Perfect Simulation in Stochastic Geometry

Wilfrid S. Kendall* and Elke Thönnes†,
Department of Statistics, University of Warwick‡

Abstract

Simulation plays an important rôle in stochastic geometry and related fields, because all but the simplest random set models tend to be intractable to analysis. Many simulation algorithms deliver (approximate) samples of such random set models, for example by simulating the equilibrium distribution of a Markov chain such as a spatial birth-and-death process. The samples usually fail to be exact because the algorithm simulates the Markov chain for a long but finite time, and thus convergence to equilibrium is only approximate. The seminal work by Propp and Wilson made an important contribution to simulation by proposing a coupling method, *Coupling from the Past (CFTP)*, which delivers perfect, that is to say exact, simulations of Markov chains. In this paper we introduce this new idea of perfect simulation and illustrate it using two common models in stochastic geometry: the dead leaves model and a Boolean model conditioned to cover a finite set of points.

Keywords:
BOOLEAN MODEL, CONDITIONAL BOOLEAN MODEL, COUPLING FROM THE PAST, DEAD LEAVES MODEL, MARKOV CHAIN MONTE CARLO, PERFECT SIMULATION, SPATIAL BIRTH AND DEATH PROCESS, STOCHASTIC GEOMETRY

---

*Research supported by EPSRC grant GR/L56831.

†Research supported by EPSRC earmarked studentship and University of Warwick graduate award.

‡Postal address: Dept. of Statistics, University of Warwick, Coventry, CV4 7AL, UK.

1 Introduction

The purpose of this paper is to introduce a new idea in simulation to the image-processing community.

A wide range of areas, including the field of statistical image analysis and stochastic geometry, share a common problem: it is relatively easy to define certain probability densities, but their normalising constants cannot be computed in closed form and conventional numerical computation is infeasible.

For example, in stochastic geometry we define point process models using densities with respect to a Poisson point process on a bounded window W. Consider the Strauss point process [1] whose density is given for $\gamma \in (0,1)$ by

$$p(x) = \frac{1}{K(\gamma)}\, \gamma^{s(x)}, \tag{1}$$

where $s(x)$ denotes the number of neighbour pairs in the point pattern x. Two points of a pattern are neighbours if they are closer than a specified interaction range R. This is an example of a density for which computation of the normalizing constant $K(\gamma)$ is infeasible.

In statistical mechanics models are often specified as probability densities

$$p(x;\theta) = \frac{1}{K(\theta)} \exp(-E(x;\theta)). \tag{2}$$

Typically the *energy* $E(x;\theta)$ is specified explicitly in the definition of the model, but the *partition function* $K(\theta)$ is analytically intractable, being implicitly defined as the normalising constant.

In Bayesian statistics one combines a likelihood $l(y|\psi)$ for the data y given the parameter ψ with a prior density $p(\psi)$ to obtain the conditional distribution of the parameter ψ given the data y. This posterior distribution is proportional (as a function of ψ) to the joint density of data and parameter $l(y|\psi) \times p(\psi)$. Once again an analytic evaluation of the constant of proportionality is often infeasible.

Finally in computer science we also find examples where the computation of the normalizing constant is not feasible. In the theory of randomised algorithms one needs to sample from a uniform density on a high-dimensional bounded convex set K in order to compute efficiently the volume of K. But determination of the normalising constant of this density is equivalent to finding the volume of K, which is what was required in the first place!

All these examples require the equivalent of integration of a density. Simulation-based solutions to this problem are available and in the statistical context are referred to as Markov Chain Monte Carlo (MCMC). The basic idea is as follows:

(1) Find a Markov chain whose equilibrium distribution is the target distribution and which can be simulated without knowing the normalising constant. (Typically, but *not* always, the task is simplified by assuming the chain is reversible.)

(2) Simulate the Markov chain long enough that it is at least approximately in equilibrium. Then use samples of the Markov chain to estimate the normalising constant.

There are many ways of constructing such Markov chains and special features of the problem should be taken into account to deliver efficient algorithms. MCMC methods often use the Gibbs Sampler or (more generally) a Metropolis-Hastings algorithm. These are designed such that the Markov chain has the target equilibrium distribution and can be simulated without knowing the normalizing constant. An overview of these two methods is given by the article by Roberts in [2].

Most commonly, Markov chains used to simulate point processes such as the Strauss process are spatial birth/death processes, i.e. as time passes random patterns of points are born and die in a Markov fashion. Helpful surveys on MCMC methods and applications for point processes can be found in [3, 4].

After defining the Markov chain, one needs to determine for how long the simulation should proceed in order to be close to equilibrium. Convergence diagnostics have been developed to assist the decision on the length of the burn-in (that is to say, the number of initial iterations which should be discarded); for a recent review see [5]. Although these diagnostics might warn when convergence has not yet been reached, they cannot guarantee that the Markov chain is in or close to equilibrium.

Recently Propp and Wilson [6] have devised an ingenious algorithm, which augments the simulation algorithm to produce a certificate whether equilibrium has been attained. Thus we can obtain a sample of the equilibrium distribution which is *exact*, not approximate. In the following section we describe this algorithm, *Coupling from the Past* (CFTP), give a simple example based on random walks, and summarize how CFTP has already been applied successfully to the context of point processes. (For a comprehensive overview on applications of CFTP see [7].) By way of detailed illustration, in the third and fourth sections we develop CFTP algorithms for two common models for images arising in stochastic geometry: the dead leaves model (for which the algorithm is almost trivial) and a conditional variate of the Boolean model (for which the algorithm is definitely non-trivial). The last section summarizes our conclusions.

2 Exact Simulation

2.1 Coupling from the Past

The underlying idea of Coupling from the Past (CFTP) is delightfully simple. Assume a finite state-space $\mathcal{S}$. Consider the simulation of a Markov chain on $\mathcal{S}$ using independent identically distributed random transition maps $H_k : \mathcal{S} \to \mathcal{S}, k \in \mathbb{Z}$. The transition map H_k is a random map which specifies for each state of the state space the transition of the chain from time $k-1$ to time k. The common distribution of the H_k's is defined using the transition probabilities p_{ij} of the Markov chain: it is required that

$$\mathbb{P}[H_k(i) = j] \quad = \quad p_{ij}$$

and we simulate the Markov chain X by

$$X_k \quad = \quad H_k(X_{k-1}), \qquad k \in \mathbb{Z}.$$

If we somehow were able to start the Markov chain X at time $-\infty$, then (assuming the Markov chain did converge to a statistical equilibrium) at time 0 the chain would be in exact equilibrium. Of course we could never afford to simulate a chain for an infinite amount of time. Instead CFTP enables us to sample such an infinitely long simulation at time 0 by reconstructing it only over a *finite* time interval stretching back into the recent past.

The algorithm works as follows. Choose a time $-T_1 < 0$. Using a sequence of transition maps we start the chain at time $-T_1$ from every possible state and run it up to time 0. As the same realisations of transition maps are used for all initial states, paths will coalesce if they meet. If all paths have coalesced into a single path by time 0, the state at time 0 is distributed according to the equilibrium distribution. One can argue intuitively for this as follows: if we had started the chain at time $-\infty$, it would have been in one of the possible states at time $-T_1$ and hence would have yielded the same state at time 0 as our simulations.

If the paths have not coalesced into a single path by time 0, we have not observed enough of the past to reconstruct the state of the infinite-length simulation at time 0. Hence we choose a value $-T_2 < -T_1$ and repeat above procedure using T_2 instead of T_1. Note that it is essential that when extending backwards we use the same realisations of transition maps over the range $[-T_1, 0]$ as before. Step by step we venture further into the past until we achieve complete coalescence.

The individual transition maps can be combined into a *stochastic flow*, i.e. a random map $F_{S,T} : \mathcal{S} \to \mathcal{S}$ defined by the composition

$$F_{S,T} = H_{S+1} \circ H_{S+2} \circ \cdots \circ H_T.$$

If we start the chain in $i \in \mathcal{S}$ at time $-T$, then the state of the resulting path at time 0 is $F_{-T,0}(i)$. All possible paths will have coalesced if the image of $F_{-T,0}$ is a single state. Let

$$T_C := \inf\{T : \#(\text{Image}(F_{-T,0})) = 1\}.$$

If $T \geq T_C$ we have

$$F_{-T,0}(i) = F_{-T,0}(j) = F_{-T_C,0}(i) \quad \text{for all } i, j \in \mathcal{S}.$$

Thus as long as we start the chain at time $-T_C$ or earlier the resulting path will sample the same state $F_{-T_C,0}(i)$ at time 0. If we could start the chain at time $-\infty$ then even this infinite time simulation would sample $F_{-T_C,0}(i)$ at time 0. As the Markov chain converges to a statistical equilibrium the infinite time simulation has this equilibrium distribution at time 0. Thus we may think of $F_{-T,0}(i)$ as the result of a *virtual simulation from time* $-\infty$, which is in equilibrium at time 0.

In practice the question of whether CFTP is achievable depends chiefly on whether we can detect coalescence *without* having to examine each of the (typically very large number of) possible simulation results $F_{-T,0}(i)$ as

i ranges through S. If the state-space is partially ordered and the Markov chain is *monotone*, that is to say its transitions maintain this partial order, then this can be relatively easy to arrange.

Let $\overset{\mathcal{S}}{\preceq}$ be the partial order of the state-space $\mathcal{S}$. We call a Markov chain monotone, if it has transition maps $\{H_k, k \in \mathbb{Z}\}$, such that

$$H_k(x) \overset{\mathcal{S}}{\preceq} H_k(y) \qquad \text{for all } k \text{ and whenever } x \overset{\mathcal{S}}{\preceq} y.$$

A Markov chain is anti-monotone, if it reverses the partial order, i.e.

$$H_k(x) \overset{\mathcal{S}}{\succeq} H_k(y) \qquad \text{for all } k \text{ and whenever } x \overset{\mathcal{S}}{\preceq} y.$$

As the state-space $\mathcal{S}$ is finite, it will have a maximal and minimal state. For a monotone Markov chain, if the paths started from the minimal and the maximal state of $\mathcal{S}$ have coalesced, all paths from intermediate states must have also coalesced. Hence we need to only monitor the paths started from the minimal and the maximal state. We will refer to this concept as *sandwiching*.

2.2 A Reflecting Random Walk Example

Examples of CFTP in [6] include an Ising model with a very large state-space, of clear relevance to problems in image-processing where one uses such models as local priors for binary image reconstruction. Here we introduce the idea of CFTP by considering the example of a simple random walk X with reflecting barriers at 0 and N, extending an example in [6, 8].

