regard, the MCMC implementation of Caers and Journel (1998) is notable in that it is free of most above model assumptions because all required probability values are modeled from training images.
- Iterative simulation algorithms based on a form of Gibbs sampler were proposed by Srivastava (1992, 1995). Similar to the MCMC algorithms,

they consist of iterative perturbations of a prior image one pixel at a time, the perturbation ensuring that the mp statistics of a template centered on the pixel being perturbed approximate the statistics inferred from a training image.

All previous algorithms are iterative: the numerical model being built is perturbed over multiple visits of each of its pixels, and they rely on convergence criteria. If convergence is sometimes ensured by the properties of the Markov chain used, the rate of convergence is rarely known a priori and somewhat arbitrary stopping criteria must be implemented.

The paper by Guardiano and Srivastava (1993) represents a milestone in that it suggests a direct (noniterative) algorithm for imposing mp statistics into stochastic simulation. This algorithm consists of a sequential indicator simulation where all required conditional probabilities are identified to corresponding proportions read from training images. Guardiano and Srivastava's idea is remarkable for its extreme simplicity: no prior modeling of mp statistics or variogram is required, nor any kriging to derive the conditional probabilities; these probabilities are obtained directly by scanning the training image(s). However, although this algorithm is noniterative and hence does not suffer from convergence considerations, the corresponding original code was extremely CPU demanding: the full training image had to be scanned anew at each unsampled node to infer the node-specific conditional probability distribution. The algorithm proposed in this paper is but an extension of the seminal work of Guardiano and Srivastava.

Experience with stochastic simulation of categorical variables (e.g. facies types or classes of a continuous variable) points towards developing new algorithms which would combine the flexibility and easy data-conditioning of pixel-based algorithms with the ability to reproduce "shapes" of object-based algorithms, without being too CPU and RAM demanding. Ideally such an algorithm should be pixel-based, include mp statistics allowing shape reproduction, be fast hence noniterative, and be general in that a new program need not be written to accommodate any new random geometry.

Terminology

Consider an attribute S taking K possible states $\{s_k, k = 1, \ldots, K\}$. S can be a categorical variable, or a continuous variable with its interval of variability discretized into K classes by $(K-1)$ threshold values. A data event d_n of size n centered at a location $\mathbf{u}$ to be simulated is constituted by

- a data geometry defined by the n vectors $\{\mathbf{h}_\alpha, \alpha = 1, \ldots, n\}$
- the n data values $s(\mathbf{u} + \mathbf{h}_\alpha) = s(\mathbf{u}_\alpha), \alpha = 1, \ldots, n$

The value at the center of that data template is the unknown value $s(\mathbf{u})$.

A data template τ_n comprises only the previous data geometry. A subtemplate of τ_n is a template constituted by any subset n' of vectors of τ_n, with $n' \leq n$. The data event d_n is said to be "associated" with the geometric template τ_n.

Preliminary Remark

In a stochastic mode, the K possible outcomes of the random variable $S(\mathbf{u})$ are characterized by their conditional probability distribution function (cpdf) denoted as:

$$\text{Prob}\{S(\mathbf{u}) = s_k \mid d_n\} = f(\mathbf{u}; k \mid d_n), \quad k = 1, \ldots, K \tag{1}$$

Knowledge of that cpdf for any data event d_n suffices to generate by Monte Carlo simulation realizations of the random variable $S(\mathbf{u})$. Note, however, that such knowledge is a tall proposition, indeed, if each of the n data variables can take K different values, the total number of different data events for a given data geometry is K^n; e.g. $K = 10$ decile classes and $n = 10$ lead to $K^n = 10^{10}$, a huge number!

In the practice of sequential simulation, the previous cpdf is estimated at each node $\mathbf{u}$ by some form of kriging, either kriging of the normal score transform of the continuous variable $S(\mathbf{u})$, or kriging of each of the K class indicators of the categorical variable $S(\mathbf{u})$ (Deutsch and Journel, 1998, p. 175).

An alternative to kriging would be to borrow the previous cpdf from a training image T by scanning it for replicates of the data event d_n. Suppose that J such replicates are found, the histogram of the J central values $s_T(\mathbf{u}_j)$, $j = 1, \ldots, J$, can be used as a proxy for the cpdf $f(\mathbf{u}; k \mid d_n)$. The problem is that no finite training image would be large enough to provide enough replicates for each of the K^n possible outcomes of the data event d_n, if $K^n = 10^{10}$. These are two avenues of solutions around this inference problem:

1. The traditional modeling route whereby the cpdf $f(\mathbf{u}; k \mid d_n)$ is modeled by some, preferably mp function of the n data values $s(\mathbf{u}_\alpha)$. Instead of some form of kriging, Caers and Journel (1998) utilize a neural network to fit a mp parametric cpdf to the few (much lesser than K^n!) experimental proportions found over the training image.
2. Drastically reduce the total number K^n by making K small, down to $K \leq 4$. Then only those proportions corresponding to the data events d_n actually found over the training image are utilized directly as cpdf values without any prior modeling.

The last alternative pioneered by Guardiano and Srivastava (1993) is that retained hereafter.

A SINGLE NORMAL EQUATION

The key to any sequential simulation algorithm is the cpdf (1). In traditional two-point algorithms, the conditioning is considered one datum at a time through some measure of the two-point correlation between $S(\mathbf{u})$ and $S(\mathbf{u}_\alpha)$, for example an indicator covariance model. Instead, we suggest considering jointly the n data of the conditioning data event d_n, which requires a $(n+1)$-point covariance to measure the dependence of $S(\mathbf{u})$ on the data event d_n.

