



# Sequential Simulation of a Conditional Boolean Model

Alan Troncoso<sup>1</sup>  · Xavier Freulon<sup>1</sup> ·  
Christian Lantuéjoul<sup>1</sup>

Received: 10 September 2020 / Accepted: 2 September 2021  
© International Association for Mathematical Geosciences 2021

**Abstract** Simulations of object-based models are widely used in various fields of geoscience to represent subsurface heterogeneities. A prototype of such a model is the Boolean model. Based on the stability and decomposition properties, a novel algorithm is proposed to simulate Boolean models, stationary or not, subject to foreground and background point conditions. This algorithm amounts to simulating two independent Boolean models, namely an avoiding Boolean model made of objects that avoid all conditioning data points and a hitting Boolean model, the objects of which hit one or several foreground conditioning data points. The first model is simulated using thinning techniques, and the second model is simulated by particle filtering. Overall, the algorithm produces exact simulations. It is very fast and easy to implement.

**Keywords** Stochastic geometry · Geostatistics · Poisson process · Particle filtering · Importance sampling

## 1 Introduction

Stochastic models are commonly used to characterize subsurface heterogeneities for the management of natural resources, such as water, minerals, oil and gas. These models have been developed to generate different synthetic geological architectures

---

✉ Alan Troncoso  
alan.troncoso@mines-paristech.fr  
Xavier Freulon  
xavier.freulon@mines-paristech.fr  
Christian Lantuéjoul  
christian.lantuejoul@mines-paristech.fr

<sup>1</sup> Centre de Géosciences, MinesParisTech, 35 rue Saint-Honoré, 77305 Fontainebleau, France

and property fields, the realizations of which (conditioned to observations) can be used to quantify the uncertainties and risks associated with different development options (Pyrzcz and Deutsch 2014; Joseph et al. 1993; Damsleth et al. 1992). Object-based models are categorical models used to represent facies or rock types, with object shapes mimicking those of actual geological entities or sedimentary bodies. The Boolean model is the simplest model in this generic family, as it assumes independence between random objects. This simplicity allows for the development of efficient algorithms that ensure the reproduction of the data and the statistical properties of the various outcomes (Vargas-Guzmán and Al-Qassab 2006; Allard et al. 2006; Lantuéjoul 2002).

A Boolean model  $X$  is defined as the union of independent random objects located at the points of a Poisson process. The model is stationary if the underlying Poisson process is homogeneous and the objects are identically distributed; it is nonstationary otherwise. This paper deals with the simulation of a Boolean model conditioned on a set of observation points. Among these points, the points in the background  $B$  defining the background condition, that is,  $X \cap B = \emptyset$ , and the points in the foreground  $F$  defining the foreground condition, that is,  $F \subset X$ , can be distinguished.

This simulation problem is usually solved using a Markov chain Monte Carlo algorithm (Gedler 1991; Lantuéjoul 2002; Allard et al. 2006); objects are added and removed according to a *birth-and-death* Markov kernel, the equilibrium distribution of which is the conditional distribution of the Boolean model. The algorithm is run on a population of random objects that fulfill the preset constraints until the equilibrium distribution is reached. It is quite general but may be slow due to a burn-in phase that is required to discard the objects of the initial population. Moreover, the rate of convergence of the Markov chain is not easy to establish. From a theoretical standpoint, the convergence rate is governed by the second largest eigenvalue of the Markov kernel that has to be estimated (Lantuéjoul 2002). From a practical standpoint, a Markov chain may stay in a metastable state of equilibrium for long periods. This prompted Kendall and Thönnès (1999) to develop an algorithm based on a specific coupling technique originally proposed by Propp and Wilson (1996). This algorithm provides exact, rather than approximate, conditional simulations, but it applies only to uniformly bounded objects. In this paper, a new, sequential and exact algorithm is proposed for simulating a Boolean model. It rests on two nonstandard properties of the general Boolean model:

- (i) Decomposition property a conditional Boolean model can be split into two independent Boolean models. Each object belongs to the hitting Boolean model or to the avoiding Boolean model depending on whether or not it contains foreground conditioning data points.
- (ii) Stability property a conditional Boolean model that is subject only to a background condition is a Boolean model with appropriate Poisson intensity and distribution of objects.

Whereas the simulation of the avoiding Boolean model requires nothing more than a simple thinning technique, that of the hitting Boolean model is performed using a particle filter that sequentially assimilates the foreground conditions.

The paper is organized as follows. Section 2 gives some background on the statistical characterization of random sets, as well as a brief summary of Boolean models. Their stability and decomposition properties are established. Section 3 is specifically

dedicated to the design of the developed algorithms. It includes the simulation of objects using weights to compensate for the presence of conditioning points and the bounded nature of the simulation domain. It also explains the general principles of the utilized particle filtering techniques. Section 4 presents a detailed illustration of the algorithm, followed by a discussion concerning the performance of the proposed algorithm in terms of its memory requirements, speed and range of applicability.

## 2 Boolean Model: A Few Properties

As a topologically closed random set, a Boolean model has its statistical properties completely specified by its hitting and avoiding functionals (Matheron 1975). These characterization tools are discussed first.

### 2.1 Hitting and Avoiding Functionals

Any random closed set  $X$  on  $\mathbb{R}^d$  has its statistical properties characterized by a functional, denoted as  $T_X$ , which assigns the probability that  $X$  intersects each compact subset  $K$  of  $\mathbb{R}^d$

$$T_X(K) = P\{X \cap K \neq \emptyset\}.$$

This functional is called the hitting functional of  $X$ . It plays the same role for  $X$  as a cumulative distribution function does for a random variable. Its complement to 1 is called the avoiding functional of  $X$

$$Q_X(K) = 1 - T_X(K) = P\{X \cap K = \emptyset\}.$$

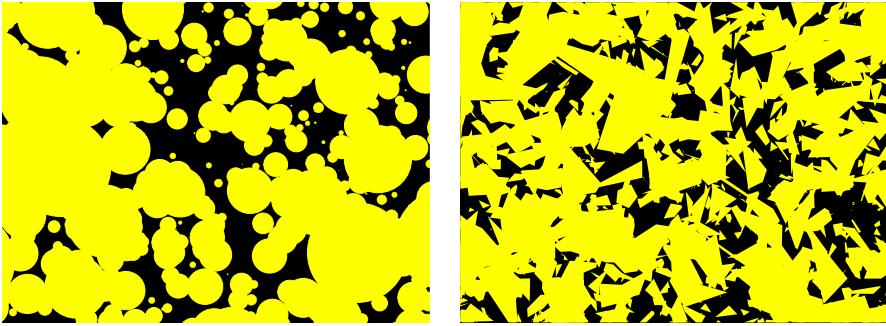
Although these functionals provide the same probabilistic information, it frequently happens that the properties of random sets are more conveniently expressed using one functional rather than the other. For example, consider the union of two independent random closed sets  $X$  and  $Y$ . The hitting and avoiding functionals of their union are equal to  $T_{X \cup Y} = T_X + T_Y - T_X \times T_Y$  and  $Q_{X \cup Y} = Q_X \times Q_Y$ , respectively. Their intersection is characterized by  $T_{X \cap Y} = T_X \times T_Y$  and  $Q_{X \cap Y} = Q_X + Q_Y - Q_X \times Q_Y$ .

### 2.2 Boolean Model

A Boolean model  $X$  is defined as the union of a family of independent random compact sets (called *objects* here for short) located at the points of a Poisson process. More formally, a Boolean model can be written as

$$X = \bigcup_{s \in \mathcal{P}} A_s, \quad (1)$$

where



**Fig. 1** Boolean models of discs (left) and Poisson polygons (right)

- $\mathcal{P}$  is a Poisson (point) process with intensity function  $(\theta(s), s \in \mathbb{R}^d)$  that is assumed to be locally integrable;
- $(A_s, s \in \mathbb{R}^d)$  is a family of nonempty and mutually independent objects. As a random closed set, each  $A_s$  possesses a hitting functional  $T_s(K) = P\{A_s \cap K \neq \emptyset\}$  and an avoiding functional  $Q_s(K) = P\{A_s \cap K = \emptyset\}$ .