Of course one can easily calculate the equilibrium distribution π of X, but suppose we want to use MCMC to sample from π. The above Markov chain can be simulated using an independent sequence of transition maps $\{H_k : k \in \mathbb{Z}\}$, such that (for fixed k) with probability p we have

$$H_k(i) \quad = \quad \min\{i+1, N\} \quad \text{for } i \in \{0, \ldots, N\}$$

and with probability $1-p$ we have

$$H_k(i) \quad = \quad \max\{i-1, 0\} \quad \text{for } i \in \{0, \ldots, N\}.$$

Notice that (for fixed k) $H_k(0), H_k(1), \ldots, H_k(N)$ are highly dependent: for example if

$$H_k(0) = 1 \quad \text{then} \quad H_k(1) = 2, \ \ldots, \ H_k(N-1) = N, \ H_k(N) = N.$$

Conventional MCMC methods compose this sequence to produce a stochastic flow $F_{S,T} : \mathcal{S} \to \mathcal{S}$ and deliver $\{X_T, T \geq 0\}$ via

$$X_T = F_{0,T}(i) \qquad \text{if} \qquad X_0 = i.$$

Then the equilibrium distribution of X is approximated by sampling $X_T(i) = F_{0,T}(i)$ for large T. In contrast CFTP samples $Y_T(i) = F_{-T,0}(i)$, but (as Propp and Wilson point out for the general case, and has indeed been part of probabilistic folk-lore for a long time) $Y_T(i)$ has the same distribution as $X_T(i)$.

Now, if we choose $T \geq T_C$, where T_C is defined as in the previous section, then

$$F_{-T,0}(j) = F_{-T,0}(i) \qquad \text{for all } j \in \mathcal{S},$$

and so $Y_T(i)$ does not depend on i. As the Markov chain X_T converges to a statistical equilibrium and $Y_T(i)$ has the same distribution as $X_T(i)$ for all $i \in \mathcal{S}$ it follows that $Y_\infty = \lim_{T\to\infty} Y_T$ exists and has equilibrium distribution. At the same time

$$Y_T(i) = Y_\infty(i) \qquad \text{for } T \geq T_C,$$

so we only need to observe Y_T until $T \geq T_C$. In practice, we determine whether $T \geq T_C$ by checking whether all paths of our simulations have coalesced. Note that this Markov chain is monotone, since

$$H_k(i) \leq H_k(j) \quad \text{if} \quad i \leq j.$$

Thus we only need to check that the path started from state 0 and the path started from state N yield the same state at time 0.

In our example it is easy to check that T_C is almost surely finite. In general T_C is distributed as the first time when the paths started from all possible initial states have coalesced into one. This time is almost surely finite for all finite state-space Markov chains which are irreducible and ergodic – exactly those finite Markov chains for which a distributional convergence theorem holds!

2.3 Exact Simulation of Point Processes

The original CFTP algorithm was developed for Markov chains with a finite state-space. Murdoch and Green [9] show how CFTP can be extended to various MCMC samplers on a continuous state-space using the concept of "small sets" (see [10, 11]).

Point process theory is another area in which CFTP has been successfully extended to Markov chains with a continuum state-space. *A priori*, this appears a difficult task: we can no longer start our Markov chain from all possible initial states as there are uncountably many of them. More seriously, although the empty set $\emptyset$ can be viewed as a minimal state, there is no unique maximal state. However, many Markov chains used in this area are monotone or anti-monotone in a manner analogous to the finite case. This enables us to use the idea of sandwiching introduced and exemplified in special cases in [12, 13] ([14] describes the idea in considerable generality). The references show how to construct an observable dominating process D, which will lie above the virtual simulation from time $-\infty$. A typical dominating process D is a constant-rate birth-and-death process, where, as time passes, points are added and removed in a Markov fashion. A Markov chain with the target equilibrium distribution is derived by thinning the births of the dominating process. Births are proposed in accordance with D, but not all of them are accepted. Thus the process D represents the points that might be present in the point pattern at a given time. The rule of acceptance of births depends on the target model and is a function of the current pattern of the Markov chain.

If the acceptance rule is such that it yields a monotone Markov chain, we proceed as follows. Choose $-T_1 < 0$ and simulate a maximal and a minimal process $Y^{max(-T_1)}, Y^{min(-T_1)}$ from time $t = -T_1$ up to time 0 according to the acceptance rule. The minimal process $Y^{min(-T_1)}$ is started from the vacant pattern $\emptyset$, the maximal process $Y^{max(-T_1)}$ is started from the current point pattern of D at time $-T_1$. If the two processes coalesce during the run, we sample the point pattern at time 0. If not, we have to extend our simulation backwards, as before in the finite case, until coalescence occurs. The sample eventually obtained will have the equilibrium distribution, since all relevant realisations of the target model are sandwiched between minimal and maximal process.

If the acceptance rule for births produces an anti-monotone chain then

CFTP can still be applied with just a minor modification [12, 13]. We simulate a maximal and minimal process $Y^{max(-T_1)}$, $Y^{min(-T_1)}$ as before, only that now points are added to the minimal process $Y^{min(-T_1)}$ according to the acceptance rule applied to the current state of the maximal process $Y^{max(-T_1)}$. Analogously, births in the maximal process $Y^{max(-T_1)}$ depend on the current state of the minimal process $Y^{min(-T_1)}$. This cross-over ensures that all relevant realisations of the target process are still sandwiched between the minimal and the maximal process. Hence coalescence of these two processes ensures that equilibrium has been reached.

Further details, such as application to Metropolis-Hastings algorithms, can be found in [14]. A different algorithm for CFTP perfect simulation of some point processes is to be found in [15]. Although this algorithm is more efficient, it is less general than the above as it is developed for the special case of the penetrable spheres mixture process and associates.

Note that CFTP is not the only method to produce perfect samples. Fill [8] proposed a perfect simulation algorithm for finite state-space Markov chains based on a version of Rejection Sampling. This algorithm has been extended to the context of point processes [16], but as with [15] the extension appears to be tied to the special case of the penetrable spheres mixture process.

3 Exact Simulation of Dead Leaves

The simplest example of perfect simulation in the context of stochastic geometry is the Dead Leaves model, which was originally proposed by Matheron [17, 18]. Imagine looking down onto a piece of ground which is covered in dead leaves. The leaves on top partially hide the ones below. The resulting image is a superposition of randomly placed leaves: the *Dead Leaves process* models this kind of superposition.

Let Ξ' be a random primary grain, for example a randomly orientated leaf outline. Independent, identically distributed realisations of this grain fall randomly onto a plane at time instants given by a Poisson process of intensity $\theta\, dt$. Grains which appear between time t and time $t + dt$ cover portions of former grains generated at times $u < t$. As a grain is laid down, the boundaries of the new grain is recorded, while the boundaries of older grains which are covered by the new grain are erased. The procedure is

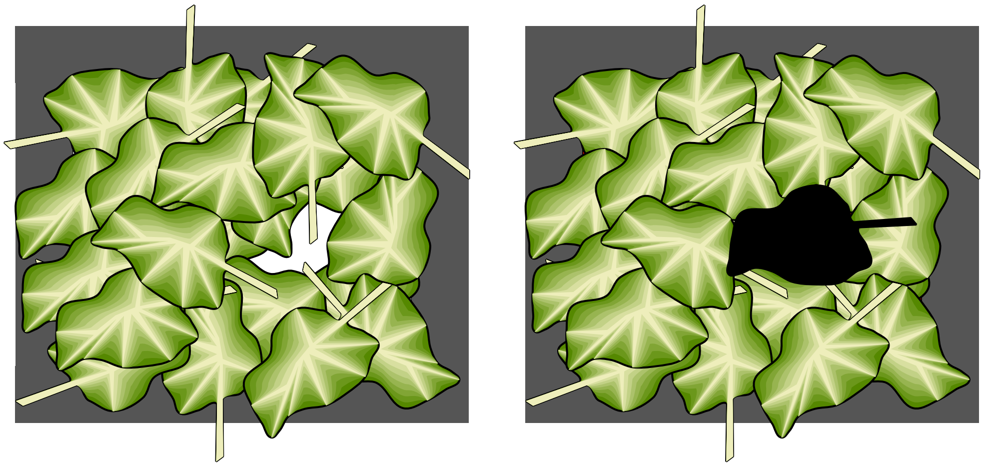

Figure 1: Simulated dead leaves model. Careful inspection shows that the image on the left still has an uncovered portion. The image on the right is obtained using one further grain, which happens to cover the previously uncovered portion. Continuing this simulation will further alter the image, as new leaves are superimposed on top of the previous pattern.

illustrated in Figure 1.

If this process is started at time $-\infty$, then routine arguments demonstrate that at time 0 the dead leaves process will be in statistical equilibrium and the plane will be completely covered, producing a random tessellation of the plane. This tessellation is a realisation of the Dead Leaves model. In his description of this model Serra [19] assumes simulation from time $-\infty$. Indeed perfect simulation for the specific case of the Dead Leaves model is well-understood by Jeulin and other workers at the Matheron-Serra group in Fontainebleu. Statistical properties, extensions and applications of this model are discussed in [20], further applications can be found in [21, 22].

How can we produce an exact sample of the Dead Leaves model? We need to simulate a Markov chain which has the Dead Leaves model as equilibrium distribution.

Algorithm 3.1 Forward Dead Leaves Algorithm
We start with a space-time Boolean model as follows. The germs are a Poisson process of space-time points in $W \times (0,\infty) \subset I\!R^2 \times (0,\infty)$, where W is a bounded window. With each space-time point (x_i, t_i) (where $0 < t_1 < t_2 < \ldots$) we associate a grain in form of a space-like fragment Ξ_i distributed as $\Xi' \oplus x_i = \{\xi + x_i : \xi \in \Xi'\}$. Consider the process $\{\Psi_t : t > 0\}$ which describes the evolution of the tessellation produced by the space-time Boolean model. At time $t = 0$ we start with the empty set, i.e. $\Psi_0 = \emptyset$. Whenever a space-time point is born, we add to Ψ_t the boundary $\partial\Xi_i$ of the associated grain and take away the parts of boundaries of previous grains covered by the new grain. This means Ψ_t changes only at time points $\{t_i\}$, with

$$\Psi_{t_i} = \partial\Xi_i \cup \left(\Psi_{t_i-} \setminus int(\Xi_i)\right), \tag{3}$$

where Ψ_{t_i-} is the configuration of Ψ immediately prior to time t_i. (Figure 1 illustrates a single step of this construction.) The large-time equilibrium distribution of $\{\Psi_t : t > 0\}$ is the Dead Leaves model.

We adopt the useful pseudo-code notation from computer science to describe the algorithm, indicating blocks by indentation. For example,

---

```
if A then
   B
else
   C
```

D

---

means, *"if condition A holds then do B else do C. After that, do D"*.