Denoted by A_k the binary (indicator) random variable associated to the occurrence of state s_k at location $\mathbf{u}$:

$$A_k = \begin{cases} 1 & \text{if } S(\mathbf{u}) = s_k \\ 0 & \text{if not} \end{cases}$$

Similarly, let D be the binary random variable associated to the occurrence of the data event d_n constituted by the n conditioning data $S(\mathbf{u}_\alpha) = s_{k_\alpha}, \alpha = 1, \ldots, n$, considered jointly:

$$D = \begin{cases} 1 & \text{if } S(\mathbf{u}_\alpha) = s_{k_\alpha}, \forall \alpha = 1, \ldots, n \\ 0 & \text{if not} \end{cases}$$

If the $(n+1)$-point statistics relevant to A_k and its data event D are available, then the **exact** conditional probability is given by the simple kriging expression (Journel, 1993):

$$\text{Prob}\{A_k = 1 \mid D = 1\} = E\{A_k\} + \lambda[1 - E\{D\}] \tag{2}$$

where $D = 1$ is the observed data event, $E\{D\} = \text{Prob}\{D = 1\}$ is the probability for the conditioning data event to occur, $E\{A_k\} = \text{Prob}\{S(\mathbf{u}) = s_k\}$ is the prior probability for the (unknown) state at $\mathbf{u}$ to be s_k; this probability is "prior" to knowledge of the data event $D = 1$.

The single extended normal (kriging) equation providing the single weight λ is written as

$$\lambda \, \text{Var}\{D\} = \text{Cov}\{A_k, D\} \tag{3}$$

where $\text{Cov}\{A_k, D\} = E\{A_k D\} - E\{A_k\}E\{D\}$ is a $(n+1)$-point statistics.

Then, as per the kriging Equation (3):

$$\lambda = \frac{E\{A_k D\} - E\{A_k\}E\{D\}}{E\{D\}\,(1 - E\{D\})}, \text{ leading to the solution:}$$

$$\begin{aligned} \text{Prob}\{A_k = 1 \mid D = 1\} &= E\{A_k\} + \frac{E\{A_k D\} - E\{A_k\}E\{D\}}{E\{D\}} \\ &= \frac{E\{A_k D\}}{E\{D\}} = \frac{\text{Prob}\{A_k = 1, D = 1\}}{\text{Prob}\{D = 1\}} \end{aligned} \tag{4}$$

That exact solution identifies the definition of the conditional probability, as given by Bayes' relation.

Scanning the Training Image

The exact solution (4) calls for $(n+1)$-point statistics much beyond the traditional two-point variogram or convariance model. There is usually no hope to be able to infer such mp statistics from actual sample data, hence the idea to borrow them by scanning one or several training images under a prior decision of stationarity (export license):

- The denominator $\text{Prob}\{S(\mathbf{u}_\alpha) = s_{k_\alpha}, \alpha = 1, \ldots, n\}$ of expression (4) can be inferred by counting the number $c(d_n)$ of replicates of the conditioning data event $d_n = \{S(\mathbf{u}_\alpha) = s_{k_\alpha}, \alpha = 1, \ldots, n\}$ in the training image(s). A replicate should have same geometric configuration and same data values.
- The numerator $\text{Prob}\{S(\mathbf{u}) = s_k \text{ and } S(\mathbf{u}_\alpha) = s_{k_\alpha}, \alpha = 1, \ldots, n\}$ is obtained by counting the number $c_k(d_n)$ of replicates, among the c previous ones, associated to a central value $S(\mathbf{u})$ equal to s_k.

The required conditional probability is then identified to the training proportion $c_k(d_n)/c(d_n)$:

$$p(\mathbf{u}; s_k \mid (n)) = \text{Prob}\{A_k = 1 \mid D = 1\} = \text{Prob}\{S(\mathbf{u}) = s_k \mid (n)\} \simeq \frac{c_k(d_n)}{c(d_n)} \tag{5}$$

The limit case of a single extended normal (kriging) Equation (3) is absolutely straightforward in that it reduces to the very definition of a conditional probability, which is obtained by scanning the training image with the conditioning data template. The intermediary step of modeling permissible functions to sample statistics, e.g. a covariance model, is completely shortcut. The conditional probability (4) is, by definition, permissible, i.e. it does not suffer any of the order relation problems associated to determining an estimate of it through a kriging limited to two-point statistics. Note, however that expression (4) is "exact" only up to the decision of

stationarity which allows exporting mp statistics from the training image to the actual phenomenon under study.

An important aspect of the extended normal equation approach relates to which mp statistics should be retained to condition the simulation of any particular node $\mathbf{u}$. The larger the data neighborhood, the larger the size n of the data event d_n, the more specific this data event, hence the fewer replicates of it will be found over the training image(s) for inference of the corresponding conditional probability $p(\mathbf{u}; s_k \mid (n))$. Also, if d_n is too specific, the probability distribution $\{p(\mathbf{u}; s_k \mid (n)), k = 1, \ldots, K\}$ may be too specific to the training image(s) retained, hence exporting it to the model may be questionable: we may be exporting idiosyncratic patterns of the training image instead of its essence. The approach proposed hereafter consists of always inferring the full multiple $(n + 1)$-point statistics from the training image, reducing progressively the size n until the data event d_n retained is found "often enough" in the training image(s). This implies that the training image has a repetitive character, i.e. the geometrical shapes to be reproduced are repeated several times, say 10–20 times, in the training image. The minimal number $c_{\min}$ of replicates of the conditioning data event d_n to be found in the training image for the corresponding cpdf to be retained is then related to the repetitiveness of the training image, hence $c_{\min}$ may be, e.g., 10 or 20. Consequently the stationarity decision which allows exporting mp statistics from the training image refers to the essential (i.e. commonly found) features of the training image.

A critical aspect of this approach is its total dependence on the training image(s) used. The structures are borrowed directly from that training image, without any modeling or filtering, and anchored to the actual data. The training image is used here as a replacement for the implicit mp distribution (spatial law) underlying any mapping algorithm, whether deterministic as simple as hand-contouring, or stochastic. One could argue that a training image has the advantage of making fully explicit the structural information being exported to the mapping of the actual phenomenon. That prior information can be easily evaluated, then accepted or rejected, whereas it is much more difficult to evaluate the appropriateness of, say, a Gaussian or Gaussian-related random function model.

Training images depict the patterns of geological heterogeneities deemed relevant to the application under study. They need not carry any locally accurate information on the actual phenomenon; they merely reflect a prior geological/structural concept. Thus, a training image can be an unconditional realization generated by an object-based algorithm, or a simulated realization of an analogous field, or simply a geologist's sketch processed with CAD algorithms and properly digitized.

