Perfect Simulation of some Point Processes for the Impatient User

Elke Thönnes

Department of Statistics, University of Warwick*

February 9, 1998

Abstract

Recently Propp and Wilson [14] have proposed an algorithm, called Coupling from the Past (CFTP), which allows not only an approximate but perfect (i.e. exact) simulation of the stationary distribution of certain finite state space Markov chains. Perfect Sampling using CFTP has been successfully extended to the context of point processes, amongst other authors, by Häggström et al. [5]. In [5] Gibbs sampling is applied to a bivariate point process, the penetrable spheres mixture model [19]. However, in general the running time of CFTP in terms of number of transitions is not independent of the state sampled. Thus an impatient user who aborts long runs may introduce a subtle bias, the user impatience bias. Fill [3] introduced an exact sampling algorithm for finite state space Markov chains which, in contrast to CFTP, is unbiased for user impatience. Fill's algorithm is a form of rejection sampling and similar to CFTP requires sufficient monotonicity properties of the transition kernel used. We show how Fill's version of rejection sampling can be extended to an infinite state space context to produce an exact sample of the penetrable spheres mixture process and related models. Following [5] we use Gibbs sampling and make use of the partial order of the mixture model state space. Thus

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

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

we construct an algorithm which protects against bias caused by user impatience and which delivers samples not only of the mixture model but also of the attractive area-interaction and the continuum random-cluster process.
Keywords: Markov Chain Monte Carlo, Perfect Simulation, Point Processes, Rejection Sampling, User Impatience Bias, Gibbs Sampler, Penetrable Spheres Mixture Process, Area-Interaction Process, Continuum Random-Cluster Model, Monotone Transition Rule, Time Reversal.
AMS 1991 subject classifications: Primary 60J10, Secondary 68U20, 60G57, 60D05.

1 Perfect Simulation and Coupling from the Past

Markov Chain Monte Carlo (MCMC) methods are applied when one seeks to sample from a distribution π, but elementary sampling is not feasible. Instead we construct a Markov chain whose stationary distribution is π, and simulate a long run. For an overview on MCMC techniques see e.g. Gilks et al. [4]. A common problem when using Markov Chain Monte Carlo methods is the difficulty of determining the length of simulation required to obtain a sufficiently good approximation. Bounds on the rates of convergence exist, see e.g. [11, 15, 18], but are often not precise enough to be useful in practice. Propp & Wilson [14] have recently shown how this problem can be solved for certain finite state space Markov chains. Using coupling techniques, the conventional MCMC algorithm can be extended to certify whether equilibrium has yet been reached. Thus by extending the run time if necessary, the user can ensure to obtain a *perfect* (i.e. exact) sample of the stationary distribution. This technique called Coupling from the Past (CFTP) works as follows. Suppose X is a Markov chain with transition matrix P on a finite state space $\mathcal{S}$ which converges to statistical equilibrium. We can simulate this chain using *transition rules*:

Definition 1.1 *A transition rule for a Markov transition matrix P on the state space $\mathcal{S}$ is a measurable function $f : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ together with a random* *variable U on the state space $\mathcal{U}$ such that*

$$\mathbb{P}\Big(f(x,U) = \cdot\Big) = P(x,\cdot).$$

If $f(\cdot,U)$ is a transition rule for P then we can use a sequence $\{U_n\}, n \in \mathbb{Z}$, of independent copies of the random variable U to simulate the chain X via

$$X_k = f(X_{k-1}, U_k).$$

Now, if it was possible to start X at time $-\infty$ then at time 0 the chain would be in exact equilibrium. CFTP is an algorithm which samples such an infinitely long simulation by reconstructing it over a finite time in the recent past. We choose a time $-T_1 < 0$ and obtain a realization $\{u_n\}_{n=-T_1+1}^{0}$ of the sequence $\{U_n\}$ for $n \in \{-T_1+1, -T_1+2, \ldots, 0\}$. Using this realisation we start the chain at time $t = -T_1$ from every state in S and run it up to time 0. As we use the same realisation $\{u_n\}_{n=-T_1+1}^{0}$ for all initial states, the paths of the chain will coalesce if they meet. Suppose all paths have coalesced into a single path by time 0 and thus sample the same state x at time 0. Then assuming that a simulation from time $-\infty$ would use the same sequence $\{u_n\}_{n=-T_1+1}^{0}$ to specify its transitions from time $-T_1$ to 0, this simulation would also sample x at time 0. Thus we can regard this sample as the result of a *virtual simulation from time* $-\infty$ which is in perfect equilibrium.