Suppose that `PoissonPointProcess`$(\theta, Leb, W \times (0,\infty))$ delivers a Poisson point process over $W \times (0,\infty)$, of intensity θ, using Lebesgue measure Leb as reference measure. Suppose further that successive calls of the procedure `Grain` return independent random sets distributed as Ξ'. Then

---

```
ForwardDeadLeaves(θ, W,Grain):
   Π ← PoissonPointProcess(θ, Leb, W × (0, ∞))
   Ψ ← ∅
   t ← 0
   repeat
      (x, t) ← first incident of Π after t
      draw Ξ from Grain
      Ξ ← Ξ ⊕ x
      Ψ ← ∂Ξ ∪ (Ψ\ int(Ξ))
      until convergence to equilibrium (!)
   return Ψ
```

How do we decide that the chain has converged to equilibrium? If we stop at a fixed time then there is a positive probability that the window W is not yet covered. Thus the sample cannot be a sample from the Dead Leaves model, which would produce complete coverage with probability one. If we instead stop at the first time that complete coverage is achieved, then the sample might be biased towards configurations which cover W efficiently. Thus the above algorithm can be implemented only in an approximate form.

Inspired by Serra's definition, we can achieve *exact* samples from the Dead Leaves model by simulating *backwards* instead of forwards in time. This can be done by using germs which form a space-time Poisson process in $W \times (-\infty, 0) \subset \mathbb{R}^2 \times (-\infty, 0)$. (Note that simulating the Dead Leaves model backwards in time has previously been considered in [20]).

Algorithm 3.2 Backward Dead Leaves Algorithm
Again we generate grains $\Xi_1, \Xi_2, \ldots$ at corresponding ordered negative *time*

points $0 > t_1 > t_2 > \ldots$. *Now each time a grain is laid down, we record the part of its boundary that is not covered by previous grains. We use the process* $\{\Psi_t : t < 0\}$ *to record the evolution of the tessellation until the whole window is covered. As before* $\Psi_0 = \emptyset$ *and changes in* Ψ_t *occur only at time points* $\{t_i\}$, *but now*

$$\Psi_{t_i} = \Psi_{t_i+} \;\cup\; \Big((\partial \Xi_i) \setminus \bigcup_{j<i} int(\Xi_j)\Big)\,, \tag{4}$$

where Ψ_{t_i+} *is the configuration of* Ψ *immediately subsequent to time* t_i. *We continue until complete coverage is first obtained, i.e.* $\bigcup_{j<i} \Xi_j = W$. *Figure 2 illustrates the construction, using the same leaves as in Figure 1, but placed in reverse order.*

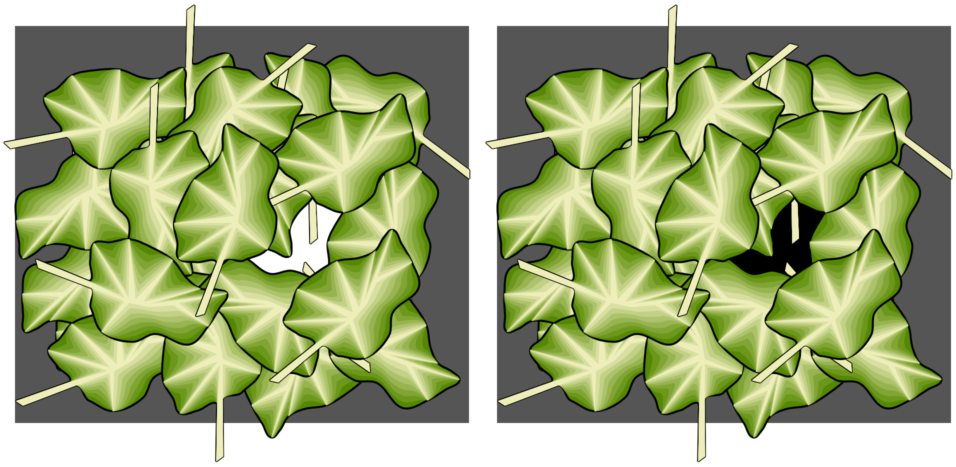

Figure 2: Perfect simulation of dead leaves model. The image on the left still has an uncovered portion. The image on the right is obtained using one further grain, which happens to cover the previously uncovered portion. In contrast to the conventional construction, however, the new leaf is added at the bottom, corresponding to simulation backwards in time and thus to perfect simulation. Clearly the backwards simulation can now be continued indefinitely without altering the resulting image, which therefore must be a sample from the equilibrium distribution.

The algorithm translates into pseudo-code notation as follows.

```
BackwardDeadLeaves(θ, W,Grain):
  Π ← PoissonPointProcess(θ, Leb, W × (−∞, 0))
  Ψ ← ∅
  Φ ← ∅
  t ← 0
  repeat
    (x, t) ← first incident of Π before t
    draw Ξ from Grain
    Ξ ← Ξ ⊕ x
    Ψ ← Ψ ∪ (∂Ξ\Φ)
    Φ ← Φ ∪ Ξ
    until W ⊆ Φ
  return Ψ
```

Clearly, we could continue the construction backwards in time indefinitely and this would still produce the same output. We therefore have a *virtual simulation from time* $-\infty$, but it actually terminates after a finite random time. It follows directly that we have a perfect sample from the equilibrium distribution, i.e. from the Dead Leaves model. Whereas backwards simulation is unbiased if we stop the algorithm at the first time of complete coverage, forward simulation will lead to a subtle bias if the simulation is run only till the first time the window is covered by grains. To illustrate this consider the following simplified example. Assume we obtain random grains $\Xi_1, \ldots, \Xi_n$ at times $t_1 > \ldots > t_n$ in the backwards simulation such that the last grain Ξ_n covers the whole window W. To allow the patterns of backward and forward simulation to agree we assume the grains of the forward simulation to be $\Xi_n, \ldots, \Xi_1$ at times $0 < -t_n + t_{n-1} < \ldots < -t_n + t_0$, where $t_0 = 0$. As Ξ_n covers all of W, complete coverage occurs in the forward simulation before time $-t_n + t_1$, in fact already at time 0. Thus we see the subtle bias mentioned above: in the case that a single grain can cover the whole window there is an increased possibility of single-grain coverage.

Note that the difference between forwards and backwards simulation here is merely one of a point of view: the conventional forwards simulation views the ground from above as it becomes covered with dead leaves. The back-

wards simulation views the same process from beneath the ground, as if one had adopted the point of view of a small animal looking upwards from a hole in the ground! This difference is illustrated in an animated simulation which can be found on website
`http://www.warwick.ac.uk/statsdept/Staff/WSK/dead.html`.

4 Conditional Boolean models

Problems arising in applied geology and mining engineering motivate the modelling of populations of (random) objects using *Boolean random sets*: a point pattern of *germs* is generated according to a Poisson point process and at each germ an object or *grain* is located which is an independent realisation of a random compact set (see [23, 24, 25]).

Often experimental data are available which provide further information on the location of grains. One may know that certain locations are covered by grains, one may know certain locations are not covered by grains, one may know the number of grains covering a certain location or one may know that certain connectivity relations hold. Simulations should incorporate this data by imposing adequate conditions on the Boolean model. Lantuéjoul [26] studies the simulation of such *conditional* Boolean models. Extending an algorithm originally proposed by Matheron, Lantuéjoul produces a spatial birth-and-death process which converges to the target conditional Boolean model exponentially fast in distribution.

But how can we obtain *exact* samples of a conditional Boolean model? We restrict our attention to Boolean models conditioned to cover a set of points $\mathcal{C} = \{z_1, \ldots, z_k\}$ in a bounded window W and show how CFTP can be applied to this setting. Following Lantuéjoul, we use a spatial birth-and-death process as an underlying process and prohibit transformations which do not respect the covering condition.

Consider first an unconditional Boolean model Ξ of germ intensity measure μ and grain distribution $\mathcal{G}$. Let H denote the space of grains with grain distribution $\mathcal{G}$. We derive a Boolean model evolving in time as follows. Consider a space-time Poisson process of mean measure $\mu(\cdot) \times \mathrm{Leb}(\cdot)$ on $W \times \mathbb{R}$. Each point of this process is marked independently with a (random) grain drawn from $\mathcal{G}$ and an Exponential variable of unit mean which represents the life-time of the grain. Thus we obtain a marked space-time Poisson process

Z on $W \times \mathbb{R} \times H \times (0, \infty)$ based on space-time points

$$(x, s, G, l) \in W \times \mathbb{R} \times H \times (0, \infty).$$

Here s represents the birth time and l represents the life time of germ-grain pair (x, G). We derive a time-evolving marked point process $\{X_t, t \in \mathbb{R}\}$, where X_t is a list of the germ-grain pairs of Z alive at time t:

$$X_t = \{(x, G) \ : \ (x, s, G, l) \in Z, \ s \leq t \leq s + l\}.$$

Notice that $\{X_t, t \in \mathbb{R}\}$ evolves as a birth-death process of objects, each object being a germ-grain pair. Also note that $\{X_t, t \in \mathbb{R}\}$ parametrizes a space-time Boolean model $\{\Xi_t, t \in \mathbb{R}\}$, where:

$$\Xi_t = \bigcup_{(x,G) \in X_t} (x \oplus G) \, .$$

The random set Ξ_t is distributed according to the above unconditional Boolean model.

How can one generate a conditional Boolean model, conditioned to cover the points in $\mathcal{C}$ with grains? We alter $X = \{X_t : t \in \mathbb{R}\}$ allowing only transformations which respect the covering condition, i.e. we do not permit any transitions which uncover a point in $\mathcal{C}$. Let $Y = \{Y_t : t \in \mathbb{R}\}$ denote this *conditional* spatial birth-and-death process. Note that births never violate the covering condition, whereas deaths might uncover a conditioning point. Consequently whenever a germ-grain pair is born in X it is also born in Y. If a germ-grain pair dies in X the germ-grain pair becomes a candidate for death in Y but in contrast to X does not necessarily die. A death is only permitted if the removal of the grain does not uncover a conditioning point.

It will be helpful to describe the simulation of such a process in rather greater detail. To each point z_i in $\mathcal{C}$ we assign a Poisson process P_i of unit mean, supplying potential death times which will be used as follows. If a germ-grain pair of Y is not allowed to die then the Poisson process assigned to the conditioning point covered by the germ-grain pair provides us with new potential death times. At the next of these death-times the germ-grain pair becomes again a candidate for death. We call a germ-grain pair which does not die at its earliest possible death-time a *perpetuated* germ-grain pair. In case of ambiguity, when the germ-grain pair covers more than one conditioning point, we need to decide which of the Poisson processes assigned to these conditioning points should provide the potential death-times for the germ-grain pair. We only consider those Poisson processes which do not yet provide death-times for a perpetuated germ-grain pair. It is arbitrary which of these processes we choose. There will always be at least one Poisson process to choose from, because a germ-grain pair will only be perpetuated if it covers at least one conditioning point which is not covered by any other germ-grain pair.

The space-time marked Poisson process X has a birth rate given by measure $\mu(\cdot)$ and a unit death rate per grain, and so Y has also birth rate $\mu(\cdot)$ but the death rate of Y varies according to coverage.