THE SNESIM ALGORITHM

We chose to name the simulation algorithm presented hereafter *snesim* to insist that it involves only one single normal equation (sne), which leads to the very

expression (4) of a conditional probability. The *snesim* algorithm has been developed to simulate categorical attributes, e.g. geological facies, but can be extended to simulate continuous attributes discretized into a small finite number K of classes.

The *snesim* algorithm is based on the sequential simulation paradigm whereby each simulated value becomes a hard datum value conditioning the simulation of nodal values visited later in the sequence (Goovaerts, 1997, p. 376). Because the local conditioning data event includes previously simulated nodes, and nodes are visited along a random path, the geometry of that data event changes from one node to the other. Guardiano and Srivastava (1993) proposed scanning the complete training image anew at each unsampled node to infer the conditional probability distribution specific to the data informing that node. Such repetitive scanning can be very CPU demanding, especially when considering a large training image or when generating a large number of realizations each with many nodes.

The algorithm implementation proposed in this paper is much less CPU demanding without being too memory (RAM) demanding. This new implementation is based on the two following properties:

Property 1. *Given a template τ_n of n data variables, the number of cpdf's associated with τ_n (pdfs conditional to data events d_n associated with τ_n) that can be actually inferred from the training image is related to the training image dimensions, hence is generally much smaller than the total number K^n of cpdfs associated with τ_n.*

Property 2. *The probability distribution conditional to a data event $d_{n'}$, associated with a subtemplate $\tau_{n'}$ of τ_n ($n' \leq n$) can be retrieved from the probability distributions conditional to the data events d_n associated with τ_n and for which $d_{n'}$ is subset.*

Let $d_{n'}$ be a data event associated with a subtemplate $\tau_{n'}$ of τ_n ($n' \leq n$). The number $c(d_{n'})$ of replicates of $d_{n'}$ is equal to the sum of the replicates of all data events d_n associated with τ_n and for which $d_{n'}$ is a subset (Strebelle, 2000):

$$c(d_{n'}) = \sum_{\substack{d_n \text{ ass. with } \tau_n \\ d_{n'} \subset d_n}} c(d_n)$$

Similarly for the number $c_k(d_{n'})$ of d_n-replicates with a central value $S(\mathbf{u})$ equal to s_k:

$$c_k(d_{n'}) = \sum_{\substack{d_n \text{ ass. with } \tau_n \\ d_{n'} \subset d_n}} c_k(d_n)$$

Knowledge of $c_k(d_{n'})$ and $c(d_{n'})$ allows then estimating the probability distribution conditional to $d_{n'}$, using relation (5).

Denote by $W(\mathbf{u})$ the data search neighborhood centered on location $\mathbf{u}$. Consider the data template τ_n constituted by the n vectors $\{\mathbf{h}_\alpha, \alpha = 1, \ldots, n\}$ defined such that the n locations $\mathbf{u} + \mathbf{h}_\alpha, \alpha = 1, \ldots, n$ correspond to the n grid nodes present within $W(\mathbf{u})$. The *snesim* algorithm proceeds in two steps:

1. First, store in a dynamic data structure, called search tree (Roberts, 1998, p. 537–592), only those cpdfs associated with τ_n that can be actually inferred from the training image. More precisely, store in a search tree the numbers of occurrences of data events and central values $(c_k(d_n))$ actually found over the training image, and from which the training proportions (5) can be calculated. Because of property 1, the amount of RAM required by the search tree is not too large, if a data template τ_n with a reasonable number n of nodes, say, less than 100 nodes, is retained. The construction of that search tree requires scanning the training image one *single* time prior to the image simulation, hence it is very fast (Strebelle, 2000).
2. Next, perform simulation by visiting each grid node one single time along a random path. At each node $\mathbf{u}$ to be simulated, the conditioning data are searched in $W(\mathbf{u})$, hence the local conditioning data event is necessarily associated with a subtemplate of τ_n. According to property 2, local cpdf can be retrieved from the search tree. The training image need not be scanned anew at each unsampled node, which renders the *snesim* algorithm much faster than Guardiano and Srivastava's original implementation.

The main steps of the *snesim* simulation algorithm are now presented in detail:

1. Scan the training image(s) to construct the search tree. Only those data events which actually occur over the training image are stored in the search tree. A maximum data search template is defined to limit the geometric extent of those data events.
2. Assign the original sample data to the closest grid nodes. Define a random path visiting once and only once all unsampled nodes.
3. At each unsampled location $\mathbf{u}$, retain the conditioning data actually present within the maximum search template used to construct the search tree. Let n' be the number of those conditioning data, and $d_{n'}$ the corresponding data event. Retrieve from the search tree the proportions of type (5) corresponding to the data event $d_{n'}$. To ensure that these proportions are significant, if the total number $c(d_{n'})$ of training $d_{n'}$-replicates is less than an input minimum value $c_{\min}$, the most distant conditioning datum is dropped, reducing the number of conditioning data to $(n' - 1)$; proportions conditioned to this lesser data event $d_{n'-1}$ are retrieved again from the search tree, and so on ... If the number of data drops to $n' = 1$, and $c(d_{n'})$ is still lower than $c_{\min}$, the conditional probability $p(\mathbf{u}; s_k \mid (n'))$ is replaced by the marginal probability p_k.

4. Draw a simulated s-value for node $\mathbf{u}$ from the cpdf read from the search tree. That simulated value is then added to the s-data to be used for conditioning the simulation at all subsequent nodes.
5. Move to next node along the random path and repeat steps 3 and 4.
6. Loop until all grid nodes are simulated. One stochastic image has been generated. Reiterate the entire process from step 2 with a different random path to generate another realization.

To ensure exact reproduction of the hard data at their locations, all original sample data are relocated to the nearest simulation grid node and their values are frozen. The small scale spatial continuity of the training image is passed to the simulated realizations through the training proportions, hence as the node being simulated gets closer to a hard datum location the conditional variance decreases as it does on the training image.