If the paths have not coalesced into one path by time 0, we need to observe more of the past to reconstruct the infinite-length simulation. We choose a time $-T_2 < -T_1$ and extend backwards as follows. We sample a sequence $\{u_n\}_{n=-T_2+1}^{-T_1}$ of independent copies of U and combine it with the previous sequence $\{u_n\}_{n=-T_1+1}^{0}$, thus obtaining a realisation of $\{U_n\}_{n=-T_2+1}^{0}$. We then proceed as before, iteratively venturing further into the past until we achieve complete coalescence.

Propp and Wilson [14] show that a near optimum choice for the iterative starting times $-T_1 > -T_2 > \ldots$ is

$$-T_k = -2^{k-1},\ k \in \mathbf{N} \cup \{0\}.$$

Thus in the kth iteration of CFTP we simulate a run of length 2^{k-1}.

Whether we can use CFTP in practice mainly depends on whether we can check for coalescence without having to examine the resulting paths for all of the initial states of $\mathcal{S}$. Suppose $\mathcal{S}$ has a partial order $\leq_{\mathcal{S}}$, a maximal state $\hat{1}$ and a minimal state $\hat{0}$ such that

$$\hat{0} \ \leq_{\mathcal{S}} \ x \leq_{\mathcal{S}} \ \hat{1} \quad \text{for all } x \in \mathcal{S}.$$

Then detecting coalescence can be relatively easy if the Markov chain is *monotone*.

Definition 1.2 *A Markov chain X on $\mathcal{S}$ is monotone with respect to $\leq_{\mathcal{S}}$, if its transition matrix P maintains the partial order, i.e. if for $x, y \in \mathcal{S}$*

$$P(x,\cdot) \overset{\mathcal{D}}{\leq} P(y,\cdot) \quad \textit{whenever} \quad x \ \leq_{\mathcal{S}} \ y,$$

where $\overset{\mathcal{D}}{\leq}$ denotes the stochastic domination with respect to $\leq_{\mathcal{S}}$.

See Lindvall [10] for further comments on stochastic domination and monotonicity. X will be monotone if there exists a *monotone* transition rule for its transition matrix P:

Definition 1.3 *Let the function $f(\cdot, U)$ together with the random variable U be a transition rule for P. Then $f(\cdot, U)$ is monotone if for all realisations u of U*

$$f(x,u) \ \leq_{\mathcal{S}} \ f(y,u) \quad \textit{whenever} \ \ x \ \leq_{\mathcal{S}} \ y.$$

Assume the chain is monotone. Then if the paths started in state $\hat{0}$ and in state $\hat{1}$ have coalesced, the paths of all intermediate states must have also coalesced. Therefore we only need to monitor the two paths started in state $\hat{0}$ and in state $\hat{1}$.

CFTP has been applied to a variety of areas, a comprehensive overview is given in [13]. Häggström et al. [5], Kendall [7] and Kendall & Møller [8] have extended CFTP to the context of point processes. In [7] spatial birth/death processes were used allowing the exact simulation of several point processes, in particular the attractive and repulsive area interaction process [1, 19]. In [5] CFTP was used for a Gibbs sampler producing exact samples of a bivariate point process, the penetrable spheres mixture model [19], and related models, in particular the attractive (but not repulsive) area-interaction. As will be shown in [8] CFTP can also be applied to Metropolis-Hastings algorithms.

2 User Impatience Bias

Fill [3] points out that a drawback of CFTP is the fact that the running time in terms of Markov transitions and the state sampled by the algorithm are not independent. Hence an impatient user who aborts long runs may introduce a subtle bias. As an example consider the Markov chain X on the state space $\{0,1,2\}$ with transition matrix

$$P_X = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}.$$

The stationary distribution of X is given by $\pi_0 = 2/5$, $\pi_1 = 1/5$ and $\pi_2 = 2/5$. We can simulate X using the following transition rule:
for $x = 0, 2$

$$f(x,U) = \begin{cases} 0 & \text{if } U \leq 1/2 \\ \min\{x+1, 2\} & \text{if } U > 1/2 \end{cases}$$

for $x = 1$

$$f(1,U) = 2,$$

where U is a uniform random variable on $[0,1]$. Note that this transition rule is not monotone, so we need to start our simulation from all three initial states and check the resulting paths for coalescence.