Like X the process Y parametrizes a space-time random set $\{\Theta_t, t \in \mathbb{R}\}$:

$$\Theta_t := \bigcup_{(x,G)\in Y_t} (x \oplus G).$$

We call Θ_t the induced set of Y_t. If $p(x, y)$ is the transition kernel of a Markov chain then subject to regularity conditions the distribution π is the equilibrium distribution of this chain if it satisfies the detailed balance equation:

$$\pi(x)p(x, y) = \pi(y)p(y, x).$$

As X satisfies detailed balance with respect to the unconditional Boolean model, Y also satisfies detailed balance but with respect to the conditional Boolean model. Thus subject to regularity conditions the equilibrium distribution of $Y = \{Y_t, t \in \mathbb{R}\}$ is the conditional Boolean model.

The following definitions will be useful for the description of the algorithm. Let $Y_t = \{(x_1, G_1), \ldots, (x_n, G_n)\}$, then we call the set $\bigcup_{i=1}^n (x_i \oplus G_i)$ the *induced* set of Y_t. Let $N^c(Y_t)$ denote the number of conditioning points which are covered by the induced set of Y_t. Note that we say a germ-grain pair (x, G) *covers a conditioning point* if this point is a member of the induced set $x \oplus G$. A conditioning point is *uncovered* if there is no germ-grain pair which covers this point.

The following pseudo-code (`MCMC-ConditionalBooleanModel`) describes how we can derive a simulation of Y over the time-interval $[R, S]$ from a realisation of X over the same time interval. If we choose the unit-rate Poisson process P_k to supply the potential death times of perpetuated germ-grain (x, G), then we assign index k to (x, G). This assignment is carried out by the algorithm $\texttt{index}_Y(\cdot)$. The algorithm `BooleanModel` produces a realisation of the space-time unconditional marked Poisson process X, the algorithm `Poisson` samples a unit mean Poisson process. The set I in the algorithm keeps track of the selection of Poisson processes which currently supply death times for perpetuated germ-grain pairs.

Algorithm 4.1 MCMC-ConditionalBooleanModel

```
MCMC-ConditionalBooleanModel(μ, 𝒢, 𝒞 = {z_1, ..., z_k}, W × H × [R,S]):
    X([R,S]) ← BooleanModel(μ, Leb(·), 𝒢, W × H × [R,S])
    for i = 1 to k
        P_j([R,S]) ← Poisson([R,S])
    {t_1, ..., t_m(R,S)} ← ordered list of incident times of X([R,S])
                           and P_j([R,S]) for all j = 1, ..., k
    Y ← germ-grain pairs of X at time R
    I ← ∅
    for i = 1 to m(R,S)
        if t_i birth time of (x,G) then
            Y ← Y ∪ {(x,G)}
        if t_i death time of (x,G) then
            if (N^c(Y\{(x,G)}) = N^c(Y)) then
                Y ← Y\{(x,G)}
            else
                I ← I ∪ {index_Y ((x,G))}
        if t_i incident time of P_j and j ∈ I then
            (x,G) ← grain with index j
            if (N^c(Y\{(x,G)}) = N^c(Y)) then
                Y ← Y\{(x,G)}
                I ← I\{j}
    return(Y)
```

4.1 CFTP for conditional Boolean models

How can we use CFTP to produce perfect samples of the conditional Boolean model? We define a maximal process Y^{max} and a minimal process Y^{min} which sandwich an intermediate process Y^{mid}. The minimal, the intermediate and the maximal process need to satisfy the following requirements:

1. The equilibrium distribution of the intermediate process Y^{mid} is the conditional Boolean model.

2. Given a suitable partial order $\preceq$, the intermediate process Y^{mid} continues to lie between the minimal and the maximal process if it does so initially: let $Y_t^{max(-T)}$, $Y_t^{mid(-T)}$, $Y_t^{min(-T)}$, $t \in [-T, 0]$, denote the maximal, the intermediate and the minimal process started at time $-T$; then

$$Y_t^{min(-T)} \preceq Y_t^{mid(-T)} \preceq Y_t^{max(-T)} \quad \text{for all } t \in [-T, 0] \text{ if} \qquad (5)$$
$$Y_{-T}^{min(-T)} \preceq Y_{-T}^{mid(-T)} \preceq Y_{-T}^{max(-T)}.$$

3. Let $-S < -T$, then for $t \in [-T, 0]$:

$$Y_t^{min(-T)} \preceq Y_t^{min(-S)} \preceq Y_t^{mid(-S)} \preceq Y_t^{max(-S)} \preceq Y_t^{max(-T)}. \qquad (6)$$

   (The second and the third inequality follow from (5).) The first and the last inequality ensure that the intermediate process lies between minimal and maximal process regardless of from how far in the past we simulate. Thus even a virtual simulation from time $-\infty$ is sandwiched between $Y_t^{min(-T)}$ and $Y_t^{max(-T)}$ on $[-T, 0]$.

4. If

$$Y_u^{min(-T)} = Y_u^{max(-T)} \text{ for some } u \in [-T, 0]\,,$$

   then we have *coalescence* and we require that

$$Y_t^{min(-T)} = Y_t^{max(-T)} \text{ for all } t \in [u, 0]\,.$$

   Note that if the third requirement holds then

$$Y_t^{min(-S)} = Y_t^{max(-S)} \text{ for all } t \in [u, 0] \text{ and } -S \leq -T\,.$$

5. For sufficiently large finite T (which will be random!) minimal and maximal process must coalesce if started before $-T$.

If coalescence occurs we can thereafter regard the coalesced minimal/maximal process as a virtual simulation from time $-\infty$ which is therefore in perfect equilibrium. Samples of the maximal and minimal processes form part of the figure 3.

Lemma 4.2 *Assume that the above requirements are satisfied. Set*

$$T_C = \sup\{s \leq 0 : Y_0^{max(s)} = Y_0^{min(s)}\},$$

then the distribution of

$$Y_0^{max(s)} \quad = \quad Y_0^{min(s)} \text{ for } s \leq T_C$$

is that of the conditional Boolean model.

Proof: From the fifth requirement above we know that T_C is finite. Consider a virtual simulation Y^* from time $-\infty$ defined as

$$Y_t^* = \lim_{s \to -\infty} Y_t^{min(s)}$$

(this sequence is ultimately stationary because of coalescence and the third requirement, so must converge!). Due to coalescence it follows

$$Y_t^* = \lim_{s \to -\infty} Y_t^{max(s)}.$$

By the second requirement we obtain

$$Y_t^* = \lim_{s \to -\infty} Y_t^{mid(s)}.$$

The first requirement means that $Y_0^{mid(s)}$ converges weakly to the conditional Boolean model for $s \to -\infty$. As $Y_0^{mid(s)}$ also converges weakly to Y_0^* it follows that Y_0^* is the conditional Boolean model. Now note that

$$Y_0^* = \lim_{s \to -\infty} Y_0^{min(s)} = \lim_{s \to -\infty} Y_0^{max(s)} = Y_0^{min(T_C)} = Y_0^{max(T_C)}.$$

This proves Lemma 4.2. (Note that both Lemma and proof are special cases of Theorem 1 and its proof in [14].) □

If the minimal, the intermediate and the maximal process satisfy the stated requirements, then as a consequence of Lemma 4.2 we can use CFTP to produce a perfect sample of the conditional Boolean model.

In the following we define Y^{min} and Y^{max}, then derive the intermediate process Y^{mid} and finally check the above requirements. We assume that the grain distribution $\mathcal{G}$ is such that grains have a fixed maximal size. First we need to define a partial order on our state-space $W \times H$.

Definition 4.3 Set-inclusion order
Consider two germ-grain pair lists: $S = \{(x_1, G_1), \dots, (x_n, G_n)\}$ *and* $\tilde{S} = \{(\tilde{x}_1, \tilde{G}_1), \dots, (\tilde{x}_m, \tilde{G}_m)\}$. *Let*

$$P(S) = \{j : (x_j, G_j) \in S \textit{ is a perpetuated germ-grain pair } \}$$

be the index-set of perpetuated germ-grain pairs of S. *We call*

$$S \preceq \tilde{S}$$

if there is at least one permutation $\pi = (\pi_1, \dots, \pi_m)$ *of* $(1, \dots, m)$ *such that we have*

(a) $(x_i, G_i) = (\tilde{x}_{\pi_i}, \tilde{G}_{\pi_i})$ *for all* $i \in \{1, \dots, n\} \backslash P(S)$ *and*

(a) $x_i \oplus G_i \subseteq \tilde{x}_{\pi_i} \oplus \tilde{G}_{\pi_i}$ *for all* $i \in P(S)$.

In other words $S \preceq \tilde{S}$, if each non-perpetuated germ-grain pair of S is also in $\tilde{S}$ and each perpetuated germ-grain pair of S is covered by a germ-grain pair in $\tilde{S}$. The greater the multiple coverage of conditioning points by a germ-grain list, the smaller the chance of uncovering a conditioning point and thus the greater the chance of a death. Hence the decision-rule for the acceptance of deaths is anti-monotone with respect to the set-inclusion order. To define the minimal and the maximal process we use the cross-over trick developed in [12, 13] for the case of anti-monotone point processes. Thus we make deaths of the minimal process dependent on the current state of the maximal process and vice versa.

4.2 Definition of the minimal and the maximal process

Without loss of generality we assume that the grains of the Boolean model are discs of at most unit radius. We choose a time $-T < 0$ and define the evolution of the minimal and the maximal process in $[-T, 0]$ conditional on a realisation of X and on k independent realisations of a unit mean Poisson process on the same time interval. We give a rather careful specification of the algorithms using pseudo-code, since the essence of the matter is that one must be explicit about how the resultant coupling is defined.

The algorithm `BooleanInitial`$(\mathcal{C}, X_{-T})$ sets the initial configurations of the minimal and the maximal process as follows:
The initial configuration of the minimal process is the list of germ-grain pairs describing the realisation of X at time $-T$. For the initial configuration of the maximal process we use the same list of germ-grain pairs but add an additional germ-grain pair (z, D) for each point in $z \in \mathcal{C}$, where D is a disc of radius 2. Thus this grain has a radius which is at least twice the radius of a grain in X. If the realisation of X_{-T} yields N points, whose locations are $(x_1, \ldots, x_N)$ and whose marks are $(G_1, \ldots, G_N)$, then the initial configuration of the minimal and the maximal process are given by

$$\begin{aligned} Y_{-T}^{min(-T)} &= \{(x_i,\ G_i) : i = 1, \ldots N\} \\ Y_{-T}^{max(-T)} &= Y_{-T}^{min(-T)} \cup \{(z,\ D) : z \in \mathcal{C}\}. \end{aligned}$$

Note that the initial configuration of the minimum process lies below the initial configuration of the maximum process with respect to the partial order $\preceq$.