Multiple Grid Implementation

The maximum data search template retained should not be taken too small, otherwise large scale structures of the training image would not be reproduced. On the other hand, a search template including too many grid nodes would lead to storing a large number of cpdfs in the search tree, increasing CPU cost and memory demand. One solution to capture large scale structures while considering a data search template with a reasonably small number of grid nodes is provided by the multiple grid concept initially proposed by Gómez-Hernández (1991) and further developed by Tran (1994). The multiple grid approach implemented in *snesim* consists of simulating a number G of increasingly finer grids. The gth $(1 \leq g \leq G)$ grid is constituted by each (2^{g-1})th node of the final simulation grid $(g = 1)$. The data search template adopted for the various nested simulation grids need not have the same geometric configuration. The larger search templates of the coarser simulation grids allow capturing the large scale structures of the training image.

One search tree needs to be constructed per simulation grid, possibly using a different training image reflecting the heterogeneities specific to that scale. When the gth grid simulation is completed, its simulated values are frozen as data values to be used for conditioning on the next finer simulation grid.

Reproduction of the Marginal Distribution

In *snesim*, no explicit constraint ensures reproduction of the sample histogram, or any other target marginal distribution $\{p_k, k = 1, \ldots, K\}$. Simulated realizations may display global proportions significantly different from the original sample distribution, particularly if probability values are borrowed from a

training image having a histogram significantly different from the target marginal distribution.

Programs, such as the GSLIB program *trans* (Deutsch and Journel, 1998, p. 227), allow postprocessing of any training image or simulated realization to approximate any target histogram, while honoring the original hard data values at their locations and without affecting the values rank orders. However, if the target histogram is very different from the original image histogram, both variograms and mp statistics may be changed significantly by this transform.

The first recommendation is to generate training images with global proportions reasonably similar to the target proportions one wish to impose to the final model. Next a servosystem, described in detail in Strebelle (2000), has been implemented which modifies gradually the proportions (5) so that the simulation algorithm always remains close to the global target proportions $\{p_k, k = 1, \ldots, K\}$ as it progresses from one node to another.

Integration of Secondary Data

Conditioning could also include soft information, e.g. seismic data in reservoir applications. Bayes' relation (4) can be extended to evaluate the probability distribution of a variable $S(\mathbf{u})$ conditioned to both the nearest n hard data $s(\mathbf{u} + \mathbf{h}_\alpha) = s_{k_\alpha}, \alpha = 1, \ldots, n$, and the collocated soft datum $y(\mathbf{u}) = y$:

$$\text{Prob}\{A_k = 1 \mid D = 1, y\} = \frac{\text{Prob}\{A_k = 1, D = 1, y\}}{\text{Prob}\{D = 1, y\}} \tag{6}$$

Starting from a training image of the primary variable, the corresponding training image of the soft data variable can be forward simulated. Both hard and soft training images are considered as a single vectorial training image $\mathbf{T}$. Under a prior decision of stationarity (export license), this vectorial training image can be scanned to evaluate expression (6). In practice, this scanning requires reducing the y-conditioning to only a few locations of y-data values, for example the sole collocated value $y(\mathbf{u})$ as in expression (6), and a few classes of y-values, for example the four quartile classes of the y-sample histogram.

Note that D includes the previously simulated nodes; hence a hard data training image is required even if no original sample hard data are available.

Extension of the algorithm to conditioning to more than one soft y-datum is absolutely straightforward. Increasing the number n_Y of conditioning soft data, however, entails increasing the size of the extended conditioning data event d_n. If each of the n hard data variables can take K different values and each of the n_Y soft data variables can take L different values, the total number of different data events for a given data geometry is $K^n L^{n_Y}$, which increases dramatically with n_Y. The number $c(d_{n+n_Y})$ of training replicates of the extended conditioning data event

decreases dramatically with n_Y, which renders the inference of the local probability distribution more difficult. The search tree relates only to those extended data events which actually occur over the vectorial training image.

If n_Y has to be large, e.g. to allow capturing patterns or mp statistics displayed by the soft data, a different algorithm is proposed in Strebelle (2000). This algorithm amounts to combine the *snesim*-derived probability $\text{Prob}\{A_k = 1 \mid D = 1\}$ with the probability $\text{Prob}\{A_k = 1 \mid y\}$, the latter being possibly derived by neural net modeling of a pair of training images (facies vs. soft data).

SIMULATION OF A FLUVIAL RESERVOIR

The *snesim* algorithm was tested on the simulation of a horizontal 2D section of a fluvial reservoir. Fluvial reservoirs are characterized by the presence of sinuous sand-filled channels within a background of mudstone (Facies 1). For this example, we consider two types of sand: border sands (Facies 2) constituted by levies and crevasses splays which are intermediate in reservoir quality, and channel sands (Facies 3) which correspond to the best reservoir rock. A realistic modeling of the curvilinear sand channel patterns is critical for reliable connectivity assessment and flow simulation of such reservoir.

The object-based program *fluvsim* (Deutsch and Wang, 1996) was used to generate the reference "true" image of size $100 * 100 = 10000$ pixels shown in Figure 1(A). Thirty-two sample data were collected at random over the true image with facies proportions close to the true global proportions (Fig. 1(B)).

Two training images which could have been hand drawn by a geologist, then digitized, were considered.

- A first large scale training image, comprising $100 * 100 = 10000$ pixels, depicts the deemed geometry of sand channels versus their background constituted by Facies 1 and 2 pooled together (Fig. 1(C)). This image carries information about the mean width, the major direction, and the sinuosity of the channels and their grouping in space.
- A second small scale training image, comprising only $35 * 35 = 1225$ pixels, carries the prior information about the mean width of the levies and crevasses, and their locations relative to the channels (Fig. 1(D)).

Simulation proceeds in two nested sequences:

- First, Facies 3 (channel sands) was simulated versus Facies 1 and 2 pooled together, conditional to the 32 sample channel indicator data of Figure 1(B), using the large scale training image. The multiple grid option was used to capture the large scale channel continuity: four increasingly finer grids were considered.