Now suppose the user terminates a run of CFTP whenever I iterations have been completed without producing output. Conditional on obtaining output the user will not sample output $Z \in \{0,1,2\}$ according to the equilibrium distribution as the following calculations show. Recall that the length of the run in the i^{th} iteration of CFTP is equal to 2^{i-1}. In this example coalescence cannot occur in the first iteration. After the first iteration the possible states are $\{0,2\}$ or $\{1,2\}$. Moreover until coalescence occurs the set of possible states alternates between $\{0,2\}$ and $\{1,2\}$. In either case the probability of coalescence in the next step is $1/2$. Further elementary calculations show that conditional on coalescence within the first I iterations the user will sample Z with the following probabilities:

$$\mathbb{P}\Big(Z = r \,\Big|\, I \,\Big) \;=\; \frac{\mathbb{P}\Big(\text{coalescence within } I \text{ iterations and } Z = \; r \Big)}{\mathbb{P}\Big(\text{coalescence within } I \text{ iterations} \Big)}.$$

Let $N = 2^{I-1}$, then N is the number of transitions in the longest run and we have

$$\mathbb{P}\Big(Z = r \Bigm| I \Big) = \frac{\sum_{k=2}^{N} (\frac{1}{2})^{k-1} \Big[\frac{1}{2} P^{(N-k)}(0,r) + \frac{1}{2} P^{(N-k)}(2,r) \Big]}{1 - (\frac{1}{2})^{N-1}},$$

where $P^k(s,r)$ is the probability of going from state s to state r in k steps. For $I = 2$, we get:

$$\mathbb{P}\Big(Z = 0 \Bigm| I = 2\Big) = 1/2 \qquad \mathbb{P}\Big(Z = 2 \Bigm| I = 2\Big) = 1/2.$$

For $I \geq 3$, we obtain

$$\begin{aligned}
\mathbb{P}\Big(Z = 0 \Bigm| I\Big) &= \frac{2}{5}\Big[\frac{1 - 2^{-N}}{1 - 2^{-N+1}}\Big] \\
\mathbb{P}\Big(Z = 1 \Bigm| I\Big) &= \frac{1}{5}\Big[\frac{1 - 2^{-N}}{1 - 2^{-N+1}}\Big] \\
\mathbb{P}\Big(Z = 2 \Bigm| I\Big) &= \frac{2}{5}\Big[\frac{1 - 7 \times 2^{-N-1}}{1 - 2^{-N+1}}\Big].
\end{aligned}$$

Although the bias reduces with increasing N and thus with increasing maximum number of iterations I, the sample will always be biased.
Not all Markov chains suffer bias from user impatience. Here is an example of a Markov chain for which protection against this kind of bias is automatic. Consider a Markov chain Y on $\{0,1\}$ with transition matrix

$$P_Y = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{pmatrix}.$$

This Markov chain, which was also considered in [3] and [7], has equilibrium distribution $(\pi_0, \pi_1) = (2/3, 1/3)$. The following function is a transition rule for Y:

$$f(0,U) = \begin{cases} 0 & \text{if } U \leq 1/2 \\ 1 & \text{if } U > 1/2 \end{cases}$$

$$f(1,U) = 0,$$

where U is a uniform variable on the unit interval.

If the user allows for more than one iteration of the usual CFTP algorithm we have no user impatience bias as the following calculations show. Let I be the maximum number of iterations the user wants to use and $Z \in \{0, 1\}$ the output of the CFTP algorithm. Then for $N = 2^{I-1}$

$$\begin{aligned}
\mathbb{P}\Big(Z = 0 \,\Big|\, I\Big) &= \frac{\mathbb{P}(\text{ coalescence within } I \text{ iterations and Z } = 0)}{\mathbb{P}(\text{ coalescence within } I \text{ iterations })} \\
&= \frac{\sum_{k=1}^{N} \; P^{(N-k)}(0,0) \times (\frac{1}{2})^k}{1 - (\frac{1}{2})^N} \\
&= \frac{\sum_{k=1}^{N} \; \Big(\frac{2}{3} - \frac{2}{3}(-\frac{1}{2})^{N-k+1}\Big) \times (\frac{1}{2})^k}{1 - (\frac{1}{2})^N} = \frac{2}{3}.
\end{aligned}$$

For more examples illustrating the bias caused by user impatience, bounds on the magnitude of this bias and further comments see sections 5.2 and 6.1 in [3].