Figure 1 shows a few examples of Boolean models.

Since the Poisson intensity is locally integrable, the corresponding Boolean model is a closed random set (Matheron 1975). Moreover, the number of objects hitting a compact subset  $K$  is Poisson-distributed with mean

$$\vartheta(K) = \int_{\mathbb{R}^d} \theta(s) T_s(K) ds, \quad (2)$$

(Matheron 1975; Lantuéjoul 2002), from which it results that the avoiding functional of the Boolean model is

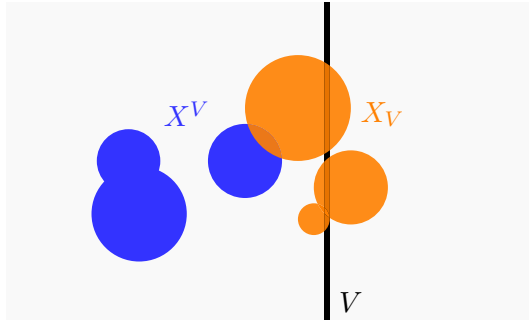
$$Q_X(K) = \exp\left(-\int_{\mathbb{R}^d} \theta(s) T_s(K) ds\right). \quad (3)$$

Starting from the objects of  $X$  and a subset  $V$  of  $\mathbb{R}^d$ , the avoiding and the hitting processes are respectively defined by

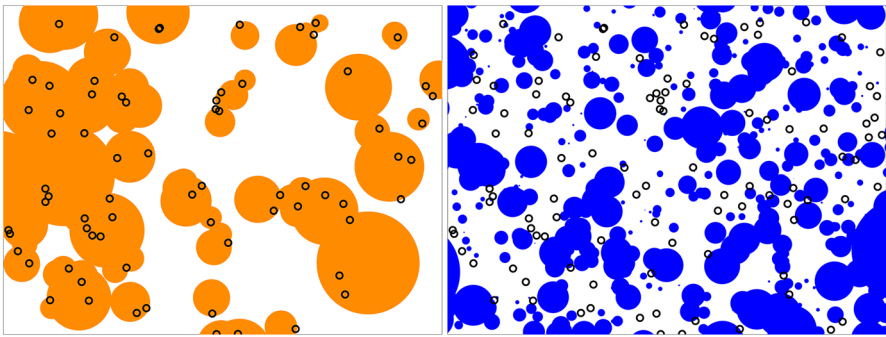
$$X^V = \bigcup_{\substack{s \in \mathcal{P} \\ A_s \cap V = \emptyset}} A_s \quad \text{and} \quad X_V = \bigcup_{\substack{s \in \mathcal{P} \\ A_s \cap V \neq \emptyset}} A_s. \quad (4)$$

Despite their cumbersome notations, these random sets possess simple interpretations (see Fig. 2).

The main properties of  $X^V$  and  $X_V$  are summarized in the following proposition.



**Fig. 2** Decomposition of  $X$  into  $X_V$  and  $X^V$



**Fig. 3** Example of sets of  $X_V$  type (left) and  $X^V$  type (right)

**Proposition 1**  $X = X_V \cup X^V$ . Moreover,  $X^V$  and  $X_V$  are two independent Boolean models with the following avoiding functionals:

$$Q_{X^V}(K) = \frac{Q_X(K \cup V)}{Q_X(V)} \quad Q_{X_V}(K) = \frac{Q_X(K) Q_X(V)}{Q_X(K \cup V)}. \tag{5}$$

The proof of this proposition is deferred to Appendix A.

Figure 3 shows an example of such random sets. Their objects are those of the Boolean model of discs in Fig. 1. The points in  $V$  are represented by small black circles. The decomposition of  $X$  into  $X^V$  and  $X_V$  reveals many hidden objects. Note also that the objects of  $X_V$  are much larger than those of  $X^V$ .

More generally, the construction of  $X^V$  and  $X_V$  can be iterated. Let  $U$  and  $V$  be two subsets of  $\mathbb{R}^d$ . Proposition 1 shows that the random set

$$X_U^V = (X^V)_U = (X_U)^V,$$

is a Boolean model with the avoiding functional

$$Q_{X_U^V}(K) = \frac{Q_{X^V}(K)}{Q_{X_U \cup V}(K)}. \tag{6}$$

This formula will play an important role in the subsequent decomposition of the conditional Boolean model into two simpler, independent models.

### 2.3 Conditional Boolean Model

Now, let  $X$  be a Boolean model along with two disjoint subsets  $B$  and  $F$  of  $\mathbb{R}^d$ , where  $F$  is finite. The Boolean model subject to the background condition  $X \cap B = \emptyset$  and the foreground condition  $F \subset X$  is denoted by  $X(B, F)$ . Its avoiding functional is related to that of  $X$  according to the following formula

$$Q_{X(B,F)}(K) = \frac{\sum_{C \subset F} (-1)^{|C|} Q_X(K \cup B \cup C)}{\sum_{C \subset F} (-1)^{|C|} Q_X(B \cup C)}, \quad (7)$$

where  $|C|$  denotes the cardinality of  $C$  (see Appendix B for the proof). This equation shows that, in general, a conditional Boolean model is not a Boolean model. A notable exception is when only background conditions are imposed. In that particular case,  $F = \emptyset$ , and the formula simplifies to

$$Q_{X(B,\emptyset)}(K) = \frac{Q_X(K \cup B)}{Q_X(B)} = Q_{X^B}(K)$$

due to Proposition 1. Hence

**Proposition 2** (Stability property) If  $F = \emptyset$ , then the conditional Boolean model is equivalent in distribution to the Boolean model  $X^B$

$$X(B, \emptyset) \equiv X^B. \quad (8)$$

In the general case, by applying formula (7) to the Boolean model  $X_F$  and replacing  $Q_{X_F}$  by its expression (5), we obtain

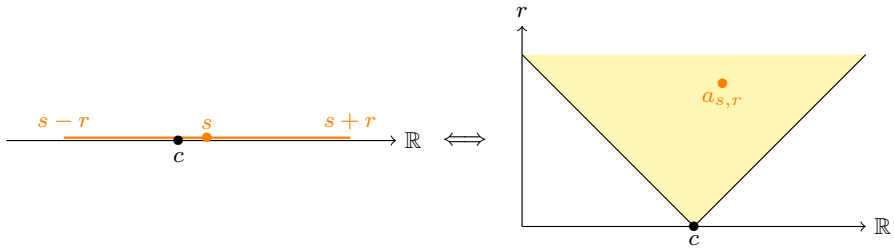
$$Q_{X_F(B,F)}(K) = \frac{Q_X(B \cup F)}{Q_X(K \cup B \cup F)} \frac{\sum_{C \subset F} (-1)^{|C|} Q_X(K \cup B \cup C)}{\sum_{C \subset F} (-1)^{|C|} Q_X(B \cup C)}.$$

Thus  $Q_{X_F(B,F)}(K)$  appears to be the product of two fractions. According to Eqs. (5) and (7), the first fraction is the reciprocal of  $Q_{X(B \cup F, \emptyset)}(K)$ , and the second is nothing but  $Q_{X(B,F)}(K)$ . We thus obtain the following property

**Proposition 3** (Decomposition property)

$$Q_{X(B,F)}(K) = Q_{X(B \cup F, \emptyset)}(K) Q_{X_F(B,F)}(K). \quad (9)$$

This formula constitutes the starting point of the conditional simulation algorithm that is to be presented in the next section.



**Fig. 4** Parameter space associated with the objects of a unidimensional Boolean model of random segments. The objects containing the conditioning point  $c$  belong to a cone pointed at  $(c, 0)$

### 3 Sequential Simulation of a Conditional Boolean Model

The objective of this section is to present a novel algorithm for simulating a Boolean model  $X$  in the domain  $D$ , subject to the conditions that a finite subset  $F$  is contained in the foreground and that an arbitrary subset  $B$  is contained in the background. For the sake of simplicity,  $F$  is assumed to be part of  $D$ .