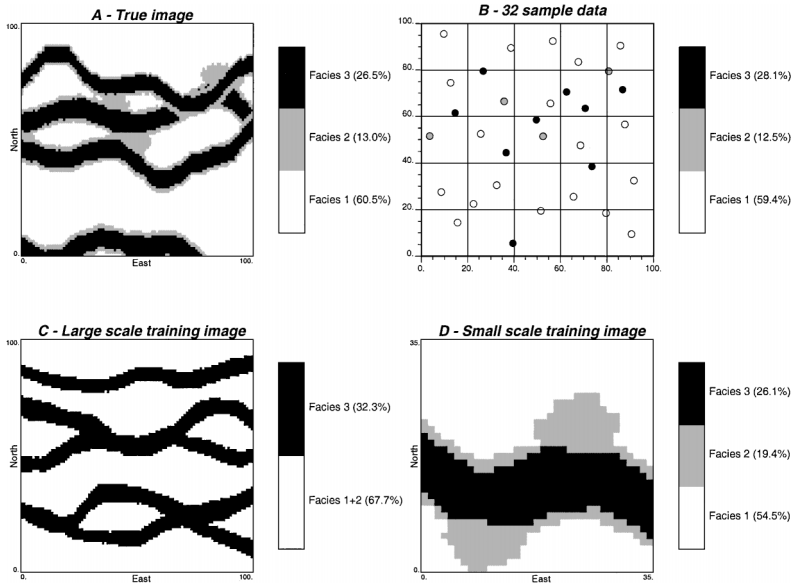

Figure 1. Simulation of a horizontal 2D section of a fluvial reservoir. (A) True image, (B) 32 sample data, (C) large scale training image, (D) small scale training image.

Ten realizations were generated using *snesim*. Their generation took 2.4 CPU second per realization with a DEC 600 MHz desktop computer under Unix. About 17 nearby data were retained in average for the simulation of any single node.

The first *snesim* realization is displayed in Figure 2(A). To provide a yardstick for comparison, Figure 2(B) shows a realization conditioned to the same 32 sample data, generated by the GSLIB sequential indicator simulation program *sisim* (Deutsch and Journel, 1998, p. 175), hence using a variogram model accounting only for two-point correlation. In contrast to *sisim*, *snesim* allows reproducing reasonably well the sinuous channel patterns displayed by the training image, although with some deficit of large scale continuity: the simulated channels do not all cross the image from one side to the other. This is a problem of ergodicity which can be alleviated by using a much larger training image, at least twice larger than the maximum distance of continuity of the channels.

The servosystem described in Strebelle (2000) allows one to reproduce reasonably well the sample channel proportion $\hat{p} = 28.1\%$: the sand proportions of the ten realizations fluctuate between 27.3 and 28.3% with a mean of 27.7%. Note that the difference between the training and the target sample channel proportions is greater than 4%. This indicates that the

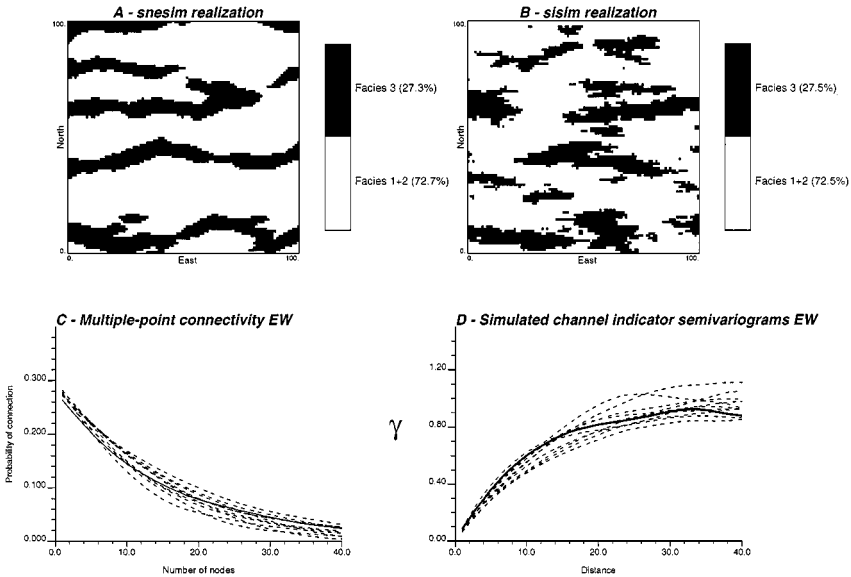

Figure 2. Simulation of Facies 3 (channel sand). (A) One *snesim* realization, (B) one *sisim* realization, (C) multiple-point connectivity along EW (continuous line refers to true image, dashed lines refer to the 10 *snesim* realizations), (D) EW channel sand indicator variograms standardized to variance 1 (continuous line refers to true image, dashed lines refer to the 10 *snesim* realizations).

algorithm can handle training images with inaccurate global proportions as long as they depict accurately the geometry of heterogeneities.

A static measure of connectivity of the sand channels can be obtained through the mp connectivity function proposed by Journel and Alabert (1989). This function is the proportion of connected strings of n channel pixels plotted versus n. Figure 2(C) displays this connectivity function calculated along the EW direction of channel orientation for the true image and the 10 *snesim* realizations. All realizations reproduce reasonably well the true connectivity of channels.

Although there is no variogram modeling nor explicit constraint for variogram reproduction in *snesim*, the simulated channel sand indicator variograms along the EW direction are close to the variogram of the reference image (Fig. 2(D)). The two-point statistics are implicitly included in the mp statistics exported to the simulated image.

- Next Facies 1 and 2 were simulated within the areas previously simulated as Facies $1 + 2$ (nonchannel sand), conditional to the original Facies 1 and 2 hard data and the previously simulated Facies 3 nodal values, using the small scale training image. When scanning that small scale training image, data events for which the central value is Facies 3 are discarded. Hence,

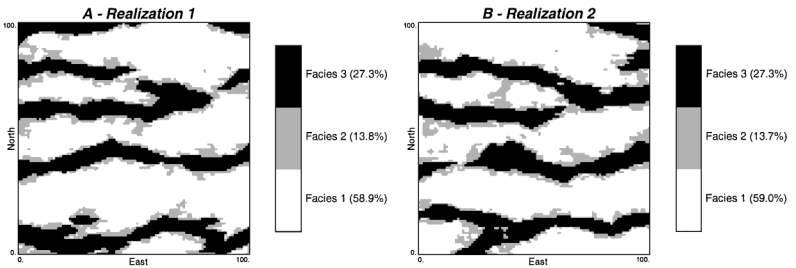

Figure 3. Simulation of Facies 1 and 2 (mudstone and border sand). Two *snesim* simulated realizations.

the conditional probability for Facies 3 is zero, which ensures that Facies 3 cannot be simulated in the Facies $1 + 2$ area.