3 Fill's Interruptible Algorithm

Fill's algorithm does not use CFTP but is based on a different idea, namely rejection sampling, which protects against user impatience bias. The setting for this algorithm is as follows. We have a Markov chain X on a partially ordered finite state space $(\mathcal{S}, \leq_{\mathcal{S}})$ which converges to the equilibrium distribution π. The state space has a unique maximal element $\hat{1}$ and a unique minimal element $\hat{0}$. The transition matrix P of X is such that its *time reversal* $\tilde{P}$ defined by

$$\tilde{P}(x, y) = \frac{\pi(y)P(y, x)}{\pi(x)} \quad \text{for } x \text{ such that } \pi(x) > 0$$

is *monotone* with respect to $\leq_{\mathcal{S}}$. As $\tilde{P}$ is monotone, for each $x \in \mathcal{S}$ and $y \geq_{\mathcal{S}} x$ there exists an *upward kernel* $K_{(x,y)}(\cdot, \cdot)$ such that

$$\tilde{P}(y, y') = \sum_{x' \in \mathcal{S}} \tilde{P}(x, x')K_{(x,y)}(x', y') \quad \text{for all } y' \in \mathcal{S}.$$

If there exists a monotone transition rule $\tilde{f}(\cdot, U)$ for $\tilde{P}$ then we can use

$$K_{(x,y)}(x', y') = \mathbb{P}\Big(\tilde{f}(y, U) = y' \mid \tilde{f}(x, U) = x'\Big)$$

as an upward kernel. We assume that we can simulate transitions from the measure $K_{(x,y)}(x', \cdot)$ whenever $\tilde{P}(x, x') > 0$ and $y \geq_{\mathcal{S}} x$. Let $P^{(N)} = P^N$ denote the N-step transition matrix of X and $\tilde{P}^{(N)}$ the time reversed N-step transition matrix.

Like any rejection sampling procedure Fill's algorithm proposes a sample which is then accepted or rejected with an appropriate probability. The interruptible algorithm proposes an observation z drawn from $P^{(N)}(\hat{0}, \cdot)$ as a sample of $\pi(\cdot)$ and accepts it with probability

$$\frac{1}{c} \times \frac{\pi(z)}{P^{(N)}(\hat{0}, z)},$$

where $c = \dfrac{\pi(\hat{0})}{\tilde{P}^{(N)}(\hat{1}, \hat{0})}$ is an upper bound on the ratio $\dfrac{\pi(z)}{P^{(N)}(\hat{0}, z)}$.

The choice of c is adequate because according to the definition of $\tilde{P}$ we have that

$$\frac{\pi(z)}{P^{(N)}(\hat{0}, z)} = \frac{\pi(\hat{0})}{\tilde{P}^{(N)}(z, \hat{0})},$$

and due to the monotonicity of $\tilde{P}$

$$\frac{\pi(\hat{0})}{\tilde{P}^{(N)}(z, \hat{0})} \leq \frac{\pi(\hat{0})}{\tilde{P}^{(N)}(\hat{1}, \hat{0})}.$$

The rejection sampling procedure consists of three steps, where the first step samples from $P^{(N)}(\hat{0}, \cdot)$ and the second and third step are designed such that this sample is accepted with the appropriate probability:

(1) Start X in $\hat{0}$ and simulate it for N steps. Record the obtained trajectory $(X_0 = \hat{0}, \ldots, X_N = z)$. The state of X_N is the proposed sample state.

(2) Reverse the obtained trajectory in time leading to the trajectory

$$(\tilde{X}_0 = z, \tilde{X}_1, \ldots, \tilde{X}_N = \hat{0}) = (X_N = z, X_{N-1}, \ldots, X_0 = \hat{0}).$$

This time-reversed trajectory is regarded as a $\tilde{P}$-trajectory conditioned to start in z and to end in $\hat{0}$.

(3) A second Markov chain $\tilde{Y}$ is simulated for N steps using the upward kernels $K_{(x,y)}(\cdot,\cdot)$ together with the time-reversed trajectory. The initial state of $\tilde{Y}$ is set as $\tilde{Y}_0 = \hat{0}$. Then $\tilde{Y}_k$ for $k \in \{1, \ldots, N\}$ is simulated according to the kernel

$$K_{(\tilde{X}_{k-1}, \tilde{Y}_{k-1})}(\tilde{X}_k, \ \cdot).$$

If $\tilde{Y}_N = \hat{0}$ the proposed sample z is accepted, else it is rejected.

Fill [3] shows that because upward kernels of $\tilde{P}$ are used to produce $\tilde{Y}$, the probability of $\tilde{Y}_N = \hat{0}$ is equal to

$$\frac{1}{c} \times \frac{\pi(z)}{P^{(N)}(\hat{0}, z)},$$

whence the rejection sampling procedure is carried out correctly.

If a proposed state is not accepted, the number of transitions N is doubled and the above steps are repeated *independently* from the previous run. We adopt the useful pseudocode notation from computer science to describe the algorithm. Here the algorithm `Markov1`(x) draws a sample from $P(x, \cdot)$ and `Markov2`(y, x, x') draws a sample from $K_{(x,y)}(x', \cdot)$.

