

Simulation of a Gaussian random vector: A propagative version of the Gibbs sampler

Christian Lantuéjoul and Nicolas Desassis

Abstract Starting from the Gibbs sampler, an iterative algorithm is designed for simulating a gaussian random vector, that requires neither the inversion nor the factorization of a covariance matrix, without resting on a markovian assumption. A brief survey is given of the various ways to implement it. An example illustrates its feasibility, and a theoretical result is stated about its rate of convergence.

1 On the Gibbs sampler and its implementation

The motivation for the present work is the conditional simulation of a number of stochastic models that involve gaussian random fields, such as certain point processes (Cox-lognormal), random sets (gaussian excursion sets) or random fields (substitution random fields), to name but a few. The underlying idea is to have an iterative algorithm for simulating the gaussian field. At each step a candidate field is provided that is subsequently accepted or rejected depending on the conditions to honour.

Among all possible algorithms, the most classical one is certainly the Gibbs sampler developed by Geman and Geman [4]. From a geostatistical standpoint, it works as follows. Let $Y = (Y_a, a \in A_o)$ be a standardized (zero mean, unit variance) gaussian random vector with covariance matrix C . Let also y^c be the current step of the simulation. At the next step, a component a is selected at random and the current value y_a^c is replaced by a new value y_a^n that is drawn from a gaussian distribution, whose mean $y_a(A_o) = \sum_{b \neq a} \lambda_b^a y_b^c$ is the simple kriged estimate of the a^{th} component on the other components of A_o , and variance σ_a^2 is the corresponding kriging variance. All other components remained unchanged, i.e. $y_b^n = y_b^c$ for $b \neq a$, or $y_{A_g}^n = y_{A_g}^c$

Christian Lantuéjoul and Nicolas Desassis

MInesParisTech, Centre de Géosciences, Equipe Géostatistique, 35 rue Saint-Honoré, 77300 Fontainebleau, France, e-mail: christian.lantuejoul@mines-paristech.fr, nicolas.desassis@mines-paristech.fr

Ninth International Geostatistics Congress, Oslo, Norway, June 11. – 15., 2012

for short. Here is the corresponding algorithm. It has been written with some redundancy to facilitate its comparison with the one of the next section.

- (i) generate independently $y_a^c \sim \mathcal{G}$ for each $a \in A_o$;
- (ii) generate $a \sim \mathcal{U}(A_o)$;
- (iii) generate $y_a^n \sim \mathcal{G}(y_a(A_o^a), \sigma_a^2)$, and put $y_{A_o^a}^n = y_a^c$;
- (iv) put $y_{A_o}^c = y_{A_o}^n$ and goto (ii).

An interesting aspect of this approach is that all (simple) kriging weights as well as all kriging variances can be derived from the entries of the inverse C^{-1} of the covariance matrix:

$$\lambda_b^a = -\frac{C_{ab}^{-1}}{C_{aa}^{-1}} \quad \sigma_a^2 = \frac{1}{C_{aa}^{-1}} \quad (1)$$

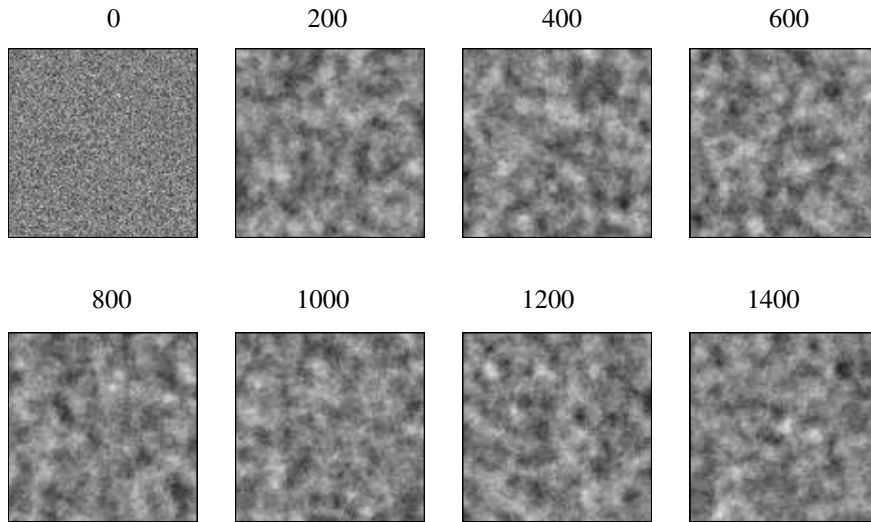


Fig. 1 Simulation of a gaussian random field by an approximate Gibbs sampler (circular neighbourhoods of radius 15). This figure shows how the simulation evolves as the number of scans (number of updating of each pixel) increases

