

CONDITIONAL SIMULATION OF SPATIAL STOCHASTIC MODELS

C. LANTUÉJOUL

*Ecole des Mines, Centre de Géostatistique
35 rue Saint-Honoré, 77305 Fontainebleau, France*

Abstract How to produce realizations of a spatial stochastic model when they are subject to a set of conditions or constraints? These realizations are called conditional simulations. This paper presents some statistical tools to carry out conditional simulation of spatial stochastic models. This includes regression techniques for the conditional simulation of gaussian random functions, Gibbs sampler for truncated gaussian or plurigaussian random functions, and Metropolis-Hasting algorithm and restriction of a Markov chain for the boolean model.

Keywords: regression, Gibbs sampler, Metropolis-Hastings algorithm, restriction of a Markov chain, Gaussian random function, plurigaussian random functions, boolean model.

1. Introduction

Let $z = (z(x), x \in \mathbb{R}^d)$ be a numerical function. Suppose that the values of z are known at some data locations: $z(x_i) = z_i$ for $i = 1, \dots, n$. Let also v be a bounded domain of \mathbb{R}^d and λ be a numerical value. The problem is to assess numerical variables like

$$\frac{1}{|v|} \int_v z(x) dx \quad \frac{1}{|v|} \int_v 1_{z(x) \geq \lambda} dx \quad \frac{1}{|v|} \int_v z(x) dx \geq \lambda$$

starting from the available information. To address this problem, the classical statistical approach consists of interpreting z as a realization of a random function, say Z , and of building predictors of the random variables

$$\frac{1}{|v|} \int_v Z(x) dx \quad \frac{1}{|v|} \int_v 1_{Z(x) \geq \lambda} dx \quad \frac{1}{|v|} \int_v Z(x) dx \geq \lambda$$

At first, one could think that it suffices to predict the random value $Z^*(x)$ at each location $x \in v$ (for instance by linear regression), and then derive the

predictors

$$\frac{1}{|v|} \int_V Z^*(x) dx \quad \frac{1}{|v|} \int_v 1_{Z^*(x) \geq \lambda} dx \quad \frac{1}{|v|} \int_v Z^*(x) dx \geq \lambda$$

But except for the first numerical variable that is a linear functional of the $z(x)$'s, the predictors obtained do not have good properties. In particular they are biased, and even severely biased [9].

This is the reason why we prefer resorting to a Monte Carlo approach. This consists of generating a family of realizations (or simulations) of Z . For each simulation the variable of interest is assessed. This gives a family of values that can then be combined to produce not only a predictor but also confidence limits.

As a matter of fact, the Monte Carlo approach is quite standard. The only real novelty here is that this approach is not based on all simulations of Z , but only on those that honor the data. They are called conditional simulations. This paper deals with the design of algorithms for conditional simulations. It addresses the following general question: what are the models (of random functions, random sets, point processes...) that we know how to simulate when they are subject to a set of conditions (honoring data points) or constraints (regional averages, connectivity for random sets...)? Although it is impossible to give a comprehensive answer to that question, it is nonetheless possible to give algorithms for some prototype models, as well as to bring out the statistical tools used for their design. Due to the limited number of pages, many details are omitted. For a more systematic presentation, the reader can consult either [10] or chapter 7 of [2].

2. Gaussian random function

Recall that a random function $Z = (Z(x), x \in \mathbb{R}^d)$ is said to be gaussian if every linear combination of its variables follows a gaussian distribution. In the stationary case, it is well known that the statistical properties of Z are specified by its mean value m and its covariance function C . How can we produce realizations of Z subject to the conditions $Z(x_i) = z_i, i = 1, \dots, n$? The procedure adopted here is usually attributed to Journé [8]. Let $Z^r(x)$ be the regression of $Z(x)$ on the data, namely

$$Z^r(x) = \sum_{i=1}^n \lambda_i(x) Z(x_i) + m \left(1 - \sum_{i=1}^n \lambda_i \right)$$

where the coefficients $\lambda_i(x)$ are the solution of the linear system of equations

$$\sum_{j=1}^n \lambda_j(x) C(x_j - x_i) = C(x - x_i) \quad 1 \leq i \leq n$$

It can be easily shown that the regression Z^r and the residual $Z - Z^r$ are independent gaussian random functions. Thus Z can be written as the sum

of the regression and an independent gaussian residual. This leads to the conditional simulation algorithm that consists of simulating the residual $Z - Z^r$ and adding it to the regression. Explicitly

$$Z^{cs}(x) = Z^r(x) + Z^s(x) - Z^{sr}(x) = Z^s(x) + \sum_{i=1}^n \lambda_i(x)[Z(x_i) - Z^s(x_i)]$$

where Z^s denotes a non conditional simulation of Z . Various algorithms exist to produce it (circulant embedding, Cholesky, FFT, turning bands...), see [2, 3, 10].