The simulated realizations corresponding to the first two channel realizations generated in the first step are shown in Figure 3. Because levies were treated as small scale heterogeneities, their actual large scale continuity is understated. Conversely, the crevasse splays being lumped together with the levies are simulated more elongated than they are over the small scale training image. Note that the resulting simulated Facies 2 (border sand) is correctly attached to the channels.

Sensitivity to the Training Image

To analyze the sensitivity of the *snesim* simulated realizations to the training image, sand channels are now simulated using the two alternative large scale training images displayed in Figure 4(A) and (C). The first alternative training image displays much thinner channels than the original training image of Figure 1(C). The second alternative training image was simply obtained by a 90° rotation of the original training image. One *snesim* realization was generated using each alternative training image (Fig. 4(B) and (D)). In both cases, the simulated channels reproduce the shape of the corresponding training channels: they display the smaller thickness of the first alternative training image, or the NS preferential orientation of the second alternative training image.

Those results show the dependence of the simulated realizations on the training image. *snesim* anchors the mp structures borrowed from the training image to the original sample data. The "essence" of the training image, constituted by the geometrical patterns found often enough in that image and consistent with the sample data, is reflected by the simulated realizations. The training image determines the main features of the simulated images, however less, so as the sample data are more numerous. Indeed hard data are frozen at their locations; hence if they are many, the structures they reveal, especially medium to large scale structures,

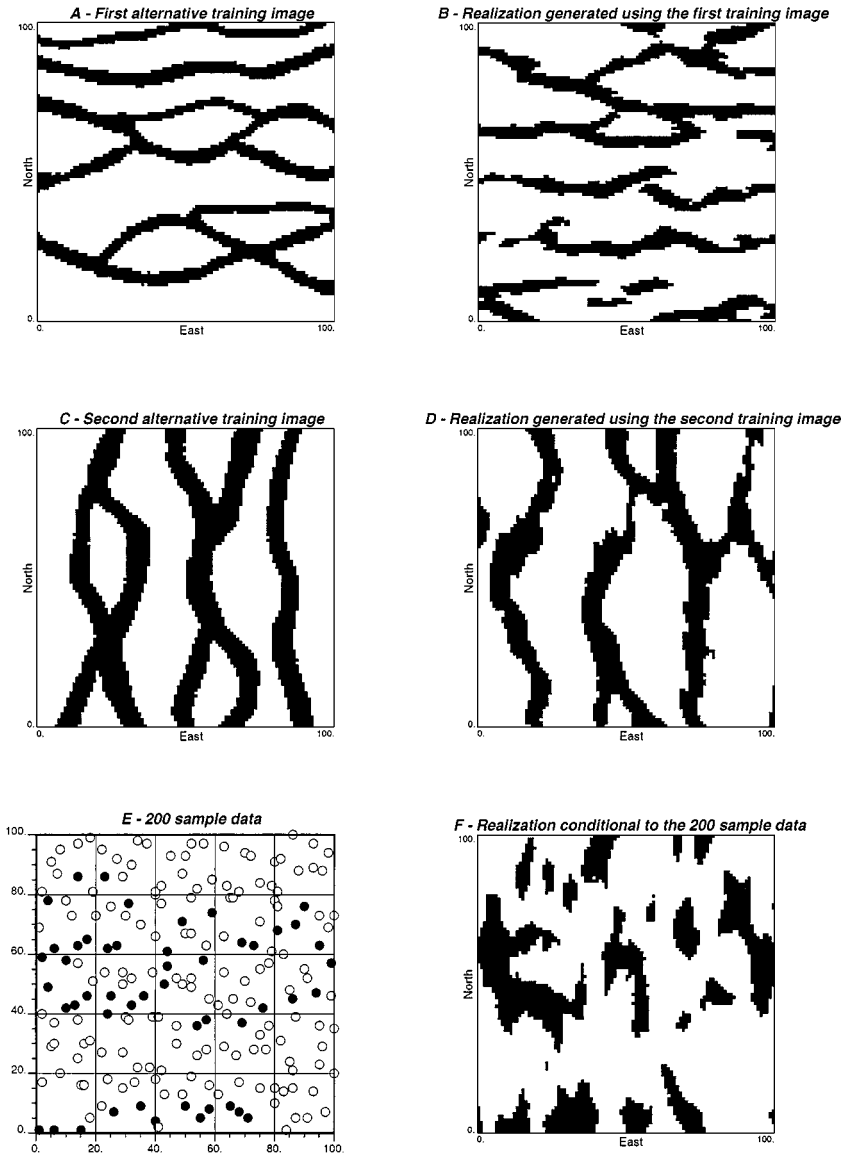

Figure 4. Sensitivity of the *snesim* simulated realizations to the training image. (A and B) First alternative large scale training image and corresponding simulated realization conditional to the 32 sample data of Figure 1B, (C and D) second alternative training image and corresponding simulated realization conditional to the 32 sample data of Figure 1B, (E) set of the 200 sample data collected at random from the reference image of Figure 1A; (F) simulated realization conditional to the 200 sample data using the second alternative training image.

prevail over the structures read from the training image and conflicts may occur resulting in poorer reproduction of certain structures of the training image.

To illustrate such possible conflict, 200 sample data were collected at random over the true image of Figure 1(A) (Fig. 4(E)), and a simulated realization conditional to those 200 sample data was generated using the second alternative training image shown in Figure 4(C). Because the 200 sample data reflect the actual EW continuity, the large scale NS-oriented training channel patterns are reproduced poorly (Fig. 4(F)). Such poor reproduction of the structures of the training image is indicative of either unreliable hard data or an inappropriate training image.