We call the additional germ-grain pairs in the maximal process *virtual* germ-grain pairs and treat these pairs as perpetuated. For $j = 1, \ldots, k$ we assign index j to the germ-grain pair (z_j, D), where $z_j \in \mathcal{C}$. Thus the potential death-times of the perpetuated germ-grain pair (z_j, D) are provided by the j^{th} Poisson process of unit mean. The following pseudo-code describes `BooleanInitial`:

`BooleanInitial`$(\mathcal{C}, X_{-T})$:
  $y^{min} \leftarrow \{(x, G) : (x, G) \in X_{-T}\}$
  $y^{max} \leftarrow y^{min} \cup \{(z,\ D) : z \in \mathcal{C}\}$
  `for` $j = 1$ `to` $|\mathcal{C}|$

```
    index(z_j, D) ← j
I ← { index((z, D)) : z ∈ C}
return(y^min, y^max, I)
```

Notice that in fact I is always the set $\{1, \ldots, k\}$. The initial configuration of the maximal process is motivated by the following. A virtual simulation from time $-\infty$ will have germ-grain pairs which cover the conditioning points, some of which might be perpetuated. As we will show later, adding the virtual germ-grain pairs ensures that the maximal process covers the perpetuated germ-grain pairs of the virtual simulation and thus will lie *above* this simulation with respect to the set-inclusion order.

We can now give a recursive definition for the minimal and the maximal process on time interval $[-T, 0]$: Let $\mathcal{T} = \{t_1, \ldots, t_{m(T)}\}$ be the combined list of the birth-and-death times of X and the incident times of the k Poisson processes on $[-T, 0]$. Note that basic properties of Poisson point processes imply that none of the above times coincide. Thus a birth (death) time t specifies a unique germ-grain pair (x, G) with birth (death) time t. Changes in the minimal and maximal process only occur at the times given in $\mathcal{T}$. We extend the two processes in a right-continuous fashion as follows. Suppose that $Y_u^{min(-T)}$ and $Y_u^{max(-T)}$ are already defined and that t_i is the smallest element in $\mathcal{T}$ which is greater than u. We set

$$\begin{aligned} Y_{u+v}^{min(-T)} &= Y_u^{min(-T)}, \\ Y_{u+v}^{max(-T)} &= Y_u^{max(-T)} \qquad \text{for } u \leq u+v < t_i. \end{aligned}$$

(If there is no such time, we set $Y_{u+v}^{min(-T)} = Y_u^{min(-T)}$ and $Y_{u+v}^{max(-T)} = Y_u^{max(-T)}$ for all $v \geq 0$.)

The changes in the configurations of the minimal and the maximal process at time $t_i \in \mathcal{T}$ depend on whether t_i is a *birth* time of X, a *death* time of X or an *incident* time of one of the k Poisson processes.

Let t_i be a *birth* time. Then the algorithm birth(t_i, y^{min}, y^{max}) adds the unique germ-grain pair (x, G) with birth time t_i to the current list by setting

$$\begin{aligned} Y_{t_i}^{min(-T)} &= Y_{t_i-}^{min(-T)} \cup \{(x, G)\}, \\ Y_{t_i}^{max(-T)} &= Y_{t_i-}^{max(-T)} \cup \{(x, G)\}, \end{aligned}$$

where $Y^{min(-T)}_{t_i-}$ denotes the state of the minimal process immediately prior to t_i; an analogous notation is used for the maximal process.

```
birth(t_i, y^min, y^max):
   (x,G) ← germ-grain pair with birth time t_i
   y^min ← y^min ∪ {(x,G)}
   y^max ← y^max ∪ {(x,G)}
   return(y^min, y^max)
```

Note that a birth does not change the relation between the minimal and the maximal process with respect to $\preceq$. If the maximal process lies above the minimal process before a birth, it also does so afterwards.

Now let t_i be a *death* time of the unique germ-grain pair (x,G). Our action depends on the current state of the minimal and the maximal process. We remove the pair from the minimal process, if its removal from the maximal process does not uncover any conditioning points and vice versa for the maximal process. Thus we have

$$Y^{min(-T)}_{t_i} = \begin{cases} Y^{min(-T)}_{t_i-}\backslash\{(x,G)\} & \text{if } N^c\Big(Y^{max(-T)}_{t_i-}\backslash\{(x,G)\}\Big) \\ & = N^c\Big(Y^{max(-T)}_{t_i-}\Big) \\ Y^{min(-T)}_{t_i-} & \text{otherwise} \end{cases}$$

$$Y^{max(-T)}_{t_i} = \begin{cases} Y^{max(-T)}_{t_i-}\backslash\{(x,G)\} & \text{if } N^c\Big(Y^{min(-T)}_{t_i-}\backslash\{(x,G)\}\Big) \\ & = N^c\Big(Y^{min(-T)}_{t_i-}\Big) \\ Y^{max(-T)}_{t_i-} & \text{otherwise.} \end{cases}$$

The above task is performed by `death(`t_i, y^{min}, y^{max}`)` as given below, which uses a global variable `death.max` to record whether perpetuation did not take place.

```
death(t_i, y^min, y^max):
   (x,G) ← germ-grain pair with death time t_i
   death.min ← FALSE
   death.max ← FALSE
   if (N^c(y^max\{(x,G)}) = N^c(y^max)) then
      death.min ← TRUE
   if (N^c(y^min\{(x,G)}) = N^c(y^min)) then
```

```
    death.max ← TRUE
if death.min then
    y^min ← y^min\{(x, G)})
if death.max then
    y^max ← y^max\{(x, G)})
return(y^min, y^max, death.max)
```

Assume that the minimal process lies below the maximal process immediately before time t_i. Recalling the definition of the partial order $\preceq$ it is clear that if the removal of a germ-grain pair from the minimal process does not uncover any conditioning point, then neither will its removal from the maximal process. On the other hand, if the removal from the maximal process uncovers a conditioning point, then its removal from the minimal process will also uncover at least one conditioning point. From this argument it follows that if a germ-grain pair dies in the maximal process it dies in the minimal process. Similarly if a germ-grain pair is perpetuated in the minimal process, it is perpetuated in the maximal process. In any case the maximal process continues to lie above the minimal process at time t_i.

If (x, G) is not removed in any of the two processes, we assign an index to it which specifies the Poisson process which provides further potential death-times for this germ-grain pair. The algorithm `perpetuate(`t_i, y^{min}, y^{max}`)` chooses this index. For $j \in \{1, \ldots, k\}$ we call the Poisson process P_j *relevant* for (x, G), if $z_j \in \mathcal{C}$ is covered by (x, G). A Poisson process which already provides death-times for a perpetuated germ-grain pair is a *used* process. If we perpetuate germ-grain pair (x, G) in the maximal process then the index of the Poisson process chosen to do this is the smallest index of all Poisson processes which are relevant and not used in the maximal process. If the pair is perpetuated in the minimal process it is assigned the same index, thus it has the same potential death-times as in the maximal process. Now the algorithm must also account for the following scenario: a germ-grain pair (x, G) is perpetuated in the maximal process because its removal from the minimal process would uncover a conditioning point, but at the same time all conditioning points which are covered by (x, G) are already covered by perpetuated germ-grain pairs in the maximal process. This means that there are no relevant Poisson processes which are not used in the maximal process. Let I_{min} be the index set of all relevant Poisson processes for (x, G)

which are not used in the *minimal* process. In the above scenario each of the perpetuated germ-grain pairs which have an index which is in I_{min} is *enlarged by* (x, G). (We say that a germ-grain (x_1, G_1) is enlarged by a germ-grain pair (x_2, G_2) if we replace it by the germ-grain pair (x_1, G), where G is such that $x_1 \oplus G = (x_1 \oplus G_1) \cup (x_2 \oplus G_2)$. A motivation for this procedure is given when we describe the intermediate process Y^{mid}.)

Here is a summary of the algorithm `perpetuate`:

`perpetuate(`t_i, y^{min}, y^{max}, I`):`
    $I_r \leftarrow$ `index-set of relevant Poisson processes of` (x, G)
    $I_1 \leftarrow \cup\{i : i \in I_r$ `and` P_i `not used in` $y^{max}\}$
    $I_2 \leftarrow \cup\{i : i \in I_r$ `and` P_i `not used in` $y^{min}\}$
    `if` $(I_1 \neq \emptyset)$ `then`
        `index`$(x, G) \leftarrow \min\{i : i \in I_1\}$
        $I \leftarrow I \cup$ $\{$`index`$(x, G)\}$
    `else`
        $y^{max} \leftarrow y^{max} \setminus \{(x, G)\}$
        `enlarge grains in` y^{max} `of index` $\in I_2$ `by` (x, G)
    `return(`y^{min}, y^{max}, I`)`

Finally let t_i be an *incident* time of the j^{th} Poisson process. We use algorithm `perpet.death(`t_i, y^{min}, y^{max}`)` to perform the following procedure. If there is no perpetuated germ-grain pair with index j in the maximal and the minimal processes, we have no changes in the configuration of theses processes.

Suppose there is a perpetuated germ-grain pair (x, G) with index j both in the maximal process and in the minimal process. Then we delete the germ-grain pair from the maximal process if its removal from the minimal process does not uncover any conditioning point and vice versa for the minimal process. Thus we update the minimal and the maximal process as in the case when t_i is a death time.

If there is a perpetuated germ-grain pair (x, G) with index j in the maximal process which does not belong to the minimal process, then the minimal process does not change. We remove the pair from the maximal process, if adding the pair to the minimal process does not cover any additional conditioning point in the minimal process. We set

$$Y_{t_i}^{min(-T)} = Y_{t_i-}^{min(-T)}$$

$$Y_{t_i}^{max(-T)} = \begin{cases} Y_{t_i-}^{max(-T)} \backslash \{(x,G)\} & \text{if } N^c\Big(Y_{t_i-}^{min(-T)} \cup \{(x,G)\}\Big) \\ & = N^c\Big(Y_{t_i-}^{min(-T)}\Big) \\ Y_{t_i-}^{max(-T)} & \text{otherwise.} \end{cases}$$

Here is a summary:

```
perpet.death(t_i, y^min, y^max, I):
   if j ∈ I then
      if (x,G) of index j ∈ y^min then
         pdeathmax = FALSE
         pdeathmin = FALSE
         if (N^c(y^max\{(x,G)}) = N^c(y^max)) then
            pdeathmin = TRUE
         if (N^c(y^min\{(x,G)}) = N^c(y^min)) then
            pdeathmax = TRUE
         if pdeathmin then
            y^min ← y^min\{(x,G)}
         if pdeathmax then
            y^max ← y^max\{(x,G)}
            I ← I\{j}
      else
         if (N^c(y^min ∪ {(x,G)}) = N^c(y^min)) then
            y^max ← y^max\{(x,G)}
            I ← I\{j}
   return(y^min, y^max, I)
```

Combining all previous steps we obtain the algorithm `BooleanEvo`:

```
BooleanEvo(X([-T,0]); P_j([-T,0]), j∈{1,...,k}; 𝒯; 𝒞):
   (y^min, y^max, I) ← BooleanInitial(𝒞, X_{-T})
   for i = 1 to |𝒯|
      if (t_i birth-time) then
```

```
      (y^min, y^max) ← birth(t_i, y^min, y^max)
   if (t_i death-time) then
      (y^min, y^max, death.max) ← death(t_i, y^min, y^max)
      if (death.max = FALSE) then
         (y^min, y^max, I) ← perpetuate(t_i, y^min, y^max, I)
   if (t_i incident time of P_j) then
      (y^min, y^max, I) ← perpet.death(t_i, y^min, y^max, I)
return(y^min, y^max)
```