Unfortunately, when C is of large dimension the computation of C^{-1} is simply not possible and the Gibbs sampler cannot be rigorously implemented. In such a case, a natural approach is to restrain the number of components for kriging Y_a . It turns out that this approximation is sometimes, but not always, workable. Consider for instance the simulation of a standardized gaussian random field with a spherical variogram of range 10 on a 100×100 grid (unit mesh size). The results look perfectly satisfactory when all kriginings are performed using circular neighbourhoods of radius 15 (cf figure 1) and the statistical fluctuations obtained coincide with those that we should expect owing to the variogram model and the size of the simulation field.

As this procedure is rather long time consuming, another experiment was conducted by reducing the radius of the kriging neighbourhoods (5 instead of 15). At the outset both simulations look similar but the situation abruptly changes after 200 scans. Then the texture of the spherical model fades out, its range lengthens and the contrasts become more pronounced (cf. figure 2). A more explicit description of what happens appears on figure 3. It turns out that the maximum and the minimum of the simulation are respectively monotonically increasing and decreasing to reach values well beyond the capabilities of the model. In other words, the algorithm diverges. One cannot exclude the conjecture that this problem has to be encountered whenever the neighbourhood radius has been restrained, even though it may go unnoticed when the number of scans performed is too small. More likely, there exists a critical neighbourhood radius below which it diverges and beyond which it converges. In which case, this would be a typical example of a phase transition that is well known among the community of physicists [1].

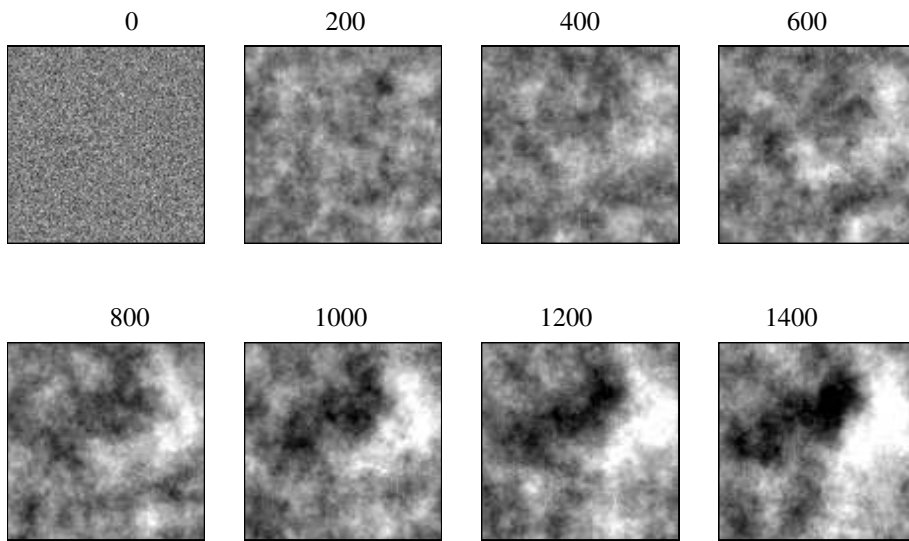


Fig. 2 Simulation of a gaussian random field by an approximate Gibbs sampler (circular neighbourhoods of radius 5). This figure shows how the simulation evolves as the number of scans increases

In what follows, a variation of the Gibbs sampler is proposed to simulate a gaussian random vector without inverting its covariance matrix. Thus, the difficulties associated with limited neighbourhoods are bypassed.

2 From the Gibbs sampler to a Gibbs propagation algorithm

Let again Y be a standardized gaussian random vector with covariance matrix C . Exactly as did Galli and Gao in [3], it is convenient to introduce the random vector $X = C^{-1}Y$. X is also a gaussian random vector and its covariance matrix is C^{-1} .