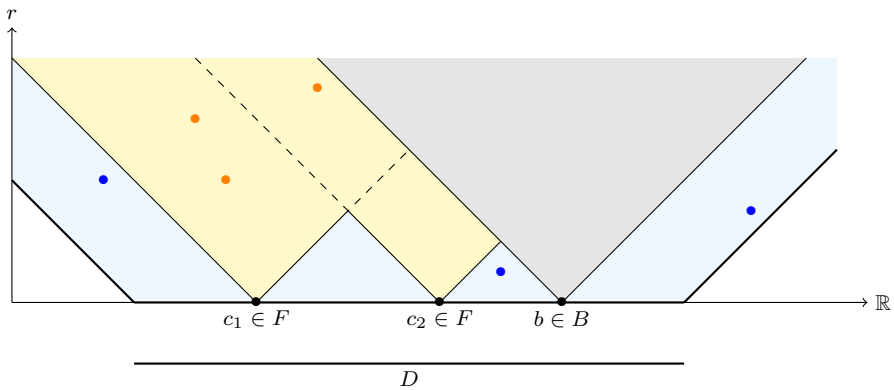
Let us start by applying the decomposition property to the Boolean model  $X_D$ . Since  $(X_D)_F = X_F$ , we obtain by Eq. (9)

$$Q_{X_D(B,F)}(K) = Q_{X_D(B \cup F, \emptyset)}(K) Q_{X_F(B,F)}(K). \tag{10}$$

This formula shows that  $X_D(B, F)$  has the same distribution as that of the union of two independent random sets, namely  $X_D(B \cup F, \emptyset)$  and  $X_F(B, F)$ . The first random set is subject only to background conditions. The second random set has a finite simulation domain. Each simulation algorithm will be presented in turn.

To gain a better understanding of how these algorithms work, it can help to show them running on a simple unidimensional Boolean model of random segments. The rationale for such a choice lies in the fact that each object is represented by two parameters (location and length) and is thus susceptible to a graphical representation on a plane. For instance, consider an object  $[s - r, s + r]$  of length  $2r$  located at the Poisson point  $s$ . This object contains a conditioning point  $c$  if and only  $s - r \leq c \leq s + r$ , or equivalently,  $|s - c| < r$ . In other words, the objects containing  $c$  belong to a cone pointed at  $(c, 0)$  in the parameter space (see Fig. 4).

Figure 5 shows what a conditional simulation looks like in the parameter space. The blue and yellow domains contain the points corresponding to the random objects of  $X_D(B \cup F, \emptyset)$  and  $X_F(B, F)$ , respectively. The yellow domain is a finite union of cones that have the foreground conditioning points as apexes. Each such cone must contain at least one point. The gray cone associated with the background conditioning point must be devoid of points. The number of points in the blue domain can be arbitrary.



**Fig. 5** Example of a conditional simulation of a unidimensional Boolean model of segments in the parameter space. The blue points represent the objects of the *avoiding* model  $X_D(B \cup F, \emptyset)$ , and the orange points represent those of the *hitting* model  $X_F(B, F)$

### 3.1 Simulation of $X_D(B \cup F, \emptyset)$

According to the stability property, we have  $X_D(B \cup F, \emptyset) \equiv X_D^{B \cup F}$ . This leads to the following thinning algorithm, where  $V$  is written in place of  $B \cup F$ :

---

#### Algorithm 1 Simulation of $X_D(V, \emptyset)$

---

**Require:** Simulation domain  $D$   
**Require:** Set of avoiding conditions  $V$   
 1:  $X = \emptyset$   
 2: generate the number of objects  $n \sim \text{Poisson}(\vartheta(D))$   
 3: **while**  $n > 0$  **do**  
 4:   generate a location  $s \sim \theta(\cdot) T.(D)/\vartheta(D)$   
 5:   generate an object  $A_s$  that hits  $D$   
 6:   **if**  $A_s \cap V = \emptyset$  **then**  $X = X \cup A(s)$   
 7:    $n = n - 1$   
 8: **end while**  
 9: **return**  $X \cap D$

---

The main difficulty in this algorithm is the simulation of an object  $A_s$  hitting  $D$ . This must be done on a case-by-case basis. The illustration section will show how to proceed with an example, but other examples can be found in Lantuéjoul (2013).

### 3.2 Simulation of $X_F(B, F)$

The task is rather simple when  $F$  is a singleton, for example,  $\{c\}$  (it suffices to generate successive independent copies of  $X_{\{c\}}^B$  until a nonempty copy has been produced), but it becomes increasingly complicated as the number of points in  $F$  increases (combinatorial issues appear when an object can cover several points). In this section,



the proposed strategy for coping with this problem is to sequentially assimilate the conditioning points using a particle filter (Del Moral 2004).

Suppose that  $F = \{c_1, \dots, c_N\}$  contains  $N$  points. These points can be totally ordered in an arbitrary way to define an increasing series of constraints  $F_0 \subset F_1 \subset \dots \subset F_n \subset \dots \subset F_N$ , with  $F_0 = \emptyset$  and  $F_n = \{c_1, \dots, c_n\}$  when  $n \geq 1$ . Consider the Boolean models  $X_n = X_{F_n}^B = X_{F_n}(B, \emptyset)$  and  $W_n = X_{\{c_n\}^{B \cup F_{n-1}}} = X_{\{c_n\}}(B \cup F_{n-1}, \emptyset)$ . A direct application of Eq. (6) gives  $Q_{X_n} = Q_{X_{n-1}} Q_{W_n}$ , which implies that  $X_n$  is distributed like the union of  $X_{n-1}$  and the independent ‘‘innovation’’  $W_n$

$$X_n \equiv X_{n-1} \cup W_n. \tag{11}$$

In other words, the sequence  $(X_1, \dots, X_N)$  is Markovian, and

$$Q_{X_F(B, F)}(K) = \prod_{i=1}^N Q_{X_{c_i}(B \cup F_{i-1}, \emptyset)}(K).$$

Starting from this Markov chain of Boolean models, a particle filter can now be designed. Instead of dealing with a single Markov chain of Boolean models, we consider the evolution of a population of  $K$  such Markov chains (or particles)  $(\xi^{(k)})_{1 \leq k \leq K}$  with each  $\xi_n^{(k)} = (X_i^{(k)})_{1 \leq i \leq n}$ . Each transition combines a proposition step and a resampling step, which are defined as follows:

- The proposition step depends only on the Markov dynamic (11). At transition  $n$ ,  $K$  candidate particles  $(\tilde{\xi}_n^{(k)})_{1 \leq k \leq K}$  are generated by setting  $\tilde{X}_i^{(k)} = X_i^{(k)}$  for each  $i < n$  and  $\tilde{X}_n^{(k)} = X_{n-1}^{(k)} \cup W_n^{(k)}$ .
- The resampling step checks the foreground conditions  $(F_n \subset \tilde{X}_n^{(k)})_{1 \leq k \leq K}$  satisfied by the candidate particles. For each  $k \in \{1, \dots, K\}$ , we independently set  $\xi_n^{(k)} = \tilde{\xi}_n^{(\dot{k})}$ , where  $\dot{k}$  is a random index generated from the distribution

$$\omega_n(k) = \frac{1_{F_n \subset \tilde{X}_n^{(k)}}}{\sum_{\ell=1}^K 1_{F_n \subset \tilde{X}_n^{(\ell)}}} \quad 1 \leq k \leq K.$$

It may happen that  $F_n$  is not totally contained in any  $\tilde{X}_n^{(k)}$ , in which case  $\omega_n$  is not defined. Then, the procedure must be restarted with a larger number of particles.

The Markov dynamic can be summarized by

$$\xi_{n-1} \xrightarrow{\text{proposition}} \tilde{\xi}_n \xrightarrow{\text{resampling}} \xi_n.$$

Provided that the number of particles is sufficiently large, all resampling probabilities considered are well defined. The particle filter has thus produced  $K$  gradual realizations of the conditional Boolean model. It then suffices to pick one of them at random to obtain a realization of  $X_F(B, F)$ .