If $Y_0^{max(-T)} = Y_0^{min(-T)}$, then this configuration is a sample of the conditional Boolean model. If maximal and minimal process differ at time 0, we choose a new starting time $-S < -T$ and repeat above algorithm exchanging S for T. Note that we have to extend backwards in such a way that we reuse the same realisation of X and of the k Poisson processes on $[-T, 0]$. The algorithms which carry out the extension backwards are called `X-Extend` and `Pois-Extend`. We proceed to extend backwards in time until minimal and maximal process coalesce; if they coalesce we sample the configuration of the minimal process at time 0. The following pseudo-code provides a summary of the algorithm:

```
CFTP-ConditionalBoolean(-T, C):
   X([-T,0]) ← X-Extend(μ, -T)
   for i = 1 to k
         P_j([-T,0]) ← Pois-Extend(-T)
   𝒯 ← ordered incident times of X([-T,0]), P_j([-T,0]), j∈{1,...,k}
   (y^min, y^max) ← BooleanEvo(X([-T,0]); P_j([-T,0]), j∈{1,...,k}; 𝒯; C)
   if y^min = y^max then
      return(y^min)
   else
      CFTP-ConditionalBoolean(-2T, C)
```

4.3 Properties of the CFTP-algorithm

The following relations are invariants over the course of the algorithm:

Lemma 4.4

$$Y_t^{min(-T)} \preceq Y_t^{max(-T)} \textit{ for any } t \in [-T, 0]. \tag{7}$$

Proof: The relation (7) holds for the initial configuration of the minimal and the maximal process at time $-T$. Whenever a germ-grain pair dies in the maximal process it dies (subject to its existence) in the minimal process as long as $Y^{min} \preceq Y^{max}$ immediately before the time of death. Analogously, whenever a germ-grain pair is perpetuated in the minimal process it is also perpetuated in the maximal process. Thus the initial relation between the minimal and the maximal process is preserved. □

Lemma 4.5 *The maximal process always covers all conditioning points in $\mathcal{C}$.*

Proof: In the initial configuration of the maximal process all conditioning points are covered. Assume the removal of a germ-grain pair in the maximal process would uncover a conditioning point. Either this pair exists in the minimal process, in which case its removal from the minimal process will also uncover this conditioning point due to the previous Lemma, or it does not exist in the minimal process, in which case the conditioning point is not covered in the minimal process. In neither case do we remove the pair from the maximal process. Thus a conditioning point is never uncovered in the maximal process. □

Lemma 4.6 *The minimal process does not have any germ-grain pairs with grains which are discs of radius 2 or any germ-grain pairs which were obtained by enlarging grains.*

Proof: The initial configuration of the minimal process does not contain any such germ-grain pairs. Furthermore no such germ-grain pairs are added during the course of the algorithm. □

Lemma 4.7 *Let $-S < -T$, then for $t \in [-T, 0]$*

$$Y_t^{min(-T)} \preceq Y_t^{min(-S)} \preceq Y_t^{max(-S)} \preceq Y_t^{max(-T)}. \tag{8}$$

Proof: Note that the second inequality follows from Lemma 4.4. Recall that we derive $Y^{min(-S)}$ and $Y^{max(-S)}$ from a realisation of X on $[-S,0]$ which is obtained by extending $X([-T,0])$ backwards. Thus for $t\in[-T,0]$, all four processes $Y_t^{min(-S)}$, $Y_t^{min(-T)}$, $Y_t^{max(-S)}$ and $Y_t^{max(-T)}$ are based on the same realisation of X. We first show that the inequalities in (8) hold for $t=-T$.

What are the configurations of $Y_{-T}^{min(-S)}$ and $Y_{-T}^{max(-S)}$? Both configurations will contain all germ-grain pairs in X_{-T}. Additionally they might contain some perpetuated germ-grain pairs. In $Y_{-T}^{min(-S)}$ these perpetuated pairs are germ-grain pairs of $X([-S,-T))$. In $Y_{-T}^{max(-S)}$ the perpetuated pairs are either virtual germ-grain pairs, original germ-grain pairs of $X([-S,-T))$ or enlarged germ-grain pairs of $X([-S,-T))$. In either case any perpetuated germ-grain pair will have an induced set which lies within a disc of radius 2 centred at the appropriate conditioning point. Now let us compare these configurations with the initial configurations $Y_{-T}^{min(-T)}$ and $Y_{-T}^{max(-T)}$. The initial configuration of the minimal process $Y_{-T}^{min(-T)}$ consists only of the germ-grain pairs in X_{-T} and does not have any perpetuated germ-grain pairs. Thus we have that

$$Y_{-T}^{min(-T)}\preceq Y_{-T}^{min(-S)},$$

which is the first inequality in (8) for time $t=-T$. The initial configuration of the maximal process $Y_{-T}^{max(-T)}$ consists of the germ-grain pairs in X_{-T} plus a virtual germ-grain pair for each conditioning point. Recall that the induced set of any virtual germ-grain pair is a disc of radius 2 centred at the appropriate conditioning point for that perpetuated germ-grain pair. Thus any of the perpetuated germ-grain pairs in $Y_{-T}^{max(-S)}$ is covered by virtual germ-grain pairs of $Y_{-T}^{max(-T)}$. It follows that

$$Y_{-T}^{max(-S)}\preceq Y_{-T}^{max(-T)},$$

which is the third inequality in (8) for $t=-T$. Note that for $t\in[-T,0]$ any germ-grain pair which is born in $Y_t^{min(-S)}$ and $Y_t^{max(-S)}$ is also born in $Y_t^{min(-T)}$ and $Y_t^{max(-T)}$ and vice versa. Now consider a potential death in the time interval $[-T,0]$. Recall that the acceptance of a death for $Y_t^{max(-S)}$ depends on the current configuration $Y_{t-}^{min(-S)}$ of the minimal process immediately before time t. Suppose $Y_{t-}^{min(-S)}$ lies above $Y_{t-}^{min(-T)}$. Then if the removal of the germ-grain pair from $Y_{t-}^{min(-T)}$ does not uncover a conditioning point it also does not do so in $Y_{t-}^{min(-S)}$. On the other hand if the removal of the germ-grain pair from $Y_{t-}^{min(-S)}$ uncovers a conditioning point it also does so in $Y_{t-}^{min(-T)}$. Therefore if the germ-grain pair dies in $Y_t^{max(-T)}$, it also dies in $Y_t^{max(-S)}$, whereas if a germ-grain pair is perpetuated in $Y_t^{max(-S)}$ it is also perpetuated in $Y_t^{max(-T)}$. Now let us assume that $Y_{t-}^{max(-S)}$ lies below $Y_{t-}^{max(-T)}$. Then if we have a death in $Y_t^{min(-S)}$ we also have a death in $Y_t^{min(-T)}$ and if a germ-grain pair is perpetuated in $Y_t^{min(-T)}$ it is also perpetuated in $Y_t^{min(-S)}$. Thus the relation between the four processes is left unchanged by birth, death and perpetuation. Consequently the initial relation between the processes at time $t = -T$ is preserved. Lemma 4.7 follows. $\square$

4.4 The intermediate process

In this section we show how to construct an intermediate process Y^{mid} conditional on a realisation of the minimal and the maximal process. Then we show that this intermediate process has the same stochastic properties as the conditional spatial birth-and-death process Y defined in Algorithm 4.1. Finally we prove that the requirements stated in Section 4.1 are satisfied.

Suppose we have a realisation of $Y_t^{max(-T)}$ and $Y_t^{min(-T)}$, $t \in [-T, 0]$. The initial configuration of the intermediate process is the same as the initial configuration of the minimal process, i.e. we set

$$Y_{-T}^{mid(-T)} = Y_{-T}^{min(-T)}.$$

Note that thus $Y_{-T}^{min(-T)} \preceq Y_{-T}^{mid(-T)} \preceq Y_{-T}^{max(-T)}$.

As is the case for the minimal and the maximal process, the intermediate process only changes at time instants given by the list $\mathcal{T}$ of birth-and-death times, and incident times of the k Poisson processes, on $[-T, 0]$.

Suppose $t_i \in \mathcal{T}$ is the *birth* time of germ-grain pair (x, G). Then we add (x, G) to the intermediate process:

$$Y_{t_i}^{mid(-T)} = Y_{t_i-}^{mid(-T)} \cup \{(x, G)\}.$$

As this germ-grain pair is also added to the minimal and the maximal process, births preserve the initial relation between the three processes.

If t_i is the *death* time of the germ-grain pair (x, G), then we remove (x, G) from the intermediate process, if its removal does not uncover any conditioning point. We have

$$Y_{t_i}^{mid(-T)} = \begin{cases} Y_{t_i-}^{mid(-T)} \backslash \{(x,G)\} & \text{if } N^c\Big(Y_{t_i-}^{mid(-T)} \backslash \{(x,G)\}\Big) \\ & = N^c\Big(Y_{t_i-}^{mid(-T)}\Big) \\ Y_{t_i-}^{mid(-T)} & \text{otherwise} \end{cases}$$

Assume

$$Y_{t_i-}^{min(-T)} \preceq Y_{t_i-}^{mid(-T)} \preceq Y_{t_i-}^{max(-T)}.$$

From this it follows that if the removal of (x, G) does not uncover a conditioning point in the minimal process then it also does not do so in the intermediate process. On the other hand if the removal of (x, G) uncovers a conditioning point in the intermediate process, it also does so in the minimal process. Thus whenever (x, G) dies in the maximal process it also dies in the intermediate process; whenever it is perpetuated in the intermediate process, it is also perpetuated in the maximal process. Analogously whenever a (x, G) dies in the intermediate process it also dies in the minimal process; whenever it is perpetuated in the minimal process it is also perpetuated in the intermediate process. Hence the initial relation between the three processes is preserved.

If (x, G) is perpetuated then we need to assign an index to it which specifies the Poisson process used to supply the potential death-times. We use the following rule. If there are any relevant Poisson processes for (x, G) which are not used in the maximum process at time t_{i-}, then we assign the smallest index of these processes to (x, G). This is the same index we assign to (x, G) in the maximal process. If there are no such Poisson processes then we use the smallest index of the relevant Poisson processes which are not used in the intermediate process. (Note that if (x, G) is perpetuated, then there is at least one relevant Poisson process which is not yet used in the intermediate process.) Recall that if there is no relevant Poisson process for (x, G) which is not used in the maximal process at time t_i-, then we use (x, G) to enlarge each of a set of perpetuated germ-grain pairs in the maximal process. This set comprises the pairs which have the index of a relevant Poisson process for (x, G) which is not used in the minimal process at time t_i-. Now if $Y_{t_i-}^{min(-T)} \preceq Y_{t_i-}^{mid(-T)}$ then any Poisson process not used in the intermediate process is also not used in the minimal process.