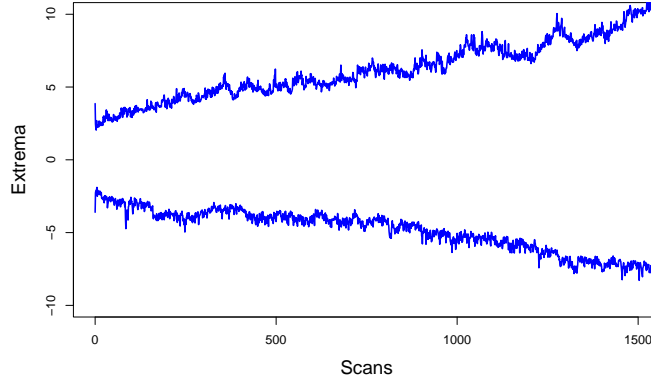


Fig. 3 The values taken by the extrema of the simulation show that the algorithm diverges

Combining this with the trivial remark that the inverse of C^{-1} is precisely C and using equation (1), it appears that the kriging of X_a on the other components of X takes the tractable expression $x_a(A_o^a) = -\sum_{b \neq a} C_{ab} x_b$. On the other hand, the associated kriging variance is $1/C_{aa} = 1$. Thus the Gibbs sampler can be applied to X free of any approximation.

This provides an algorithm for simulating Y . At each iteration, a step of the Gibbs sampler is first performed on X , then Y is updated accordingly. To be specific, let a be the component of X that is to be modified by the Gibbs sampler (hereunder referred to as a *pivot*). Then the new values of X are related to the current ones by the formulae $x_a^n = -\sum_{b \neq a} C_{ab} x_b^c + u$ with $u \sim \mathcal{G}$ and $x_b^n = x_b^c$ for $b \neq a$. Noting that x_a^n can also be rewritten as $-y_a^c + x_a^c + u$, it follows that the new values of Y satisfy

$$y_b^n = \sum_c C_{cb} x_c^n = \sum_{c \neq a} C_{cb} x_c^c + C_{ab} (-y_a^c + x_a^c + u) = y_b^c + C_{ab} (-y_a^c + u)$$

This formula is valid for each component b . In particular, for $b = a$, we obtain $y_a^n = y_a^c + (-y_a^c + u) = u$. Finally, the current and the new values of Y are related by the formula

$$y_b^n = y_b^c + C_{ab} (y_a^n - y_a^c) \quad (2)$$

In other words, at each step a gaussian value is assigned to a random pivot, then propagated to the other components. Here is the corresponding algorithm:

- (i) set $y_{A_o}^c = 0$;
- (ii) generate $a \sim \mathcal{U}(A_o)$;
- (iii) generate $y_a^n \sim \mathcal{G}$, and put $y_b^n = y_b^c + C_{ab} (y_a^n - y_a^c)$ for each $b \neq a$;

(iv) put $y_{A_o}^c = y_{A_o}^n$ and goto (ii).

Note that the initialization step has been modified. Indeed, it has been experimentally found that the convergence of this propagation algorithm is faster when y is initialized with zeroes than with independent gaussian values.

As an illustration, consider the simulation of a gaussian random field with hyperbolic variogram

$$\gamma(h) = \frac{|h|}{20 + |h|}$$

in a field 100×100 (unit mesh). The lefthand side of figure 4 shows how the variogram of a simulation evolves as the number of scans (number of updating of each pixel) increases. Such an exercise can be replicated. It turns out that less than 100 scans are required to reproduce the statistical fluctuations expected from the model. The righthand side of the same figure also shows the regional variograms of 5 independent simulations obtained after 100 scans.

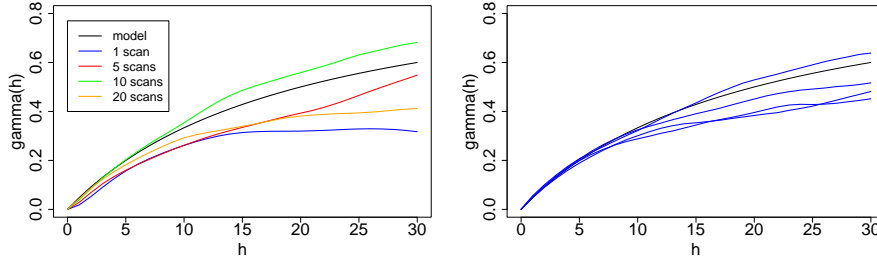


Fig. 4 Left, evolution of the variogram of a simulation versus the number of scans. Right, the variograms of 5 simulations after 100 scans

Other experiments have been conducted with the so-called stable variogram

$$\gamma_{\alpha}(h) = 1 - \exp\left(-\left(\frac{|h|}{a}\right)^{\alpha}\right) \quad 0 < \alpha \leq 2$$

by fixing the scale factor a to 30 and varying the shape parameter α from 0.5 to 2. The results are presented on figure 5. When $\alpha = 0.5$, the convergence is quite fast, and the statistical fluctuations are properly reproduced with no more than 50 scans. For $\alpha = 1$ and $\alpha = 1.5$, this number is respectively equal to 100 (exactly as for the hyperbolic variogram) and 150. When $\alpha = 2$, 200 scans are not sufficient to ensure the convergence.