To illustrate this algorithm, we have considered a standardized gaussian random function with a spherical covariance function (scale factor 40). Top left of figure 1 depicts a non conditional simulation in a field of 400×400 . From this simulation, 100 points have been uniformly selected to act as conditioning data points (top right). Based on this information, the linear regression (bottom left) and a conditional simulation (bottom right) have been computed. Their textural difference is striking. Whereas the conditional simulation has the same texture as the non conditional one, the regression is smooth. Of course, this textural difference vanishes when the number of conditioning data points is very large.

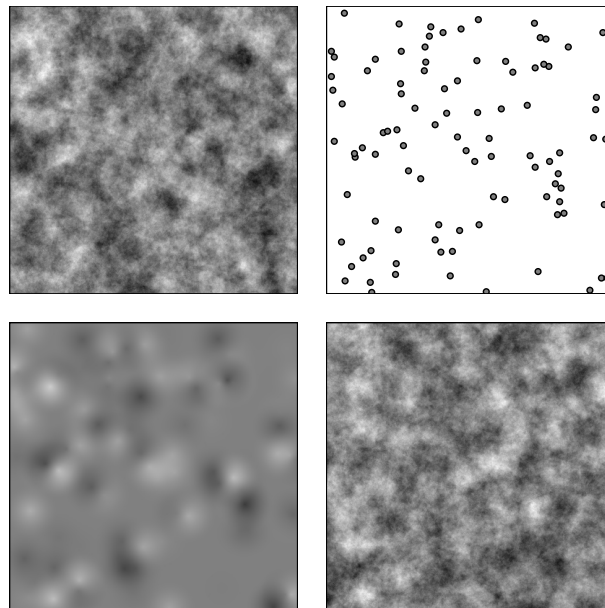


Figure 1. Conditional simulation of a gaussian random function. Top left, a non conditional simulation. Top right, a set of 100 conditional data points. Bottom left, the linear regression (simple kriging) on the data. Bottom right, a conditional simulation.

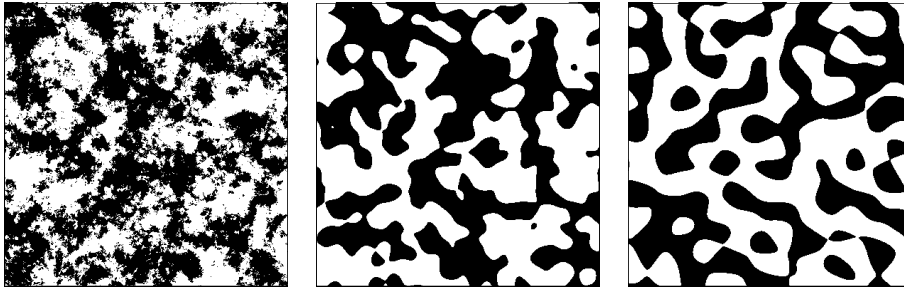


Figure 2. These realizations come from 3 different isotropic gaussian excursion sets. From left to right, their covariance function are exponential, gaussian and cardinal sine. In all cases, the excursion level is 0.

3. Excursion set of a gaussian random function

Let Z be a standardized gaussian random function, and let λ be a real value. The excursion set of Z above level λ is defined as the random set

$$X = \{x \in \mathbb{R}^d : Z(x) \geq \lambda\}$$

This random set model depends on two parameters. These are the level λ that determines the proportion occupied by X , and the covariance function C of Z that specifies the texture properties of X [2, 10, 12]. Figure 2 shows 3 realizations of excursion sets built using the same level 0 but 3 different covariance functions.

How to produce realizations of X given that a finite subset A is supposed to belong to X and another finite subset B to the complement X^c of X ? The procedure is in three steps. The first one consists of generating conditional gaussian values to all points of $A \cup B$. The values generated are then used as conditioning values to carry out a conditional simulation of the underlying gaussian random function. Finally, the gaussian simulation thus obtained is thresholded at level λ .