As a result the maximal process will contain a perpetuated germ-grain pair (x', G') which covers (x, G) and has the same potential death-times.

We use the algorithm $\texttt{index}_{mid}$ to perform this task of assigning an index:

$\texttt{index}_{mid}((x,G),\ y^{mid}, y^{max}, I)$:
- $I_r \leftarrow$ `index-set of relevant Poisson processes of` (x,G)
- $I_1 \leftarrow \cup\{i : i \in I_r$ `and` P_i `not used in` $y^{max}\}$
- $I_2 \leftarrow \cup\{i : i \in I_r$ `and` P_i `not used in` $y^{mid}\}$
- `if` $(I_1 \neq \emptyset)$ `then`
  - $\texttt{index}(x,G) \leftarrow \min\{i : i \in I_1\}$
- `else`
  - $\texttt{index}(x,G) \leftarrow \min\{i : i \in I_2\}$
- $I \leftarrow I \cup \{\texttt{index}(x,G)\}$
- `return(`I`)`

Finally if t_i is an *incident* time of the j^{th} Poisson process then we perform the following procedure. If there is no perpetuated germ-grain pair with index j in the intermediate processes, we have no change in this process.

Suppose there is a perpetuated germ-grain pair (x, G) with index j in the intermediate process. Then we delete the germ-grain pair if its removal from the intermediate process does not uncover any conditioning point. We set

$$Y_{t_i}^{mid(-T)} = \begin{cases} Y_{t_i-}^{mid(-T)} \backslash \{(x,G)\} & \text{if } N^c\Big(Y_{t_i-}^{mid(-T)} \backslash \{(x,G)\}\Big) \\ & = N^c\Big(Y_{t_i-}^{mid(-T)}\Big) \\ Y_{t_i-}^{mid(-T)} & \text{otherwise} \end{cases}$$

Analogous to the case when t_i is a death time we find that if

$$Y_{t_i-}^{min(-T)} \preceq Y_{t_i-}^{mid(-T)} \preceq Y_{t_i-}^{max(-T)},$$

then

$$Y_{t_i}^{min(-T)} \preceq Y_{t_i}^{mid(-T)} \preceq Y_{t_i}^{max(-T)}.$$

The following pseudo-code summarizes how we obtain Y^{mid}:

`Intermediate(`$X_t;\ P_j(t),_{j\in\{1,\ldots,k\}};\ Y_t^{min(-T)}, Y_t^{max(-T)}, t \in [-T, 0]; \mathcal{T};\ \mathcal{C}$`)` :
- $y^{mid} \leftarrow Y_t^{min(-T)}$

```
I ← ∅
for i = 1 to |T|
   if t_i birth time then
      (x, G) ← germ-grain with birth time t_i
      y^mid ← y^mid ∪ {(x, G)}
   if t_i death-time then
      (x, G) ← germ-grain with death-time t_i
      if (N^c(y^mid) = N^c(y^mid \ {(x, G)})) then
         y^mid ← y^mid \ {(x, G)}
      else
         I ← I ∪ {index_mid((x, G), y^mid, Y_{t_i-}^{max(-T)}, I)}
   if t_i incident time of P_j then
      if (x, G) ∈ y^mid and (x, G) has index j then
         if (N^c(y^mid) = N^c(y^mid \ {(x, G)})) then
            y^mid ← y^mid \ {(x, G)}
return(y^mid)
```

Figure 3 shows a sample of the maximal, the minimal and the intermediate process at a time before the three processes have coalesced. Note how the intermediate process is sandwiched between the minimal and the maximal process.

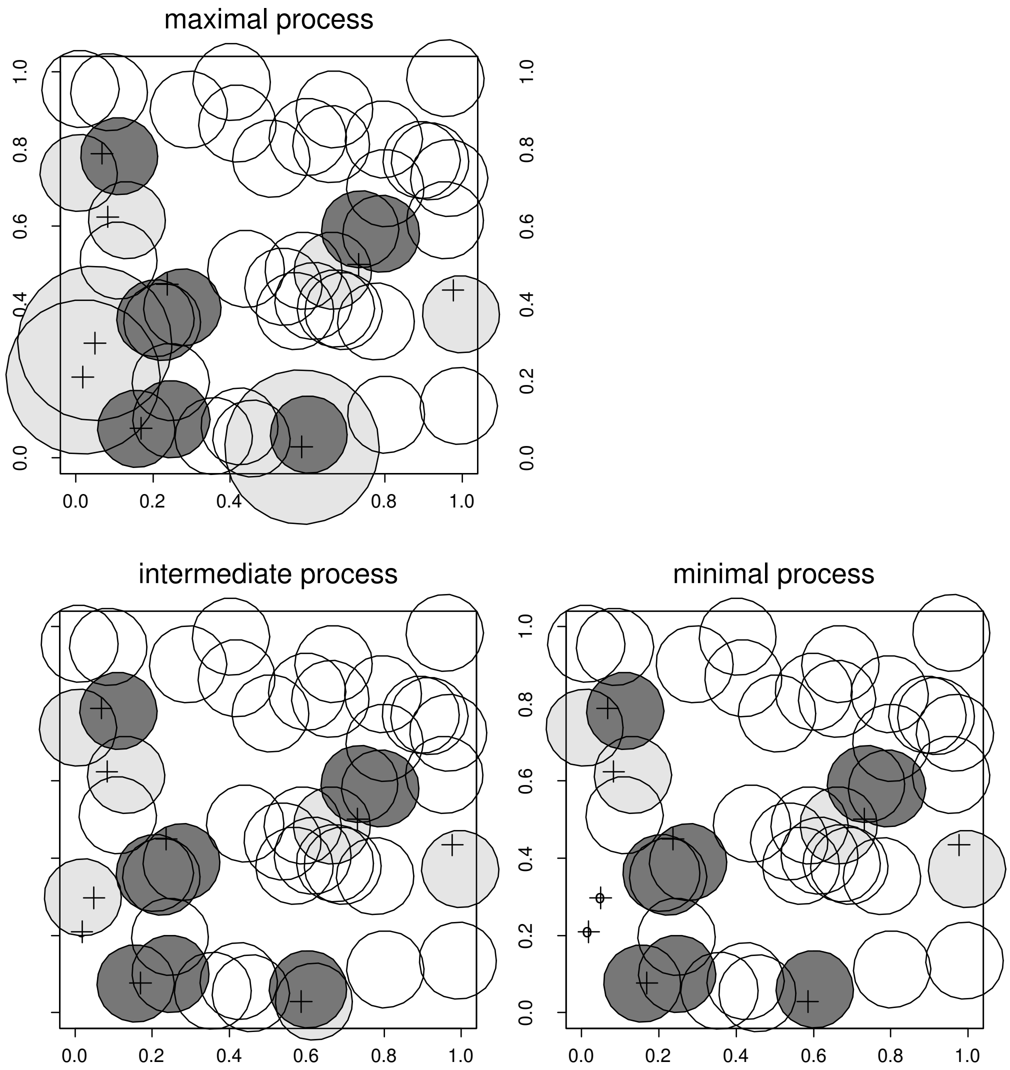

Figure 3: Illustration of maximal, minimal, and intermediate processes at a stage before coalescence has been achieved. The conditioning points are displayed as crosses. The germ intensity μ is 40 and the grains are discs of radius 0.1. The darker shaded grains are non-perpetuated grains which cover a conditioning point; the lighter shaded grains are perpetuated grains. Note that the maximal process contains virtual grains of radius 0.2.

We now show that Y^{mid} has the correct distribution when viewed on its own.

Lemma 4.8 *The process $Y_t^{mid(-T)}$, $t \in [-T, 0]$ has the same stochastic properties as the conditional spatial birth-and-death process Y in Algorithm 4.1 started at time $-T$ and run up to time 0.*

Proof: The initial configuration of Y^{mid} at time $-T$ is the list of germ-grain pairs in X_{-T}. Whenever a germ-grain pair is born in X it is also born in Y^{mid}, so Y^{mid} has the correct birth-rate. Whenever a germ-grain pair dies in X, it is a candidate for death in Y^{mid}. It is then perpetuated if and only if its removal uncovers any conditioning point, and its perpetuation uses a unit rate Poisson process to discover a sequence of further candidate death times, at each of which the germ-grain pair is perpetuated if and only if its removal uncovers any conditioning point. Thus the death rate is unit death rate per grain, except that transitions which uncover conditioning points are prohibited.

It follows that birth and death rates for Y^{mid} are the same as for the original Y discussed in Algorithm 4.1, and so the lemma is proved. □

4.5 Proofs of requirements for CFTP

We will now show that the requirements of section 3.2 are satisfied by the minimal, the intermediate and the maximal process.

Lemma 4.9 *The equilibrium distribution of the intermediate process Y^{mid} is the conditional Boolean model.*

Proof: This follows from the previous Lemma as the equilibrium distribution of Y is the conditional Boolean model. □

Lemma 4.10 *Given the partial order $\preceq$, the intermediate process Y^{mid} continues to lie between the minimal and the maximal process as long as it does initially:*

$$Y_t^{min(-T)} \preceq Y_t^{mid(-T)} \preceq Y_t^{max(-T)} \quad \text{for all } t \in [-T, 0] \text{ if } \quad Y_{-T}^{min(-T)} \preceq Y_{-T}^{mid(-T)} \preceq Y_{-T}^{max(-T)}.$$

Proof: As argued in the description of the intermediate process if the intermediate process lies between the minimal and the maximal process immediately before a birth-time, a death-time or an incident time, then it will also do so immediately after this time. Thus the initial relation is preserved.

Note that we make use of the fact that for each perpetuated germ-grain pair in the intermediate process there is a perpetuated germ-grain pair in the maximal process which covers the pair in the intermediate process and has the same potential death-times. $\square$

Lemma 4.11 *For all* $t \in [-T, 0]$*:*

$$Y_t^{min(-T)} \preceq Y_t^{min(-S)} \preceq Y_t^{mid(-S)} \preceq Y_t^{max(-S)} \preceq Y_{-T}^{max(-T)}. \tag{9}$$

Proof: This Lemma follows from Lemmas 4.10 and 4.7. $\square$

Lemma 4.12

$$Y_u^{min(-T)} = Y_u^{max(-T)} \textit{ for some } u \in [-T, 0],$$

then

$$Y_t^{min(-T)} = Y_t^{max(-T)} \textit{ for all } t \in [u, 0].$$

Proof: If the configurations of the minimal and the maximal process coincide at time u, then the acceptance rule for deaths in the minimal process is the same as in the maximal process. The above Lemma follows. $\square$

Lemma 4.13 *Minimal and maximal process coalesce in finite time.*

Proof: Let $z_1, \ldots, z_k$ be the conditioning points. Fix positive t, δ, and set

$$\begin{aligned} A_i \quad = \quad & \Big\{ \text{point } z_i \text{ is covered at time } t+\delta \text{ by a} \\ & \text{non-perpetuated grain itself born in } [t, t+\delta] \Big\} . \end{aligned}$$

The FKG inequality [27, Theorem (2.4)] applies to the events $A_1, \ldots A_k$, since adding further grains increases the family of these events which are satisfied by a realisation. Consequently we have

$$\mathbb{P}\Big(\bigcap_i A_i\Big) \quad \geq \quad \prod_i \mathbb{P}(A_i) \,.$$

On the other hand we can find $\rho > 0$ such that $\mathbb{P}(A_i) > \rho$ for all i. It follows that $\mathbb{P}\left(\bigcap_i A_i\right)$ is positive.