RejectionSampling(N):
  $x_0 \leftarrow \hat{0}$
  $\tilde{y}_0 \leftarrow \hat{1}$
  for k = 1 to N
    $x_k \leftarrow$ Markov1(x_{k-1})
  for k = 0 to N
    $\tilde{x}_k \leftarrow x_{N-k}$
  for k = 1 to N
    $\tilde{y}_k \leftarrow$ Markov2($\tilde{y}_{k-1}, \tilde{x}_{k-1}, \tilde{x}_k$)
  if ($\tilde{y}_N = \hat{0}$)
    return(x_N)
  else
    RejectionSampling(2N)

An essential requirement for the applicability of Fill's algorithm is that the time reversal of the transition kernel has appropriate stochastic monotonicity properties. As shown by Häggström et al. [5] Gibbs sampling applied to the penetrable spheres mixture model is monotone with respect to a partial order defined in [5]. This enables us to extend Fill's algorithm to produce an exact sample of the mixture model. The problem setting differs from Fill's in two respects. Firstly, in contrast to the Markov chains studied by Fill, the state space of the point processes considered is not finite but uncountable. Secondly and more importantly, although the state space is partially ordered, it does not have a unique maximal or minimal element.

We first introduce the penetrable spheres mixture model and related processes. Then we describe the three steps of our algorithm, i.e. Proposal algorithm, time reversal and Acceptance algorithm. In section 8 we combine these steps into a Rejection Sampling algorithm and explain why this algorithm provides us with an exact sample of the penetrable spheres mixture model and its relatives. The following section describes some simulations we performed, and the last section summarizes our conclusions.

4 The penetrable spheres mixture model

Let $W \subseteq \mathbb{R}^d$ be a bounded Borel set and let $\mathcal{S}$ denote the space of all finite point configurations in W without multiple points. The penetrable spheres mixture model, see [6, 16, 17, 19], is a point process with two types of points, X and Y, on the product state space $\mathcal{S}_2 = \mathcal{S} \times \mathcal{S}$. The joint density of (X, Y) with respect to the product measure of two independent unit rate Poisson processes on W is given by

$$f(x, y) = \alpha_{\text{mix}}\, \beta_1^{n(x)} \beta_2^{n(y)}\, 1_{\{d(x,y)>R\}}, \tag{1}$$

where x and y are point configurations and $n(x)$ is the number of points in x. The factor α_{mix} is the normalising constant, $\beta_1, \beta_2 > 0$ are model parameters and $d(x, y)$ denotes the shortest distance between a point in x and a point in y. In the mixture model points of different type interact with each other via the constraint that points of different type must never be as close as R. The conditional distribution of X given Y is a homogeneous Poisson process of intensity of β_1 on $W \setminus (Y \oplus G)$. Here G is a sphere of radius R and $\oplus$ denotes the Minkowski addition, i.e

$$Y \oplus G = \{y + g \ : \ y \in Y, g \in G\}.$$

Similarly, Y given X is a Poisson process of intensity β_2 on $W\backslash(X \oplus G)$. The mixture model is related to two other point processes, the area-interaction process and the continuum random-cluster model.
The area-interaction point process, studied in Baddeley and Van Lieshout [1], has the following density with respect to a unit-rate Poisson point process restricted to W:

$$\pi(x) = \alpha_1 \lambda_1^{n(x)} \gamma_1^{-m_d(x \oplus G)}. \tag{2}$$

The factor α_1 is the normalising constant, $\lambda_1, \gamma_1 > 0$ are model parameters and m_d denotes the Lebesgue measure in $\mathbb{R}^d$. If $\gamma_1 > 1$ then this process exhibits attractive interaction between the points, for $\gamma_1 < 1$ we have repulsive interaction. We assume that the grain of the process G is a sphere of radius R. Consider again the penetrable spheres mixture model. The marginal distribution of X is an area-interaction model with interaction parameters $\lambda_1 = \beta_1$ and $\gamma_1 = e^{\beta_2}$, see [5, 19]. The same result holds for the marginal distribution of Y with the rôles of β_1 and β_2 exchanged.
The continuum random-cluster model, see e.g. [2, 9, 12], is specified by its density