---

**Algorithm 2** Sequential simulation of a conditional Boolean model  $X_F(B, F)$ 


---

**Require:** Two subsets  $B$  and  $F = \{c_1, \dots, c_N\} \subset D$

- 1: set  $\xi = (\emptyset)_{k \in 1:K}$
- 2: set  $C = \emptyset$
- 3: **for**  $n = 1$  to  $N$  **do**
- 4:   **if**  $n > 1$  **then**  $C = C \cup \{c_{n-1}\}$
- 5:   **for**  $k = 1$  to  $K$  **do**
- 6:     generate Boolean model  $W \sim X_{\{c_n\}}^{B \cup C}$  (cf. Algorithm 1)
- 7:     set  $\tilde{\xi}^{(k)} = \xi^{(k)} \cup W$
- 8:   **end for**
- 9:   set  $\omega_n(k) = \frac{1_{C \subset \tilde{\xi}^{(k)}}}{\sum_{k' \in 1:K} 1_{C \subset \tilde{\xi}^{(k')}}}$  for each  $k = 1, \dots, K$
- 10:   **for**  $k = 1$  to  $K$  **do**
- 11:     generate  $\ell \sim \omega_n$
- 12:     set  $\xi^{(k)} = \tilde{\xi}^{(\ell)}$
- 13:   **end for**
- 14: **end for**
- 15: generate  $k \sim \text{Uniform}(1 : K)$
- 16: **return**  $\xi^{(k)}$

---

The sequential simulation of a Boolean model hitting a finite set using a particle filter is detailed in Algorithm 2.

### 3.3 Summary

Starting from the results obtained in the previous sections, a sequential algorithm is proposed to simulate a Boolean model  $X$  in the domain  $D$ , subject to the foreground condition  $F \subset X$  and the background condition  $B \cap X = \emptyset$ . This conditional Boolean model  $X_D(B, F)$  is split into two independent Boolean models. The first is a Boolean model  $X_D(B \cup F, \emptyset)$  that avoids all observation points (see the top of Fig. 6). The second is a conditional Boolean model  $X_F(B, F)$  that contains all points in  $F$  and avoids all background points (see middle and bottom of Fig. 6). They are simulated independently and then recombined (Algorithm 3). Thus the approach relies on the simulation of a Boolean model avoiding  $B \cup F$  (Algorithm 1) and on the simulation of a conditional Boolean model on the finite set  $F$  (Algorithm 2). The concatenation of these two algorithms yields the general conditional simulation algorithm (see Algorithm 3).

---

**Algorithm 3** Sequential simulation of a conditional Boolean model  $X_D(B, F)$ 

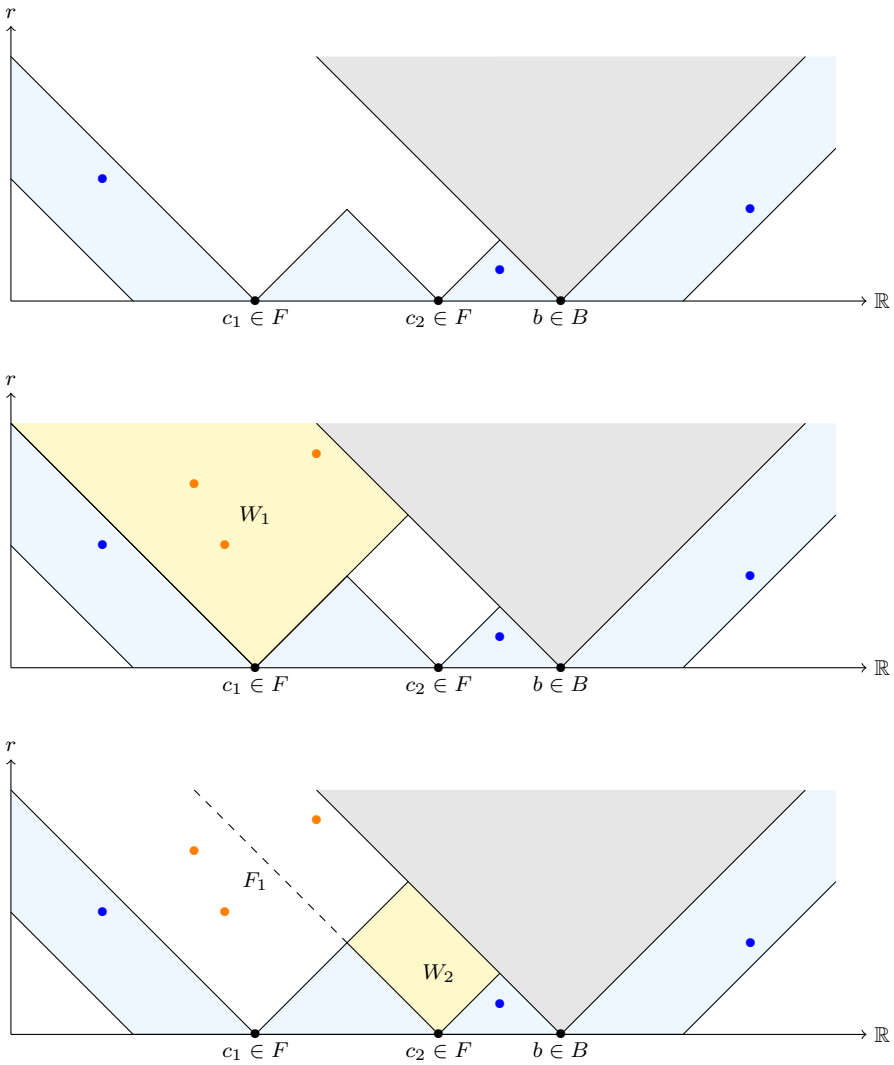

---

**Require:** Domain  $D$

**Require:** Two subsets  $B$  and  $F$  of  $D$

- 1: generate  $X_D(B \cup C, \emptyset)$  (cf. Algorithm 1)
- 2: generate  $X_F(B, F)$  (cf. Algorithm 2)
- 3: **return**  $X_D(B \cup C, \emptyset) \cup X_F(B, F)$

---



**Fig. 6** Summary of the conditional simulation algorithm using its representation in the parameter space. At first, the objects avoiding all conditioning data points are generated (top). Then all objects covering  $c_1$  are generated (center). The remaining objects covering  $c_2$  are finally generated (bottom)

## 4 Example

This section provides an illustration of the proposed sequential algorithm. The problem addressed is the simulation of a two-dimensional Boolean model of discs in a rectangular domain. The Boolean model is stationary, with Poisson intensity  $\theta$ , and the discs have their radii exponentially distributed,

$$f(r) = a \exp(-ar),$$

such that the mean disc radius is  $1/a$ , and the mean disc area is  $2\pi/a^2$ . The algorithm proposed in the previous section rests on non-conditional simulations of Boolean models in the given domain and on each foreground conditioning data point. These simulations are then thinned and combined to generate the final conditional simulation.

Consider first the case of the simulation domain. When the objects are statistically isotropic, the edge effects caused by the border of the simulation domain are more easily controlled in circular domains than rectangular domains. For that reason, the simulation is performed in the disc  $D_0$  circumscribed to the rectangular domain and then restricted to it. This disc is supposed to be centered at the origin, and its radius is denoted by  $r_0$ . To perform an exact simulation of the Boolean model in  $D_0$ , the general algorithm developed in Lantuéjoul (2013) can be applied. Appendix C gives the proofs in a slightly more general context. In a nutshell,  $X_{D_0}$  is a Boolean model with intensity function

$$\theta_{D_0}(s) = \theta \exp(-a(|s| - r_0)_+) \quad s \in \mathbb{R}^2,$$

where  $u_+ = \max(u, 0)$ . Moreover, the number of discs hitting  $D_0$  is Poisson-distributed, with mean value

$$\vartheta(D_0) = \theta\pi \left[ r_0^2 + 2r_0 \frac{1}{a} + \frac{2}{a^2} \right].$$