How can these results be interpreted? These 4 variograms mainly differ by their behaviour at the origin, namely non differentiable with infinite slope for $\alpha = 0.5$, non differentiable with finite slope for $\alpha = 1$, once differentiable for $\alpha = 1.5$ and

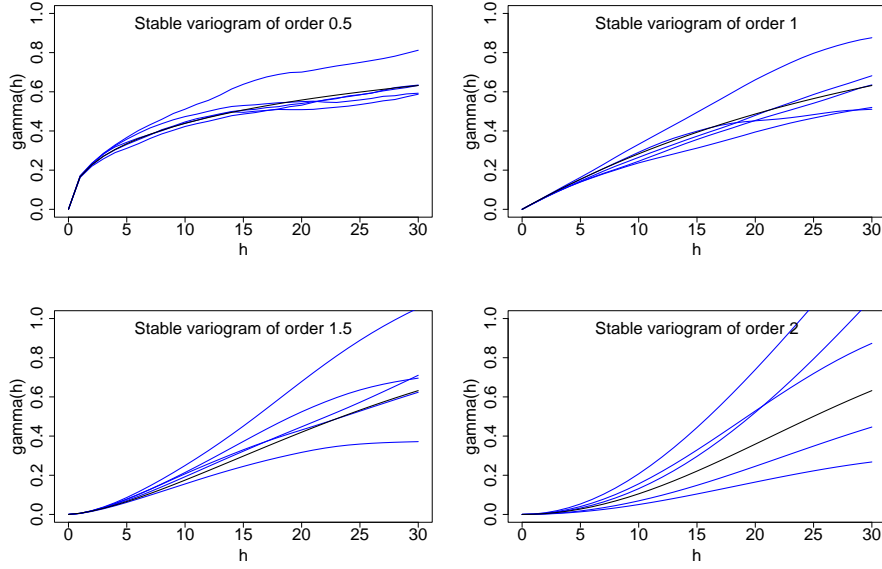


Fig. 5 Statistical fluctuations associated with stable variograms of order 0.5, 1 (exponential), 1.5 and 2 (gaussian) after 200 scans. The less regular the variogram at the origin, the faster the convergence of the simulation algorithm

infinitely differentiable for $\alpha = 2$. Accordingly, the rate of convergence seems to partly governed the degree of regularity of the variogram at the origin.

3 Generalizations

An attractive aspect of this algorithm is that it can be generalized in multiple ways.

A first generalization consists of generating the value at the pivot as a function of its current value. For instance, one can take $y_a^n = ry_a^c + \sqrt{1-r^2}u$ with $u \sim \mathcal{G}$ and $|r| < 1$. In the case $r = 0$, the algorithm of section 2 is retrieved. For $r \approx 1$, the new pivot value is not very different from the current one, which is often useful when this propagative algorithm is resorted to specify a candidate vector that has to be subsequently validated as in Metropolis-Hastings type algorithms [5]. Too large a discrepancy between the current and the new vector would result in too high a rejection rate.

A second generalization deals with a blocking strategy of pivots. Let us start by writing equation (2) like

$$y_b^n - C_{ab}y_a^n = y_b^c - C_{ab}y_a^c \quad b \in A_o$$

which expresses that the propagation algorithm preserves the (simple) kriging residuals of all components of A_o on the pivot a . We can wonder whether the approach remains valid when the pivot a is replaced by a block pivot, say $A \subset A_o$. If so, then it would suffice to generate y_A^n according to a centered gaussian distribution with covariance matrix C_{AA} (the covariance matrix of Y_A), and then update the other components using the formula

$$y_b^n - y_b^n(A) = y_b^c - y_b^c(A) \quad b \in A_o$$

where $y_b^n(A)$ and $y_b^c(A)$ are the (simple) kriging of the b^{th} component on those of A . It can be shown that this generalization effectively holds provided that the distribution \mathcal{S} used for generating the block pivots gives each component a non-negligible chance of being selected as a pivot at each iteration (see [5] for a proof). The algorithm using a blocking strategy is thus as follows:

- (i) put $y_{A_o}^c = 0$;
- (ii) generate $A \sim \mathcal{S}(A_o)$ and $y_A^n \sim \mathcal{G}(0, C_{AA})$;
- (iii) put $y_b^n = y_b^c + y_b^n(A) - y_b^c(A)$ for each $b \in A_o \setminus A$;
- (iv) put $y_{A_o}^c = y_{A_o}^n$ and goto (ii).