$$p(x) = \alpha_2 \lambda_2^{n(x)} \gamma_2^{-c(x)} \tag{3}$$

with respect to a unit-rate Poisson process on W. Here α_2 is the normalizing constant and $\lambda_2, \gamma_2 > 0$ are model parameters. We use $c(x)$ to denote the number of connected components of $x \oplus H$, where H is a sphere of radius $R/2$. Its relationship to the penetrable spheres mixture model is as follows. Consider the superposition of the two types of points $Z = X \cup Y$ in the symmetric mixture model, i.e. with $\beta_1 = \beta_2$. As shown in [5] this superposition is a continuum random-cluster model with radii $R/2$ and parameters $\lambda_2 = \beta_1 = \beta_2$ and $\gamma_2 = 1/2$.
We mentioned earlier that the conditional distributions of X and Y are Poisson processes, which are easy to sample. Hence Gibbs sampling, which is an MCMC method which samples conditional distributions, allows for an easy simulation of the mixture model, see [5, 12]. Given a sample of the mixture model, a sample of an attractive area-interaction model can be obtained by marginalising whereas a continuum random-cluster model with parameter $\gamma = 1/2$ can be obtained through superpositioning.

5 The Proposal algorithm

In the first step of the Rejection sampling algorithm we simulate a Markov chain, whose stationary distribution is the target distribution. Gibbs sampling is an easy way to obtain such a Markov chain for the penetrable spheres mixture model. We use the same Gibbs Sampler as [5]. Let (x_0, y_0) be an arbitrary point pattern in $\mathcal{S}_2$, then a single step of the Gibbs sampler is:

GibbsSampling1((x_0, y_0)):
  $y_1 \sim$ Poisson(β_2) on $W \backslash (x_0 \oplus G)$
  $x_1 \sim$ Poisson(β_1) on $W \backslash (y_1 \oplus G)$
  return $((x_1, y_1))$

Note that the Gibbs sampler does not actually use y_0 at all, it is included in the function argument of GibbsSampling1 for the sake of completeness. The equilibrium distribution of the resulting Markov chain does not depend on the order in which we update, hence we could also use:

GibbsSampling2((x_0, y_0)):
  $x_1 \sim$ Poisson(β_1) on $W \backslash (y_0 \oplus G)$
  $y_1 \sim$ Poisson(β_2) on $W \backslash (x_1 \oplus G)$
  return $((x_1, y_1))$

Here it is x_0 which is not being used.
In the Rejection Sampling algorithm we will iterate GibbsSampling1 a fixed number of times and use the result as the proposal sample. Within one iteration of the Gibbs sampler we have two transition steps, i.e. the actual sequence of point patterns is as follows:

$$(x_0, y_0) \rightarrow (x_0, y_1) \rightarrow (x_1, y_1).$$

Let $p_x(x'|y)$ be the density of a Poisson process of intensity β_1 on $W \setminus [y \oplus G]$. Then $p_x(x' \mid y)$ is the Radon-Nikodým density with respect to a unit-rate Poisson process on W given by

$$p_x(x' \mid y) = \alpha \beta_1^{n(x')} 1_{[x' \in W \setminus [\, y \oplus G\,]]}$$

where α is the normalising constant and $n(x')$ is the number of points in x'. Let $p_y(y'|x)$ be the density of a Poisson process of intensity β_2 on $W \setminus [x \oplus G]$.

Then the first transition is made according to the kernel $p_{y_0}(y|\, x_0)$ and the second transition is made according to $p_{x_0}(x|\, y_1)$. Detailed balance holds, i.e.

$$f(x,y)p_y(y'|x) = f(x,y')p_{y'}(y|x) \text{ and}$$

$$f(x,y)p_x(x'|y) = f(x',y)p_{x'}(x|y).$$

As pointed out earlier, Fill's idea requires that the transition kernels used have appropriate stochastic monotonicity properties. Consider the following partial order on $\mathcal{S}_2$ introduced in [5]:

Definition 5.1 *We call*

$$(x,y) \preceq (x',y') \qquad \textit{if} \qquad x \subseteq x' \textit{ and } y \supseteq y'.$$

Let P be a transition kernel on the partially ordered state space $(\mathcal{S}_2, \preceq)$. Analogous to the case of a finite state space we call P *monotone*, if

$$P\Big((x,y),\cdot\Big) \overset{\mathcal{D}}{\preceq} P\Big((x',y'),\cdot\Big) \quad \text{whenever } (x,y) \preceq (x',y'),$$

where $\overset{\mathcal{D}}{\preceq}$ denotes the stochastic domination with respect to $\preceq$. The transition kernel P will be monotone if there exists a *monotone transition rule*, that is a measurable function $f : \mathcal{S}_2 \times \mathcal{U} \to \mathcal{S}_2$, where $\mathcal{U}$ is the state space of some random variable U, such that

$$f\Big(\,(x,y),U\Big) \overset{\mathcal{D}}{\sim} P\Big(\,(x,y),\cdot\,\Big) \quad \text{for all } (x,y) \in \mathcal{S}_2 \text{ and}$$

$$f\Big(\,(x,y),u\,\Big) \preceq f\Big(\,(x',y'),u\,\Big) \quad \text{for all } u \in \mathcal{U} \text{ whenever } (x,y) \preceq (x',y').$$