Regarding the object radii, their distribution is given by

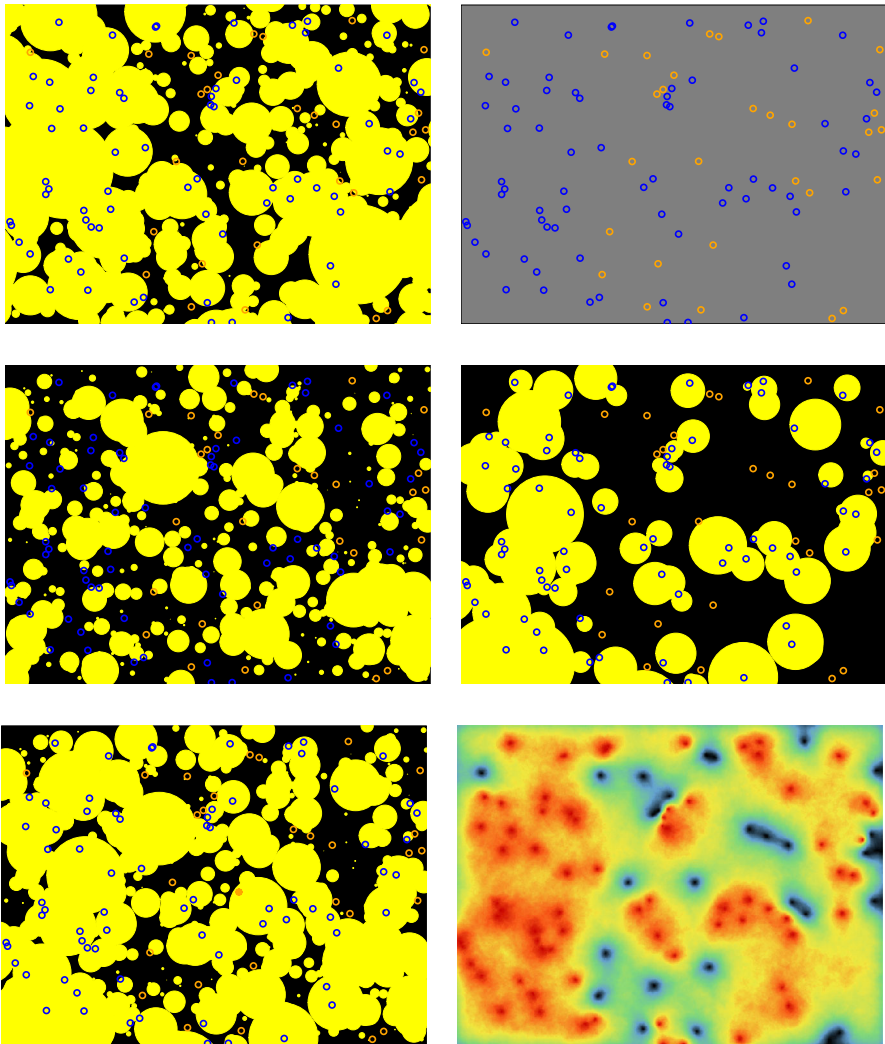
$$f_{D_0}(r) = \frac{r_0^2 g_{1,a}(r) + 2r_0 \frac{1}{a} g_{2,a}(r) + \frac{2}{a^2} g_{3,a}(r)}{r_0^2 + 2r_0 \frac{1}{a} + \frac{2}{a^2}} \quad r > 0,$$

where  $g_{\alpha,a}$  is the gamma distribution, with parameter  $\alpha$  and rate  $a$ . Thus,  $f_{D_0}$  can be simulated as a mixture of 3 gamma distributions. Finally, any object of radius  $r$  is uniformly located on the disc of radius  $r_0 + r$  centered at the origin.

Consider next the case of a conditioning point in the foreground, for example  $s$ . This can be seen as a circular simulation domain with a radius of 0. Thus, the same algorithm used before applies here, with  $r_0 = 0$ . In particular, the number of discs containing  $s$  is Poisson-distributed, with mean  $2\theta\pi/a^2$ . Moreover, the distribution of the radii of the discs containing the point  $s$  is not exponential but rather gamma, with parameter 3 and rate  $a$

$$f_{\{s\}}(r) = \frac{a^3}{2} \exp(-ar) r^2 \quad r > 0.$$

The difference between  $f$  and  $f_{\{s\}}$  is important. Indeed,  $f_{\{s\}}$  can be simulated as the sum of three independent variables distributed like  $f$ . It follows that the mean radius



**Fig. 7** Conditional simulation of a Boolean model of discs with exponential radii. Top left: a non-conditional simulation. Top right: 100 conditioning data points located at random. Middle left: a simulation of the avoiding Boolean model. Middle right: a simulation of the hitting Boolean model. Bottom left: a simulation of the conditional Boolean model. Bottom right: estimation of the probability of presence in the foreground based on 500 conditional simulations

of  $f_{\{s\}}$  is 3 times that of  $f$ . Regarding the corresponding mean areas, the ratio is equal to 6. This corroborates the idea that the more extended a disc is, the greater the chance it has to contain a fixed point.

The results of this algorithm are presented in Fig. 7. The considered Boolean model has a Poisson intensity of 10, and the rate of the exponential distribution is 7.22, which warrants an average background proportion of 30%. The top left image shows a non-

conditional simulation in an  $8 \times 6$  rectangular domain. From this simulation, 100 randomly scattered points are selected to serve as conditioning data points. Those points are shown in the top right image in blue or in orange, depending on whether they belong respectively to the foreground or the background. Subject to these observations, the middle left and middle right images are the simulations  $X_D(B \cup F, \emptyset)$  and  $X_F(B, F)$ , respectively (using 200 particles). Their union yields the conditional simulation of the bottom left image. Finally, an estimate of the probability of presence of each point in the foreground, based on 500 conditional simulations, is represented in the bottom right image.

## 5 Discussion and Conclusions

In this paper, we propose a sequential algorithm to simulate a Boolean model subject to point observations. To design this algorithm, two nonstandard properties of the general Boolean model have been used: (i) the decomposition property, which states that any conditional Boolean model can be seen as the union of two independent models, namely a Boolean model avoiding all observations and a conditional Boolean model defined on the foreground observations; and (ii) the stability property which states that a Boolean model that avoids background observations is still a Boolean model, and its simulation can be derived by thinning a non-conditional model.

This algorithm is versatile, as it applies to Boolean models, stationary or not, that are defined in Euclidean spaces or spheres of any dimensionality. The shapes and sizes of the objects can be arbitrary.

A Boolean model can be nonstationary in two ways: the intensity of the Poisson process is not constant, or (and) the features of the objects vary through space. From a formal standpoint, it is helpful to specify the distribution of the objects by their hitting functional. From a practical standpoint, it is more convenient to summarize each object by a set of parameters and to consider the corresponding distribution. In that sense, an object can be seen as a random point in a parameter space. This approach is particularly suited to the Boolean model because its population of objects can also be defined as a Poisson process on the family of the compact subsets in the workspace (Matheron 1975). Hence, the simulation of a Boolean model boils down to the simulation of a Poisson process defined in the parameter space. In the nonstationary case, simulations of Poisson processes are often feasible using acceptance/rejection algorithms. Note that the simulation of a stationary Boolean model, conditional or not, in a limited domain of a given workspace generally amounts to that of a nonstationary Boolean model. Indeed, remotely located objects have less chance to hit the simulation domain. If they do, they must be more extended. The case of a stationary Boolean model of discs presented in Sect. 4 is a typical example.