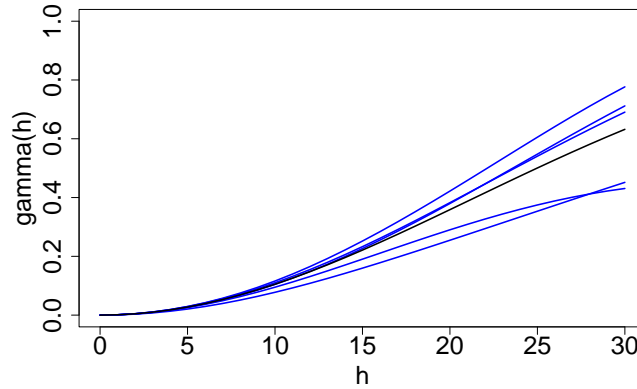


Fig. 6 Statistical fluctuations associated with the gaussian variogram after 50 scans using a blocking strategy of 5 pivots

Experiments have shown that resorting to block pivots is more efficient than point pivots. For instance, the simulation of the random vector with gaussian variogram of figure 5 has been reconsidered. The algorithm is run by successive scans. At each scan, the 10000 components are sorted at random and then batched into blocks of 5 pivots. Thus each scan consists of $10000/5 = 2000$ iterations. Using this approach,

the rate of convergence is 4 times faster than that obtained with point pivots (see figure 6).

Of course, both generalizations can be combined. Thus, y_A^n can be generated as a function of y_A^c , for instance by putting $y_A^n = ry_A^c + \sqrt{1-r^2}u_A$, where $|r| < 1$ and u_A is a centered gaussian vector with covariance matrix C_{AA} . Moreover, it can be established that the transition from $y_{A_o}^c$ to $Y_{A_o}^n$ is reversible as soon as the transition from y_A^c à y_A^n is revertible for each block pivot A .

4 About the rate of convergence

Suppose that simulation is initialized as a gaussian random vector with covariance matrix $C^{(0)}$ (this includes the case where it is set to zero). Then the random vector produced at the n^{th} scan is also gaussian and its covariance matrix $C^{(n)}$ is related to $C^{(0)}$ and C by the formula

$$C^{(n)} - C = B^n(C^{(0)} - C)B^n, \quad (3)$$

with $B = (Id - L)^{-1}U$, Id being the identity matrix, and L and U respectively denoting the lower and the upper triangular part of the matrix C [2]. It follows that the rate of convergence of the algorithm is geometric and governed by the square of the spectral radius of B .

Things behave similarly in the case where $y_\alpha^n = ry_\alpha^c + \sqrt{1-r^2}u$, and the expression of $C^{(n)}$ is similar to that of formula (3), except that the matrix B is replaced by another matrix B_r that is defined as

$$B_r = [Id - (1-r)L]^{-1}[rId + (1-r)U]$$

One could think that the rate of convergence is optimal when $r = 0$. This is not what is observed in practice. It has been experimentally observed that the minimum of the spectral radius of B_r is often reached for negative values of r ranging between -0.3 and -0.6 .

References

1. Bourdieu, P.: Science de la science et réflexivité. Editions Raisons d'agir, Paris (2001)
2. Desassis, N.: Note sur l'algorithme de propagation de Gibbs. Unpublished manuscrit, Mines-ParisTech (2011)
3. Galli, A., Gao, H.: Rate of convergence of the Gibbs sampler in the Gaussian case. Mathematical Geology, **33-6**, 653–677 (2001)
4. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. on Pattern Analysis and Machine Intelligence, **6-6**, 721–741 (1984)
5. Lantuéjoul, C.: Three dimensional conditional simulation of a Cox process using multisupport samples – Methodological aspects. Unpublished manuscrit, MinesParisTech (2010)