The third step is straightforward and the second one has been presented in the previous section. Thus only the first step remains to be investigated. It rests on a famous iterative algorithm called the Gibbs sampler [6]. Initially, each point of $A \cup B$ is assigned a gaussian value larger or smaller than λ depending whether it belongs to X or not. At the current stage, a point x is selected from $A \cup B$ (sequentially or uniformly). This point is assigned a new value generated from a gaussian distribution – its mean is the simple kriging of x using the values of all other points of $A \cup B$, and its variance is the kriging variance – and restricted to $[\lambda, +\infty[$ or $]-\infty, \lambda[$ depending on its belonging to X or X^c . The rate of convergence of this iterative algorithm is still the subject of ongoing research ([13, 4]). Figure 3 shows two conditional simulations obtained by updating all gaussian values 1000 times.

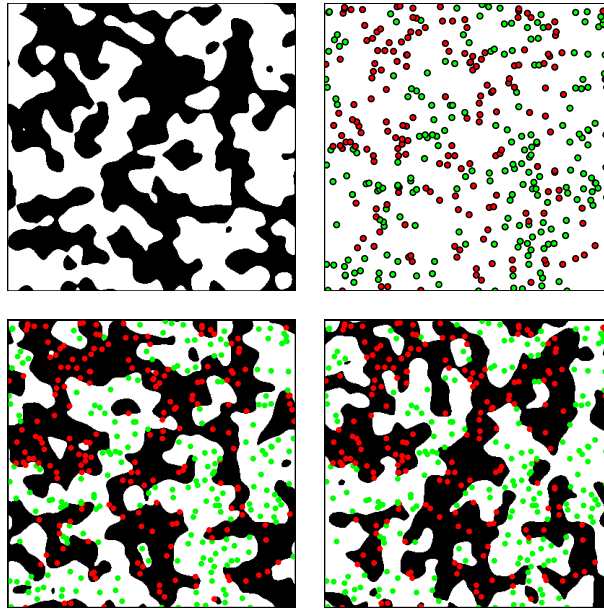


Figure 3. Conditional simulation of an excursion set of a gaussian random function. Top left, a non conditional simulation. Top right, the conditioning data points. Bottom, two conditional simulations with the conditioning data points.

4. Plurigaussian random functions

Plurigaussian random functions ([5, 1, 10] and references therein) are wide generalizations of the gaussian excursion random sets. They have been proposed to accommodate more than two phases and to handle their spatial dependence relationships. The basic ingredients required for their construction are two independent standardized gaussian random functions Y and Z defined on \mathbb{R}^d , and a family $(D_i, i \in I)$ of subsets that partitions \mathbb{R}^2 . A point $x \in \mathbb{R}^d$ is assigned the phase $i \in I$ if and only if $(Y(x), Z(x)) \in D_i$. Figure 4 shows realizations of 6 plurigaussian random functions. They have been obtained starting from the same underlying gaussian random functions but using different partitions (represented as flags below each realization). In order to understand the way the partitions have been ordered, note that the domain boundaries have a triple point that goes from $+\infty$ to $-\infty$ when considering the simulations from top to bottom and left to right. The top left simulation does not depend on Z . Indeed the yellow phase, as well as the union of the red and yellow phases, are excursion sets of Z at different levels. On the top right simulation, the 3 phases have exactly the same statistical properties. Bottom left gives an example of hierarchical model. An excursion set on Y defines the red phase. In the residual space, a second excursion set on Z specifies the green and the yellow phases. Finally, the bottom right simulation, as the top left one, does not depend on Y . The red domain is now made up of two parallel stripes limiting the green