The sensitivity of the *snesim* simulated realizations to the size and the geometric configuration of the data template used to construct the search trees is analyzed in Strebelle (2000): the *snesim* algorithm appears robust with respect to the data template retained, as long as it is not too small.

SIMULATION OF MULTIPLE COMPLEX PATTERNS

In contrast to object-based techniques, the algorithm proposed in this paper is general: *snesim* allows simulating any type of structures, of any shape, at any scale, as long as the training image has a repetitive character. A new program need not be written to fit any particular parametric object type or geometry. To illustrate this point, a second case study is proposed.

The reference "true" image and the training image, both of size $500 * 500 = 250{,}000$ pixels, display a mixture of three geometrically different patterns (Fig. 5(A) and (C)):

- ellipses elongated in the EW direction (Facies 1);
- crescents in the diagonal direction (Facies 2);
- small crosses (Facies 3).

Background is Facies 4. Crosses do not overlap each other, nor any ellipse or crescent. Crescents overlap ellipses.

A large number of sample data (400 or 0.16% of the true image) were collected at random over the true image, with facies proportions close to the true global proportions (Fig. 5(B)). Figure 5(D) displays the corresponding training image for the ellipses (Facies 1) only. Simulation proceeds in three nested sequences:

- First, Facies 1 (ellipses) is simulated versus all other facies pooled together, and conditional to the 400 Facies 1 sample indicator data of Figure 5(B). The conditional probability distributions are inferred from the ellipse training image shown in Figure 5(D), itself reconstituted from the full training image of Figure 5(C). Anisotropy of the ellipses is accounted for by considering an anisotropic search template elongated along EW to construct the search tree.

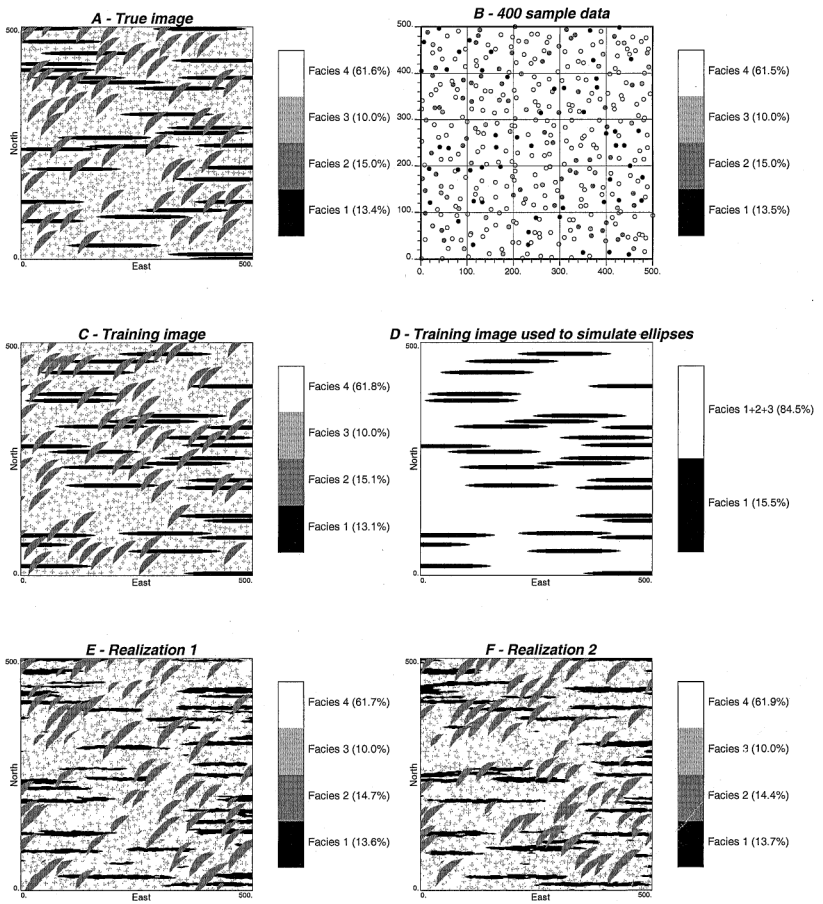

Figure 5. Simulation of multiple patterns. (A) True image, (B) location of the 400 sample data, (C) training image, (D) reconstructed training image used to simulate Facies 1, (E and F) two simulated realizations.

- Next, Facies 2 (crescents) is simulated versus all other facies pooled together. This simulation is independent of that of Facies 1, and is conditional to the 400 Facies 2 sample indicator data of Figure 5(B). The conditional probability distributions are inferred from the training image of Figure 5(C), Facies 1, 3, and 4 being pooled together. To account for the anisotropy of the crescents, an anisotropic search template elongated along N45E was used to construct the search tree.

  Because crescents overlap the ellipses in the training image of Figure 5(C), the crescent realizations were superimposed on top of the

ellipse realizations generated in the first step, resulting in a realization displaying ellipses and crescents versus Facies 3 and 4 background.

- Finally, since Facies 3 (crosses) does not overlap ellipses or crescents, it is simulated versus Facies 4 only within areas previously simulated as Nonfacies 1 and 2, and conditional to the 400 Facies 3 sample indicator data of Figure 5(B). Inference of the conditional probability distributions is limited to the Facies 3 and 4 areas of the training image displayed in Figure 5(C). An isotropic conditioning search template was retained to construct the corresponding search tree.

The previous sequence of simulations was used to generate the two realizations of Figure 5(E) and (F). The three types of structures are reasonably reproduced with facies proportions close to the target sample proportions of Figure 5(B). Although simulation proceeds in three nested sequences, the same general algorithm, namely *snesim*, has been used to simulate the three types of patterns displayed by the training image of Figure 5(C).

Recall that the *snesim* simulation algorithm is exact, in that it honors all hard data values at their exact locations, no matter the density or configuration of these data. In presence of the dense data shown in Figure 5(B), this exactitude condition would be difficult to fulfill with object-based algorithms.