Both `GibbsSampling1` and `GibbsSampling2` are monotone with respect to the partial order in Definition 5.1 as we can define monotone transition rules. Consider $f_1 : \mathcal{S}_2 \times \mathcal{S} \to \mathcal{S}_2$ and $f_2 : \mathcal{S}_2 \times \mathcal{S} \to \mathcal{S}_2$ such that:

$$f_1((x,y),z_1) = (z_1 \setminus [y \oplus G],\ y),$$

$$f_2((x,y),z_2) = (x,\ z_2 \setminus [x \oplus G]),$$

where z_j is a realisation of a Poisson point process on W with intensity $\beta_j, j = 1,2$. These functions specify monotone transition rules for our Gibbs Sampler since

$$f_1(\,(x,y),\ Z_1) \overset{\mathcal{D}}{\sim} p_x(x'|y)$$

$$f_2(\ (x,y),\ Z_2)\ \overset{\mathcal{D}}{\sim}\ p_y(y'|x).$$

and also

$$f_j((x,y),z_j) \preceq f_j((x',y'),z_j) \ \text{ whenever } \ (x,y) \preceq (x',y'). \tag{4}$$

In the following we use the above transition rules whenever we perform `GibbsSampling1` or `GibbsSampling2`.

In contrast to Fill's setting, our Markov chain is defined on an uncountable state space which, although it has a partial order, does not have a unique maximal or minimal state. Instead [5] calls an element (x,y) *quasimaximal*, if the induced set of the first type of points $x \oplus G$ covers the whole window and the induced set of the second type of points $y \oplus G$ is the empty set, that is

$$x \oplus G \supseteq W \quad \text{and} \quad y = \emptyset.$$

Similarly, the element (x,y) is *quasiminimal*, if

$$x = \emptyset \quad \text{and} \quad y \oplus G \supseteq W.$$

Although these states are not unique maximal and minimal states, they serve as effectively unique maximal and minimal states in the following sense:

Lemma 5.2 *Let (x',y') be an arbitrary element of $\mathcal{S}_2$, let (x^0,y^0) be a quasiminimal element and let (x^1,y^1) be a quasimaximal element. Suppose we perform n* `GibbsSampling1` *iterations on all three states. We couple the transitions of the chains by successively applying $f_2(\cdot,Z_2)$ and $f_1(\cdot,Z_1)$ to all three chains. Let (x'_n,y'_n) describe the state of the chain started in (x',y') after n iterations of* `GibbsSampling1`, *(x^0_n,y^0_n) the state of the chain started in (x^0,y^0) and (x^1_n,y^1_n) the state of the chain started in (x^1,y^1). Then*

$$(x^0_n,y^0_n) \preceq (x'_n,y'_n) \preceq (x^1_n,y^1_n) \ \text{ for all } n \geq 1. \tag{5}$$

Note that the same result holds for `GibbsSampling2`.

In other words if we successively use $f_2(\cdot,Z_2)$ and $f_1(\cdot,Z_1)$ to couple the transitions, the path started in a quasiminimal state will lie *below* the path started in (x',y') (with respect to $\preceq$). Similarly the path started in a quasimaximal state will lie *above* the path started in (x',y').
Proof: We will proof the first inequality in (5) for the case of `GibbsSampling1`.

The second inequality follows analogously. We perform one iteration of `GibbsSampling1` by applying successively the transition rules $f_2(\cdot, z_2)$ and $f_1(\cdot, z_1)$ to both states (x^0, y^0) and (x', y'). As (x^0, y^0) is quasiminimal we know that $x^0 = \emptyset$. Thus $(x_1^0, y_1^0) = (z_1 \setminus [z_2 \oplus G], z_2)$. From state (x', y') we reach the state $(z_1 \setminus [z_2' \oplus G],\ z_2')$, where $z_2' = z_2 \setminus [x' \oplus G]$. Clearly

$$(z_1 \setminus [z_2 \oplus G], z_2) \preceq (z_1 \setminus [z_2' \oplus G],\ z_2'),$$

whence the first inequality in (5) holds for $n = 1$. Furthermore as our transition rules are monotone, the first inequality in (5) also holds for $n > 1$. Thus the path started in the quasiminimal state will always lie *below* the path started in (x', y') (with respect to $\preceq$).