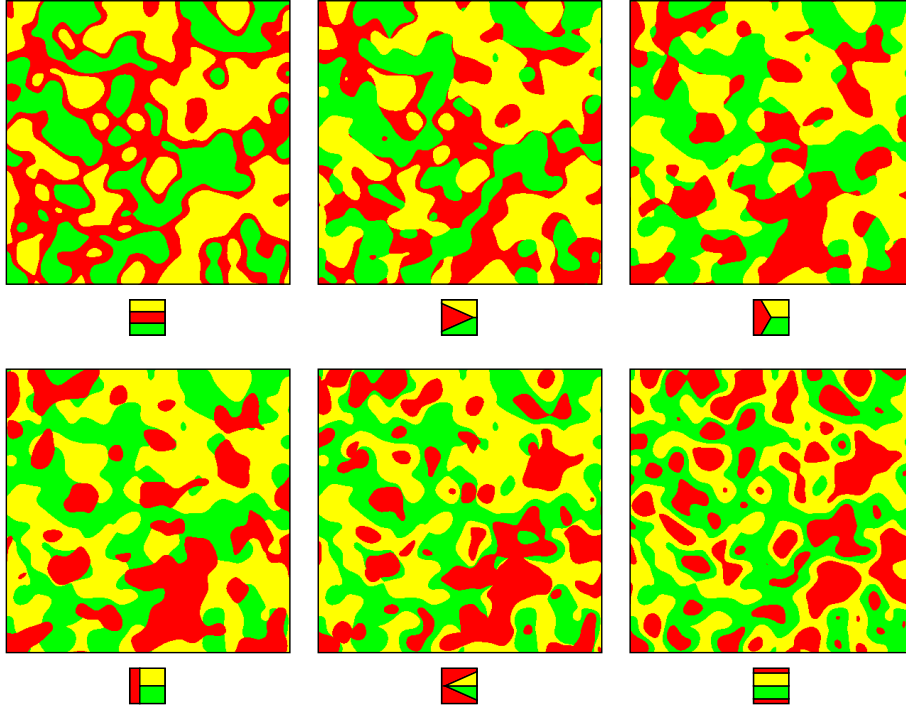


Figure 4. Realizations of 6 plurigaussian random functions. They have been built starting from the same gaussian random functions but using different partitions (the "flags").

and the yellow domains. This produces red components surrounded either by the green phase or by the yellow one.

The algorithm for conditionally simulating a plurigaussian random functions is not significantly different from that of an excursion set. It is just a multivariate extension of it. At the first step, the Gibbs sampler is used to generate the values of both underlying gaussian random functions at each data location. The second step consists of the conditional simulation of both gaussian random functions. Finally, the third step assigns a domain to each point of the simulation field. Examples of conditional simulation of the hierarchical model are given in figure 5.

5. Boolean model

This very popular model has been extensively studied by many authors ([7, 11]). Intuitively speaking, this is a union of random objects located at random. More formally, the objects come from a population $(A(x), x \in \mathbb{R}^d)$ of independent random compact subsets. They are located according to a Poisson point process \mathcal{P} with intensity $\theta = (\theta(x), x \in \mathbb{R}^d)$. The boolean model can thus be written

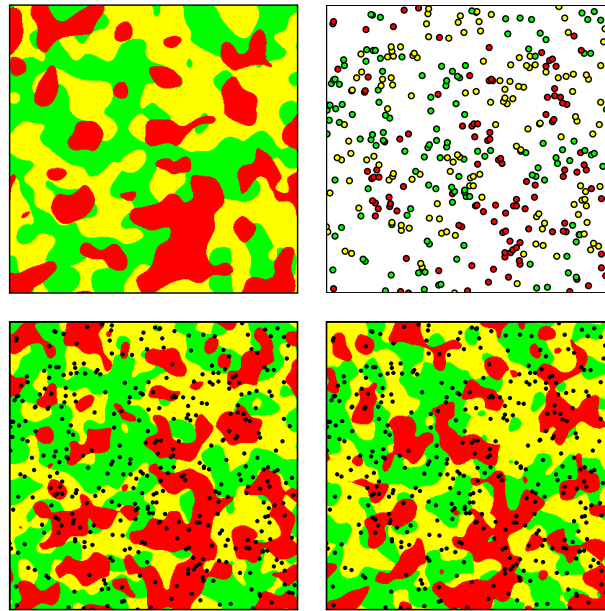


Figure 5. Conditional simulation of a plurigaussian random function (hierarchical model). Top left, a non conditional simulation. Top right, the conditioning data points. Bottom, two conditional simulations.

as

$$X = \bigcup_{x \in \mathcal{P}} A(x)$$

The problem of the simulation of a boolean model in a domain D subject to the conditions that two finite subsets C_0 and C_1 must be contained in the background (X^c) and in the foreground (X) respectively, is now addressed. It is well known that the number of objects hitting the simulation field D follows a Poisson distribution with mean

$$\vartheta = \int_{\mathbb{R}^d} \theta(x) P\{A(x) \cap D \neq \emptyset\} dx$$

There are many ways to simulate a Poisson distribution. Here we simulate it as the limit distribution of a Markov chain with transition kernel

$$P(n, n + 1) = \frac{\vartheta}{\vartheta + n + 1} \quad P(n, n) = \frac{\vartheta}{(\vartheta + n)(\vartheta + n + 1)} \quad P(n, n - 1) = \frac{n}{\vartheta + n}$$