PRELIMINARY CONCLUSIONS

A new algorithm (*snesim*) is proposed for sequential simulation of categorical random fields. This algorithm utilizes local proportions read from training images. In contrast to classical algorithms based only on the two-point variogram model, *snesim* allows reproduction of complex multiple-point patterns, such as undulating channels. Such multiple-point patterns include the traditional 2-point statistics which are thus also reproduced even though they are not modeled explicitly. The algorithm requires large amount of RAM but not exceeding what is presently (1999) available on desktop computers, and is fast enough to consider 3D simulation grids.

In contrast to object-based techniques, the proposed approach is general: the same algorithm allows simulation of any type of geological heterogeneity, of any shape, at any scale. *snesim* does not need to be modified to fit any particular parametric object type or geometry. The reason for such generality is that the user need not decide in advance which mp statistics or geometric parameters are essential and should be reproduced in the simulated realizations. All statistics are directly provided by the training images which, therefore, should display the heterogeneities deemed relevant to the phenomenon being modeled. The classical steps of variogram modeling and kriging or object parametrization are completely shortcut.

The simplicity of the *snesim* algorithm comes, however, from a greater reliance on the prior decision of stationarity: much more than a mere variogram is

borrowed from the training image(s). As of now, the *snesim* algorithm borrows all structures present in the training images without discrimination at the risk of exporting details irrelevant to the actual phenomenon being modeled. The extent of the export license (stationary decision) raises two important questions: which mp statistics should be borrowed from the training image, and how can we limit the patterns being borrowed to those specific statistics?

The first question calls for a better understanding of the "essence" of a training image: could some objective criteria, such as impact on flow response in reservoir applications, be proposed to identify those "essential" mp statistics to be exported? The on-going work of Caers, Srinivasan, and Journel (1999) explores this avenue. Or should we rely on a more subjective (visual) identification of those patterns to be reproduced?

The second question could be addressed by a proper selection of the conditioning data events. Restricting the mp statistics which can be borrowed from the training image entails constraints on the geometry of the conditioning data events. The present approach, which consists of dropping the furthest away datum whenever the probability distribution cannot be inferred reliably from the training image, does not allow such control on the geometry of the data events.

REFERENCES

Bridge, J. S., and Leeder, M. R., 1979, A simulation model of alluvial stratigraphy: Sedimentology, v. 26, p. 617–644.

Caers, J., and Journel, A. G., 1998, Geostatistical quantification of geological information for a fluvial-type North-Sea reservoir: SPE paper no. 56655.

Caers, J., Srinivasan, S., and Journel, A. G., 1999, Stochastic reservoir simulation using neural networks trained on outcrop data: SPE paper no. 49026.

Deutsch, C. V., 1992, Annealing techniques applied to reservoir modeling and the integration of geological and engineering (well test) data: Unpublished doctoral dissertation, Stanford University, 306 p.

Deutsch, C. V., and Journel, A. G., 1998, GSLIB: Geostatistical software library and user's guide, 2nd ed.: Oxford University Press, New York, 368 p.

Deutsch, C. V., and Wang, L., 1996, Hierarchical object-based stochastic modeling of fluvial reservoirs: Math. Geol., v. 28, no. 7, p. 857–880.

Farmer, C. L., 1988, The generation of stochastic fields of reservoir parameters with specified geostatistical distributions, *in* Edwards, S., and King, P. R., eds., Mathematics in oil production: Clarendon Press, Oxford, p. 235–252.

Gómez-Hernández, J. J., 1991, A stochastic approach to the simulation of block conductivity fields conditioned upon data measured at a smaller scale: Unpublished doctoral dissertation, Stanford University, 351 p.

Goovaerts, P., 1997, Geostatistics for natural resources evaluation: Oxford University Press, New York, 483 p.

Guardiano, F., and Srivastava, R. M., 1993, Multivariate geostatistics: Beyond bivariate moments, *in* Soares, A., ed., Geostatistics-Troia, Vol. 1: Kluwer Academic, Dordrecht, p. 133–144.

Haldorsen, H. H., and Damsleth, E., 1990, Stochastic modeling: J. Pet. Technol., v. 42, April, p. 404–412.

Journel, A. G., 1993, Geostatistics: Roadblocks and challenges, *in* Soares, A., ed., Geostatistics-Troia, Vol. 1: Kluwer Academic, Dordrecht, p. 213–224.

Journel, A. G., 1997, Deterministic geostatistics: A new visit, *in* Baffi, E., and Shofield, N., eds., Geostatistics Wollongong, Vol. 1: Kluwer Academic, Dordrecht, p. 174–187.

Journel, A. G., and Alabert, F. G., 1989, Non-Gaussian data expansion in the Earth Sciences: Terra Nova, v. 1, p. 123–134.

Omre, H., 1991, Stochastic models for reservoir characterization, *in* Kleppe, J., and Skjaeveland, S. M., eds., Recent advances in improved oil recovery methods for North Sea sandstone reservoirs: Norwegian Petroleum Directorate, Stavanger, 14 p.

Roberts, E. S., 1998, Programming abstractions in C: A second course in computer science: Addison-Wesley, Reading, MA, 819 p.

Srivastava, M., 1992, Iterative methods for spatial simulation: Stanford Center for Reservoir Forecasting, Rep. No. 5, 24 p.

Srivastava, M., 1995, An overview of stochastic methods for reservoir characterization, *in* Yarus, J., and Chambers, R., eds., Stochastic modeling and geostatistics: Principles, methods, and case studies, Vol. 3: AAPG Computer Applications in Geology, Tulsa, p. 3–16.

Strebelle, S., 2000, Sequential simulation drawing structures from training images: Unpublished doctoral dissertation, Stanford University, 200 p.

Tjelmeland, H., 1996, Stochastic models in reservoir characterization and Markov random fields for compact objects: Unpublished doctoral dissertation, Norwegian University of Science and Technology.

Tran, T. T., 1994, Improving variogram reproduction on dense simulation grids: Comput. Geosci., v. 20, no. 7, p. 1161–1168.

Xu, W., 1996, Conditional curvilinear stochastic simulation using pixel-based algorithms: Math. Geol., v. 28, no. 7, p. 937–949.