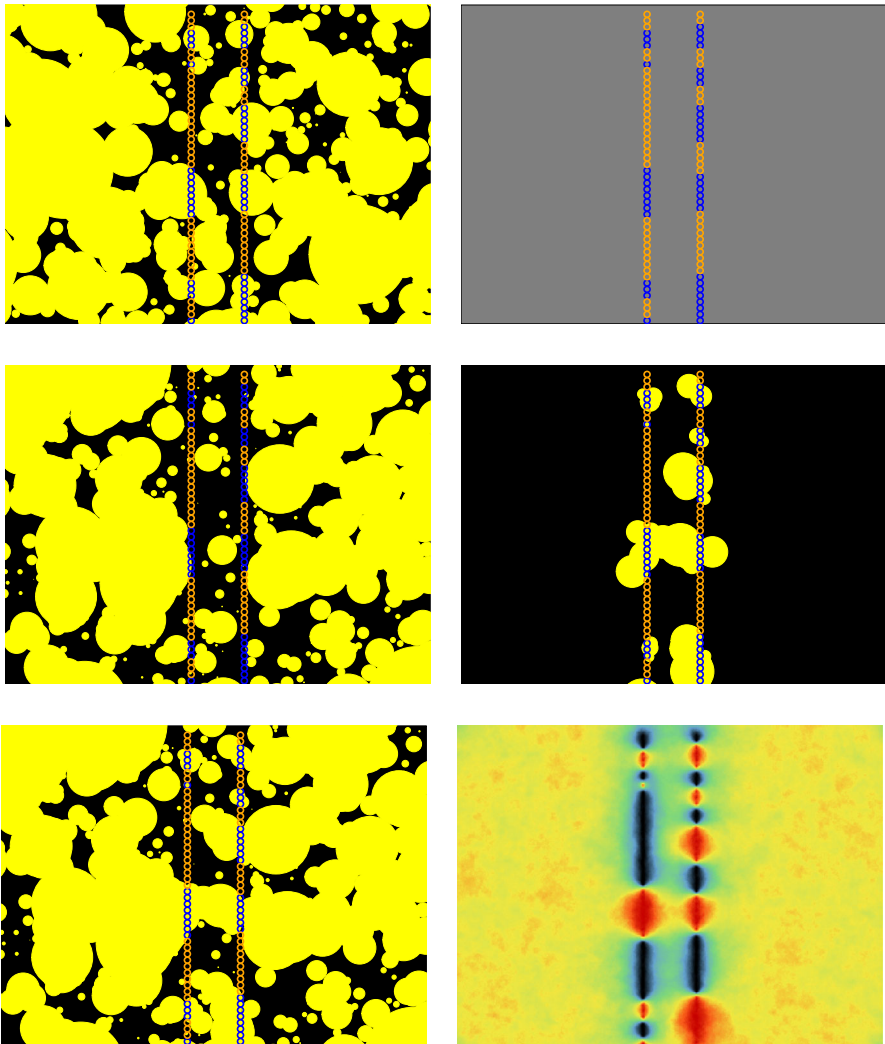
This algorithm is also efficient because the decomposition property enables us to target the conditioning effort on the random objects hitting the foreground observations. This component is achieved using a particle filter that sequentially assimilates the observed foreground points one by one. The particle filter thus performs the simulation in a finite number of steps, which is a real advantage over the standard Markov chain Monte Carlo algorithm for which the choice of the burn-in period and the assessment

of convergence are delicate. The counterpart is that one has to simulate the evolution of the so-called particles, which are Markov chains of Boolean models, with resampling steps to satisfy the constraints.

This algorithm is also quite fast. To give an idea of the order of magnitude of the speed, each of the conditional simulations presented is obtained in a few seconds on the authors' laptops, while minutes are required when using the birth and death algorithm. It turns out that the running time is practically proportional to the number of particles. It also depends on the number of observed foreground points, but not in a purely linear way. The processing time of a conditioning point in the foreground increases with its rank, as all constraints previously assimilated have to be considered. Moreover, there is no clear evidence that the order relation affects the running time.

The number of particles required to fulfill the constraints is a pending issue; they should be numerous enough to model the conditional distribution, but considering too large a number of them slows down the algorithm. In the example presented in Fig. 7 (Sect. 4), 100 observations are uniformly located over the domain. Fig. 8 shows the opposite situation, which is more representative of operational conditions, with 100 observations along two lines. The random objects can be intersected by zero, one or two lines. In both examples, 200 particles are sufficient to honor the constraints, although the geometrical configuration of the constraints differs greatly. In practice, the filter is used only for assimilating the points in the foreground; thus, the particles are usually made of a few objects that cover these points (see the middle right images in Figs. 7 and 8). Finally, Fig. 9 shows the estimates of the probability of presence in the foreground based on an increasing number of lines of observations. These estimates are computed as the average of 500 conditional simulations, where each is one conditioned using a filter with 200 particles. At this stage, it should be mentioned that the numbers of particles to produce the conditional simulations of Figs. 7 and 8 are relatively small because the conditioning data were extracting from unconditional simulations, so that the compatibility between the Boolean model and the conditioning data is automatically ensured. This is not always what happens in current practice. There are situations where the data are not perfectly compatible with the model considered; then no number of particles, however it may be, will allow the full respect of the conditioning. It is likely that the number of particles can be used as a criterion to quantify the adequacy between the model and the data, but this point deserves more investigation.

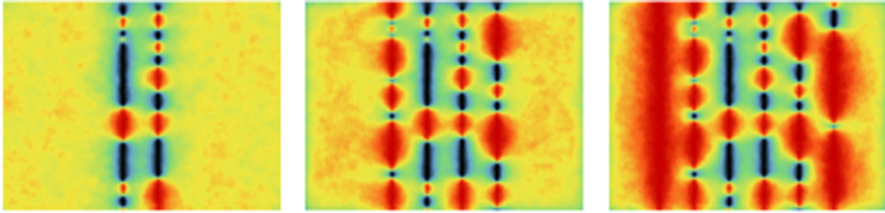
A recurring situation observed by the particle filtering practitioners is the weight degeneracy. After a number of steps, most weights become negligible in comparison with a few dominant ones. As a consequence, a limited number of particles are replicated, to the detriment of all others that are left over. This results in a dramatic reduction in the statistical fluctuations that one should expect. To alleviate this problem, powerful importance sampling techniques have been developed (Doucet et al. 2001; Del Moral 2004; Kroese et al. 2011). Instead of being generated from a prior distribution, particles are drawn from an auxiliary distribution that is better fitted to the observation to be assimilated. This change of distribution is corrected afterwards via so-called importance weights. Consider for example the Boolean model of discs presented in Sect. 4, and suppose that the point to be assimilated is closely surrounded by background conditioning points. If too many large radii are generated, then any



**Fig. 8** Conditional simulation of a Boolean model of discs with exponential radii. Top left: a non-conditional simulation. Top right: 100 conditioning data points arranged on lines. Middle left: a simulation of the avoiding Boolean model. Middle right: a simulation of the hitting Boolean model. Bottom left: a simulation of the conditional Boolean model. Bottom right: estimation of the probability of presence in the foreground based on 500 conditional simulations

disc covering the point is likely to cover background points, and thus to be rejected. Accordingly, only a few discs are accepted, from which it follows that most weights are zero, and most particles vanish. A solution to bypass this problem is to generate smaller discs (typically by replacing the exponential distribution of the radii by a gamma distribution with parameter less than 1). The importance weights compensate for the reduction in size of the discs. Also, in this example, the standard approach assigns values of 1 or 0 to the un-normalized weights, the value 1 or 0 depending on





**Fig. 9** Estimation of the probability of presence in the foreground starting from 100 observations over 2 lines (left), 200 observations across 4 lines (middle), and 300 observations across 6 lines (right). Each estimation is built using 500 conditional simulations

whether or not the point to be assimilated is covered by a disc. Another advantage of the importance sampling approach is that the un-normalized weights can take intermediary values between 0 and 1, which contributes to the survival of particles. In both examples, the conditioning pattern is drawn from a non-conditional simulation, which ensures the consistency between the model and the conditions. As mentioned by a reviewer, if the data configuration cannot be reproduced by the model, weights would degenerate whatever the resampling trick used. Linking the weight distribution as a measure of the model adequacy to the observations could be further investigated, but it was not considered in this work.

In the geosciences, realizations of three-dimensional stochastic models can be used as inputs for modeling the reactive transport in porous media. These models should be able to capture the heterogeneity of an aquifer. When the reservoir is an aggregate of geological objects, such as sand bodies or shale lenses, hierarchical models are commonly used. A discrete model first defines a partition; then each component is assigned petrophysical properties.

The discrete model often has more than two facies, which should preclude the use of a Boolean model in which the foreground is the union of independent random objects and the background is its complement. However, different Boolean models can be combined to design more complex models, in the same vein as a plurigaussian random field (Armstrong et al. 2003); independent Boolean models are coded into a categorical variable using a logical rule. To achieve a conditional simulation of such a pluriboolean random field, the point observations are first converted into point constraints on the latent Boolean models; then each Boolean model is independently simulated subject to its own local conditions; finally, the conditional Boolean models are recombined according to the logical rule. Note that if the multifacies model is nonstationary, then the latent Boolean models are also nonstationary.