Now suppose that at time $t+\delta$ the conditioning points are all covered by $\ell \leq k$ non-perpetuated grains, so that $\bigcap_i A_i$ holds. By symmetry considerations, the memoryless property of the Exponential distribution, and the fact that there can be at most k perpetuated grains covering $\mathcal{C} = \{z_1, \ldots, z_k\}$, we deduce that the probability of each of the ℓ non-perpetuated grains lasting longer than all of the $r \leq k$ perpetuated grains is

$$\binom{\ell + r}{\ell}^{-1} \quad \geq \quad \binom{2k}{k}^{-1} \quad > \quad 0\,.$$

It follows that there is a positive probability that at time $t+\delta$ the conditioning points are covered in the maximal process only by non-perpetuated grains: moreover the covering event may be refined so as to be independent of events prior to time t. Therefore a Borel-Cantelli argument shows that with probability one there is sufficiently large t such that the conditioning points are covered in the maximal process only by non-perpetuated grains. But then there can be no perpetuated grains in the maximal process, and exactly then we may deduce that maximal and minimal processes coalesce. □

The above work shows that the full algorithm does indeed produce a perfect sample of the conditional Boolean model. Figure 4 shows such a perfect sample. Nine conditioning points were placed in a unit square window. The underlying germ intensity of the Boolean model is $\mu = 40$ and the grains are discs of radius 0.1. The sample was produced on a home computer (DEL 1486-DX2, 66 MHZ, 16 MB RAM) in three and a half minutes.

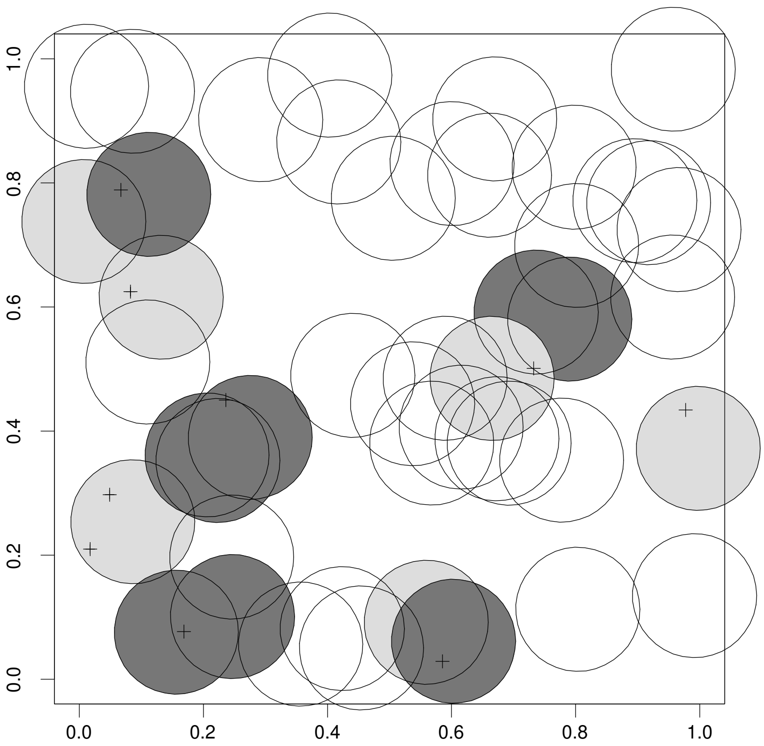

Figure 4: Perfect simulation of conditional Boolean model with constant germ intensity $\mu = 40$ and grains of radius 0.1. The conditioning points are displayed as crosses. Non-perpetuated grains which cover conditioning points are shaded dark; the shading of perpetuated grains is lighter.

5 Conclusions

In this paper we have illustrated the ideas of perfect simulation in stochastic geometry using two examples: the very simple exact simulation of the dead leaves model and the rather more complicated exact simulation of a Boolean model conditioned to cover a finite set of points. There are of course many more important variations on these ideas. Among the references we have already mentioned there are studies of perfect simulation of area-interaction point processes [15, 12, 13, 16] (of which [16] uses a totally different technique, and [12] generalizes [14] to other point processes). The techniques used above to deal with conditioned Boolean models differ yet again, in that earlier spatial birth-and-death methods used thinning of births, while here we use perpetuation of lifetimes beyond death times!

It should be noted that the method described above for perfect simulation of conditional Boolean models can be abstracted to give a method for simulation of families of correlated Poisson random variables some of which are conditioned to be positive. This and related topics will be followed up in subsequent work.

Challenges still to be met include: using the ideas of [9] to produce simulations of point processes conditioned on data observation; using perfect simulation to deal with edge-effects [28]; finding a way to do perfect simulation for conditioned Boolean models when the conditioning includes for example connectivity constraints (for applications see [26]: this is probably very hard). There is much to be done in the world of perfect simulation; however the variety of techniques already established augurs well for a bright and useful future!

References

[1] Strauss, D.J. (1975) A model for clustering, *Biometrika*, **63**, 467 - 475.

[2] Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. (1996) *Markov Chain Monte Carlo In Practice*, Chapman & Hall, London.

[3] Geyer, C.J. (1998) Likelihood Inference for Spatial Point Processes, in *Stochastic Geometry: likelihood and computation,* edited

by Barndorff-Nielsen, O.E., Kendall, W.S. and Van Lieshout, M.N.M, Chapman & Hall, London, to appear.

[4] Møller, J. (1998) Markov Chain Monte Carlo and Spatial Point Processes, in *Stochastic Geometry: likelihood and computation,* edited by Barndorff-Nielsen, O.E., Kendall, W.S. and Van Lieshout, M.N.M, Chapman & Hall, London, to appear.

[5] Cowles, M.K and Carlin, B.P. (1996) Markov chain Monte Carlo convergence diagnostics: a comparative review, *Journal of the American Statistical Association*, **91**, 883 - 904.

[6] Propp, J.G. and Wilson, D.B. (1996) Exact Sampling with Coupled Markov chains and Applications to Statistical Mechanics, *Random Structures and Algorithms* **9**, 223-252.

[7] Propp, J. G and Wilson, D.B. (1997) Coupling from the past: a user's guide, in *Microsurveys in Discrete Probability* edited by Aldous, D. and Propp, J., volume 41 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 181 - 192.

[8] Fill, J.A. (1998) An Interruptible Algorithm for Perfect Sampling via Markov Chains, *Annals of Applied Probability*, **8** 131-162.

[9] Murdoch, D. J. and Green, P. J. (1997) Exact Sampling from a Continuous State Space, *Scandinavian Journal of Statistics*, to appear.

[10] Nummelin, E. (1984) *General irreducible Markov chains and non-negative operators*, Cambridge University Press, Cambridge.

[11] Meyn, S.P. and Tweedie, R.L. (1993) *Markov chains and stochastic stability*, Springer-Verlag, New York.

[12] Kendall, W.S. (1998) Perfect Simulation for the area-interaction point process, in *Probability Perspective*, C.C. Heyde & L. Accardi, eds., World Scientific Press, Singapore, 218-234.

[13] Kendall, W.S. (1997) On some weighted Boolean Models, in *Advances in Theory and Applications of Random Sets*, D. Jeulin, ed., World Scientific Press, Singapore, 105-120.

[14] KENDALL, W.S. & MØLLER, J. (1997) Perfect Simulation of point processes using Metropolis-Hastings algorithms, in preparation.

[15] HÄGGSTRÖM, O., VAN LIESHOUT, M.N.M. & MØLLER, J. (1996) Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes, *Bernoulli*, to appear .

[16] THÖNNES, E. (1997) Perfect Simulation of some Point Processes for the Impatient User, *Advances in Applied Probability, Stochastic Geometry and Statistical Applications*, to appear.

[17] MATHERON, G. (1968) Schéma booléen séquentiel de partition aléatoire, Internal Report N-83, CMM, Fontainebleau.

[18] MATHERON, G. (1975) *Random sets and Integral Geometry*, Wiley, New York.

[19] SERRA, J. (1982) *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[20] JEULIN, D. (1997) Dead Leaves Model: From Space Tessellation to random functions, *Advances in Theory and Applications of Random Sets*, D. Jeulin, ed., World Scientific Press, Singapore, 137-156.

[21] JEULIN, D. (1993) Random models for morphological analysis of powders, *Journal of Microscopy*,**172**, 13 - 21.

[22] JEULIN, D., TEROL VILLABOS, I. AND DUBUS A. (1995) Morphological Analysis of UO_2 Powder using a Dead Leaves Model, *Microsc. Microanal. Microstruct.*,**6**, 371 - 384.

[23] CRESSIE, N. A. C. (1993) *Statistics for spatial data* Revised edition, Wiley, New York.

[24] MOLCHANOV, I. (1997) *Statistics of the Boolean Model for Practitioners and Mathematicians* J. Wiley & Sons, Chichester.

[25] STOYAN, D., KENDALL, W.S. AND MECKE, J. (1995): *Stochastic geometry and its applications*, John Wiley & Sons, Chichester, Second Edition.

[26] LANTUÉJOUL, C. (1997) Conditional Simulation Of Object-Based Models, in *Advances in Theory and Applications of Random Sets*, D. Jeulin, ed., World Scientific Press, Singapore, 271-288.

[27] GRIMMETT, G.R (1989) *Percolation* , Springer Verlag, New York.

[28] KENDALL, W.S. (1997) Perfect Simulation for Spatial Point Processes, in *Proceedings of 51st Session of ISI, Istanbul August 1997.*

About the Author — WILFRID STEPHEN KENDALL is a full professor of the Department of Statistics, University of Warwick, UK. He holds a DPhil from Oxford University (1979). He has published about 75 papers on stochastic geometry, coupling theory for random processes, computer algebra for stochastic calculus, and Brownian motion on manifolds. He is a joint author of a monograph on stochastic geometry and its applications, the coordinating editor of the stochastic geometry section of Advances in Applied Probability, and the scientific secretary of the Bernoulli Society (the world society for mathematical statistics and probability).

About the Author — ELKE THÖNNES received her MSc in Statistics in 1995 from the University College Cork. She is currently pursuing her PhD under the supervision of Wilfrid Kendall at the University of Warwick. Her research covers topics in spatial statistics and stochastic geometry. She is also interested in Markov Chain Monte Carlo algorithms, where her research has focussed on Perfect Simulation methods for point and object processes.