(Metropolis-Hastings algorithm). This gives at once an iterative algorithm for simulating the boolean model in D . Suppose that the set generated at the current stage is the union of a population of n objects. At the next step, either a new object is generated and added to the population (probability $P(n, n + 1)$), or an object is uniformly selected and removed from the population (probability $P(n, n - 1)$), or even there is no change at all (probability $P(n, n)$). Now this

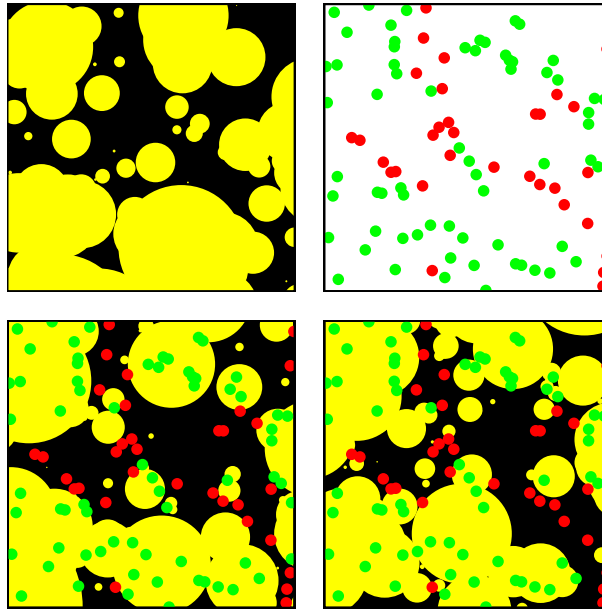


Figure 6. Conditional simulation of a boolean model. The objects are disks with exponential radii. Top left, a non conditional simulation. Top right, the conditioning data points. Bottom, two conditional simulations.

iterative algorithm can be made conditional by prescribing all conditions to be satisfied at each step. This means that a newly generated object is added to the population only if it does not conflict with the background conditions. Similarly, a newly selected object is removed from the population only if its disappearance does not conflict with the foreground conditions. Note also that a population honoring all conditions must be generated to start running the algorithm. For this, a specific procedure is required [10].

The rate of convergence of this iterative algorithm is given by the second largest eigenvalue of the transition kernel that governs the evolution of the number of objects during the conditional simulation. This eigenvalue can be experimentally assessed using an integral range technique [10]. As an illustration, both conditional simulations of figure 6 have been stopped after 2000 iterations.

References

- [1] M. Armstrong, A. Galli, G. Le Loc'h, F. Geffroy and R. Eschard. *Plurigaussian simulations*. Kluwer, Dordrecht, to appear.
- [2] J. P. Chilès and P. Delfiner. *Geostatistics: modeling spatial uncertainty*. Wiley, New York, 1999.
- [3] C. R. Dietrich and G. N. Newsam. A fast and exact method for multidimensional gaussian stochastic simulation. *Water Resour. Res.*, 31:147-156, 1993.

- [4] A. Galli and H. Gao. Rate of convergence of the Gibbs sampler in the gaussian case. *Math. Geol.*, 33(6):653-677, 2001.
- [5] A. Galli, H. Beucher, G. Le Loc'h and B. Doligez. The pros and cons of the truncated gaussian method. In M. Armstrong and P.A. Dowd, editors, *Geostatistical simulations*, pages 217-233, Kluwer, Dordrecht, 1994.
- [6] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *I.E.E.E. Trans. Pattern Anal. and Mac. Int.*, 6:721-741, 1984.
- [7] P. Hall. *Introduction to the theory of coverage processes*. Wiley, New York, 1988.
- [8] A. J. Journel and C. J. Huijbregts. *Mining geostatistics*. Academic Press, London, 1978.
- [9] C. Lajaunie. L'estimation géostatistique non linéaire. Internal report C-152, Ecole des Mines, Centre de géostatistique, 1993.
- [10] C. Lantuéjoul. *Geostatistical simulation: models and algorithms*. Springer-Verlag, Berlin, 2002.
- [11] G. Matheron. *Random sets and integral geometry*. Wiley, New York, 1975.
- [12] D. J. Nott and R. J. Wilson. Size distributions for excursion sets. In D. Jeulin, editor, *Advances in theory and applications of random sets*, pages 175-196. World Scientific, Singapore, 1997.
- [13] J. Rosenthal. Rates of convergence for Gibbs sampling for various component models. Technical report No 9322, University of Toronto, Department of Statistics, 1993.