In many situations, the objects in the reservoir are very large compared to the simulation domain. Typical cases are meandering channels. This raises the question of how to design a Boolean model of channels. A possible approach is to start with a Boolean model of segments. Then, each segment is used as a domain for the definition of a Gaussian process with a smooth covariance function. Finally, each Gaussian trajectory is dilated by a disc to assign it a nonzero width. It can be easily shown that the union of all dilated trajectories is a Boolean model. A similar approach for modeling channels was used in (Walgenwitz et al. 2015; Biver and Naumov 2011).

In both cases, the simulation of the stochastic model used to represent the heterogeneity within the reservoir boils down to that of Boolean models. An efficient and versatile algorithm for simulating a conditional Boolean model should facilitate the use of hierarchical models that are based on plain Boolean models.

In this paper, the conditioning data consist of point observations. However, there often exist a wealth of data provided by seismic information, history matching, well tests, and so on. In contrast to the observations considered in this paper, that information is not local. It is not certain that an algorithm such as particle filtering makes their integration straightforward. At a minimum, such information can be considered when performing the statistical inference of the Boolean parameters.

It frequently happens that the objects of a reservoir mutually interact (attraction or repulsion). In such a case, the object-based model must be built on a point process that is not Poisson (for example, a Gibbs or hard core process). Approximate formulas have been proposed in (Allard et al. 2006) in the case of Strauss processes (one of the simplest Gibbs processes) with conditional simulations performed iteratively. As the specific properties of the Poisson process have been exploited to design the proposed algorithm, a direct generalization does not seem possible for a number of point processes. An exception may be the Cox process (a Poisson process with a random intensity function). This point process probably gives rise to the simplest object-based models, where the objects are not independently located. However, the conditional simulation algorithm is not straightforward. Further work is still needed.

**Acknowledgements** The first author acknowledges the financial support of the Chilean National Agency for Research and Development (ANID)/Scholarship Program/DOCTORADO BECAS CHILE/2018-72190309. The authors warmly thank the three referees for their careful reading of the original manuscript, resulting in an improved and more accurate exposition.

## Appendix A: Proof of Proposition 1

Let  $K$  be a compact subset of  $\mathbb{R}^d$ . The hitting process  $X_V$  is disjoint from  $K$  if and only if each object of  $X$  either avoids  $V$  or hits  $V$  but not  $K$ ; this takes place with probability  $Q_s(V) + Q_s(K) - Q_s(K \cup V)$ . Similarly, the avoiding process  $X_V$  is disjoint from  $K$  if and only if each object of  $X$  either hits  $V$  or avoids  $K \cup V$ ; this takes place with probability  $T_s(V) + Q_s(K \cup V)$ . The avoiding functionals of  $X_V$  and  $X^V$  are explicitly calculated by first considering the restriction of the Poisson process  $\mathcal{P}$  to a compact domain  $D$  containing  $V$ , and then by extending  $D$  to  $\mathbb{R}^d$ . Denoting by  $\theta(D)$  the mean number of Poisson points in  $D$ , we obtain

$$Q_{X_V}(K) = \lim_{D \rightarrow \mathbb{R}^d} \sum_{n=0}^{\infty} \exp(-\theta(D)) \frac{\theta^n(D)}{n!} \left[ \int_D \frac{\theta(s)}{\theta(D)} [Q_s(V) + Q_s(K) - Q_s(K \cup V)] ds \right]^n,$$

$$Q_{X^V}(K) = \lim_{D \rightarrow \mathbb{R}^d} \sum_{n=0}^{\infty} \exp(-\theta(D)) \frac{\theta^n(D)}{n!} \left[ \int_D \frac{\theta(s)}{\theta(D)} [T_s(V) + Q_s(V \cup K)] ds \right]^n.$$

Once all calculations have been completed, this gives

$$\begin{aligned} Q_{X_V}(K) &= \exp\left(-\int_{\mathbb{R}^d} \theta(s) T_s(V) \left[1 - \frac{Q_s(K) - Q_s(V \cup K)}{T_s(V)}\right] ds\right) \\ &= \frac{Q_X(V) Q_X(K)}{Q_X(V \cup K)}, \\ Q_{X^V}(K) &= \exp\left(-\int_{\mathbb{R}^d} \theta(s) Q_s(V) \left[1 - \frac{Q_s(V \cup K)}{Q_s(V)}\right] ds\right) = \frac{Q_X(V \cup K)}{Q_X(V)}. \end{aligned}$$

Now, note that

$$\begin{aligned} 1 - \frac{Q_s(K) - Q_s(V \cup K)}{T_s(V)} &= P\{A_s \cap K \neq \emptyset \mid A_s \cap V \neq \emptyset\} \\ 1 - \frac{Q_s(V \cup K)}{Q_s(V)} &= P\{A_s \cap K \neq \emptyset \mid A_s \cap V = \emptyset\} \end{aligned}$$

are hitting functionals. It follows that  $X_V$  and  $X^V$  are Boolean models with intensity functions  $\theta(s) T_s(V)$  and  $\theta(s) Q_s(V)$ , respectively. The hitting functionals of their objects are given by the two formulas above. Moreover, note that

$$Q_{X_V}(K) Q_{X^V}(K) = \frac{Q_X(V) Q_X(K)}{Q_X(V \cup K)} \frac{Q_X(V \cup K)}{Q_X(V)} = Q_X(K).$$

Consequently,  $X_V$  and  $X^V$  are independent, and their union is equal to  $X$ .

### Appendix B: Avoiding Functional of $X(B, F)$

Recall that this avoiding functional is defined as

$$Q_{X(B,F)}(K) = P\{X \cap K = \emptyset \mid B \cap X = \emptyset \ \& \ F \subset X\},$$

or equivalently

$$Q_{X(B,F)}(K) = \frac{P\{X \cap (B \cup K) = \emptyset \ \& \ F \subset X\}}{P\{X \cap B = \emptyset \ \& \ F \subset X\}} := \frac{N}{D}.$$

To perform the calculation of the denominator, it is convenient to express the probability as the expectation of an indicator function

$$D = E\{1_{B \cap X = \emptyset} 1_{F \subset X}\}.$$

Now, the indicator function of  $F \subset X$  is expanded as

$$\begin{aligned} D &= E \left\{ 1_{B \cap X = \emptyset} \prod_{s \in F} 1_{s \in X} \right\} = E \left\{ 1_{B \cap X = \emptyset} \prod_{s \in F} (1 - 1_{s \notin X}) \right\} \\ &= E \left\{ 1_{B \cap X = \emptyset} \sum_{C \subset F} (-1)^{|C|} 1_{C \cap X = \emptyset} \right\} = \sum_{C \subset F} (-1)^{|C|} E \{ 1_{(B \cup C) \cap X = \emptyset} \} \\ &= \sum_{C \subset F} (-1)^{|C|} P\{(B \cup C) \cap X = \emptyset\} = \sum_{C \subset F} (-1)^{|C|} Q_X(B \cup C). \end{aligned}$$

The calculation of the numerator is now straightforward, as it can be deduced from the formula of the denominator by replacing  $B$  with  $K \cup B$

$$N = \sum_{C \subset F} (-1)^{|C|} Q_X(K \cup B \cup C).$$

We finally obtain

$$Q_{X(B,F)}(K) = \frac{\sum_{C \subset F} (-1)^{|C|} Q_X(K \cup B \cup C)}{\sum_{C \subset F} (-1)^{|C|} Q_X(B \cup C)},$$

which is precisely Eq. (7). It should be pointed out that this formula does not depend on the properties of the Boolean model. It is valid regardless of the random closed set considered.

### Appendix C: Exact Simulation of a Stationary Boolean Model of Discs

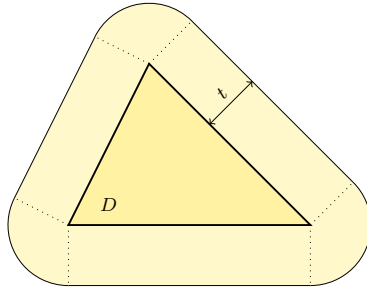
The domain of simulation  $D$  is assumed to be convex and bounded. According to Proposition 1,  $X_D$  is a nonstationary Boolean model. Its Poisson intensity function is

$$\theta_D(s) = \theta [1 - F(d(s, D))] \quad s \in \mathbb{R}^2$$

and the distribution of the disc radius at point  $s$  is

$$f_D(r | s) = \begin{cases} f(r) & \text{if } s \in D; \\ \frac{f(r)}{1 - F[d(s, D)]} & \text{if } r > d(s, D) > 0, \end{cases}$$

where  $\theta$  and  $f$  are respectively the Poisson intensity and the probability density function of the radii of the stationary Boolean model  $X$  (the cumulative pdf of  $f$  is denoted by  $F$ ), and  $d(\cdot, D)$  is the distance to the convex set  $D$ . These expressions derive from the simple fact that a disc of radius  $r$  implanted at  $s$  hits  $D$  if and only if  $r$  exceeds  $d(s, D)$ .



**Fig. 10** Illustration of Steiner’s formula. The length of the curve surrounding at distance  $t$  the convex domain  $D$  with finite perimeter  $p(D)$  is equal to  $p(D) + 2\pi t$

To simulate  $X_D$ , the number of discs hitting the domain is first drawn from a Poisson variable with parameter

$$\vartheta(D) = \int_{\mathbb{R}^2} \theta_D(s) \, ds.$$

This integral can be explicitly calculated using the change of variable  $t = d(s, D)$ . The case  $t = 0$  corresponds to the points of  $D$ , so that its contribution to  $\vartheta(D)$  is  $\theta a(D)$ , where  $a(D)$  is the area of  $D$ . The case  $t > 0$  corresponds to the points outside  $D$ . As  $D$  is convex, the length of the curve at distance  $t$  from  $D$  is equal to  $p(D) + 2\pi t$ , where  $p(D)$  is the perimeter of  $D$  (this is a simple application of Steiner’s formula illustrated in Fig. 10). Thus its contribution to  $\vartheta(D)$  is  $\theta[p(D) + 2\pi t][1 - F(t)]$ . It follows that

$$\begin{aligned} \vartheta(D) &= \theta \left[ a(D) + \int_0^\infty (p(D) + 2\pi t) \int_t^\infty f(u) \, du \, dt \right] \\ &= \theta [a(D) + p(D)m_1 + \pi m_2], \end{aligned} \tag{12}$$

where  $m_1$  and  $m_2$  are the first two moments of  $f$ . Discarding the case where the variance of  $f$  is infinite (in this case  $X_D = D$ ),  $\vartheta(D)$  is finite. Then,  $X_D$  is the union of a Poisson number of objects, which implies that it can be exactly simulated.

Then each disc is simulated independently with its center drawn from  $\theta_D(s)/\vartheta(D)$ , and conditionally to its location, its radius is drawn from  $f_D(r|s)$ . A simple way to simulate the location of a disc hitting  $D$  is to proceed by acceptance-rejection using an integrable, radially symmetric auxiliary function (for example,  $1 - F(d(s, D_0))$ , where  $D_0$  is a disc that encloses  $D$ ).

Alternatively, it is possible to first simulate the radius of the objects, and then their location. To do this, the radius distribution to consider is

$$f_D(r) := \int_{\mathbb{R}^2} f_D(s) f_D(r | s) \, ds = f(r) \frac{a(D) + p(D)r + \pi r^2}{a(D) + p(D)m_1 + \pi m_2},$$

which can be expressed as a mixture of weighted distributions

$$f_{D_0}(r) = w_0 f_0(r) + w_1 f_1(r) + w_2 f_2(r)$$

with

$$f_i(r) = \frac{r^i f(r)}{m_i} \quad i = 0, 1, 2$$

and the  $w_i$  are respectively proportional to  $a(D)$ ,  $p(D) m_1$ , and  $\pi m_2$ . Once a radius  $r$  has been generated, then the center of the disc is uniformly located in the domain  $\{s \in \mathbb{R}^2 : d(s, D) < r\}$ .

Similar results hold in more than two dimensions. Consider for instance the case of a bounded, convex domain  $D$  in  $\mathbb{R}^3$ . There are three important features of  $D$  to consider, namely its volume  $v(D)$ , the surface area of its boundary  $s(D)$ , and the integral of its mean curvature  $n(D)$  (Schneider and Weil 2008). The expressions for  $\theta_D(s)$  and  $f_D(r | s)$  are exactly the same. On the other hand, it can be shown that

$$\vartheta(D) = \theta [v(D) + s(D) m_1 + n(D) m_2 + \omega_3 m_3],$$

where the  $m_i$ 's are the first three moments of  $f$ , and  $\omega_3 = 4\pi/3$  is the volume of the unit ball in  $\mathbb{R}^3$ .  $\vartheta(D)$  is finite if and only if  $f$  admits a finite moment of order 3. Regarding the distribution of the objects of  $X_D$ , we have

$$f_D(r) = f(r) \frac{v(D) + s(D)r + n(D)r^2 + \omega_3 r^3}{v(D) + s(D)m_1 + n(D)m_2 + \omega_3 m_3},$$

which can be expressed as a mixture of 4 distributions. In practice, it frequently happens that  $D$  is a rectangular parallelepiped. Denoting its edge lengths by  $a$ ,  $b$  and  $c$ , we have  $v(D) = abc$ ,  $s(D) = 2(ab + ac + bc)$  and  $n(D) = \pi(a + b + c)$ .

## References

- Allard D, Froidevaux R, Biver P (2006) Conditional simulation of multi-type non stationary Markov object models respecting specified proportions. *Math Geol* 38(8):959–986
- Armstrong M, Galli A, Beucher H, Le Loc'h G, Renard D, Doligez B, Eschard R, Geffroy F (2003) Plurigaussian simulation in geosciences. Springer, Berlin
- Biver P, Naumov S (2011) Recent advances in geostatistical tools in order to capture realistic geological features. In: 73rd EAGE conference and exhibition-workshops 2011. European Association of Geoscientists & Engineers, pp cp–239
- Damsleth E, Tjolsen CB, Omre H, Haldorsen HH et al (1992) A two-stage stochastic model applied to a North Sea reservoir. *J Petrol Technol* 44(04):402–486
- Del Moral P (2004) Feynman-Kac formulae: genealogical and interacting particle systems with applications, probability and its applications. Springer, New York
- Doucet A, de Freitas N, Gordon N (2001) Sequential Monte Carlo methods in practice. Springer, New York
- Gedler G (1991) Algorithme de simulation du schéma booléen conditionnel. Technical report, Ecole des Mines de Paris, Centre de Géostatistique
- Joseph P, Hu L, Dubrule O, Claude D, Crumeyrolle P, Lesueur J, Soudet H (1993) The Roda deltaic complex (Spain): from sedimentology to reservoir stochastic modelling. In: Echard R, Doligez B

- (eds) Subsurface reservoir characterization from outcrop observations. Editions Technip, Paris, pp 97–109
- Kendall W, Thönnies E (1999) Perfect simulation in stochastic geometry. *Pattern Recogn* 32:1569–1586
- Kroese D, Taimre T, Botev Z (2011) *Handbook of Monte Carlo methods*. Wiley, Hoboken
- Lantuéjoul C (2002) *Geostatistical simulation: models and algorithms*. Springer, Berlin
- Lantuéjoul C (2013) Exact simulation of a Boolean model. *Image Anal Stereol* 32:101–105
- Matheron G (1975) *Random sets and integral geometry*. Wiley, New York
- Propp J, Wilson D (1996) Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Struct Algorithms* 9:223–252
- Pyrz MJ, Deutsch CV (2014) *Geostatistical reservoir modeling*. Oxford University Press, Oxford
- Schneider R, Weil W (2008) *Stochastic and integral geometry*. Springer, New York
- Vargas-Guzmán J, Al-Qassab H (2006) Spatial conditional simulation of facies objects for modeling complex clastic reservoirs. *J Petrol Sci Eng* 54–1(2):1–9
- Walgenwitz A, Allard D, Biver P (2015) Flexible objects—a way to generate more realistic object-based simulations. In: *Petroleum geostatistics 2015*. European association of geoscientists & engineers, pp cp–456