The result can be proved analogously for the case of `GibbsSampling2`. Note that this Lemma is equivalent to Lemma 4 in [5].

6 The time reversal step

In the first step of our algorithm we run N iterations of the Gibbs sampler and obtain a realisation of the sequence $(X_k, Y_k)_{k \in \{0,\ldots N\}}$. We then reverse the obtained sequence of point patterns in time, i.e. we set

$$\tilde{X}_k = X_{N-k} \quad \text{and} \quad \tilde{Y}_k = Y_{N-k}.$$

In this section we will prove a result about the distribution of this time reversed sequence which will be essential when showing that the developed algorithm samples the correct target distribution. Fill uses an analogous result but without proof, see section 7.3 in [3]. We include a proof, as the result is unfamiliar in the context of point processes, though it is very familiar to workers in Markov chain theory. Let $f(x, y)$ denote the density of the mixture model given in (1).

Definition 6.1 *Let $\tilde{p}_x(x'|y)$ and $\tilde{p}_y(y'|x)$ be the time reversal of the transition kernels $p_x(x'|y)$ and $p_y(y'|x)$, i.e.*

$$\tilde{p}_x(x'|y) = \frac{f(x', y) p_{x'}(x|y)}{f(x, y)} \quad \textit{and} \quad \tilde{p}_y(y'|x) = \frac{f(x, y') p_{y'}(y|x)}{f(x, y)} \tag{6}$$

for arbitrary x, y such that $f(x, y) > 0$.

Note that an easy calculation shows that $\tilde{p}_x(x'|y) = p_x(x'|y)$ and $\tilde{p}_y(y'|x) = p_y(y'|x)$: in this case the Markov chain is *reversible*, which is not generally required for Fill's method. However it should be noticed that despite the Markov chain reversibility the two-step construction using `GibbsSampling1` is not reversible: `GibbsSampling1` and `GibbsSampling2` stand in duality.

Lemma 6.2 *Let the time-reversed N-step transition density be given by*

$$\tilde{p}^N\Big((\tilde{x}_N, \tilde{y}_N) \;\Big|\; (\tilde{x}_0, \tilde{y}_0)\Big) = \int \tilde{p}_{\tilde{x}_0}(\tilde{x}_1|\tilde{y}_0)\, \tilde{p}_{\tilde{y}_0}(\tilde{y}_1|\tilde{x}_1) \times \cdots \times \tilde{p}_{\tilde{x}_{N-1}}(\tilde{x}_N|\tilde{y}_{N-1}) \\ \times \tilde{p}_{\tilde{y}_{N-1}}(\tilde{y}_N|\tilde{x}_N) d(\tilde{x}_1, \tilde{y}_1) \cdots d(\tilde{x}_{N-1}, \tilde{y}_{N-1}). \quad (7)$$

Then, conditional on starting point $(\tilde{X}_0, \tilde{Y}_0) = (q, r)$ *and end point* $(\tilde{X}_N, \tilde{Y}_N) = (s, t)$, *the time-reversed sequence* $(\tilde{X}_k, \tilde{Y}_k)_{k\in\{0,\dots N\}}$ *has the density*

$$\Big[\tilde{p}^N\Big((s,t) \;\Big|\; (q,r)\Big)\Big]^{-1} \tilde{p}_q(\tilde{x}_1|r)\tilde{p}_r(\tilde{y}_1|\tilde{x}_1)\tilde{p}_{\tilde{x}_1}(\tilde{x}_2|\tilde{y}_1)\cdots \tilde{p}_{\tilde{x}_{N-1}}(s|\tilde{y}_{N-1})\tilde{p}_{\tilde{y}_{N-1}}(t|s). \quad (8)$$

In other words, the time reversed sequence is distributed according to the time reversal kernels starting with an update of the first type of points if we condition on the state of the sequence at time 0 and time N. Note that the time-reversed N-step transition density specifies the distribution of the output produced by N iterations of `GibbsSampling2`.

Proof: Consider the time-inhomogenous latent Markov chain $(R_n, S_n)_0^{2N}$, which describes the sequence of forward transitions in detail, i.e. for $k \in \{0, \dots, N\}$

$$(R_{2k}, S_{2k}) = (X_k, Y_k) \quad \text{and} \quad (R_{2k+1}, S_{2k+1}) = (X_k, Y_{k+1}).$$

Suppose $(\tilde{R}_n, \tilde{S}_n)_0^{2N}$ is the time reversal of $(R_n, S_n)_0^{2N}$, i.e.

$$(\tilde{R}_n, \tilde{S}_n) = (R_{2N-n}, S_{2N-n}).$$

Let $h_j(\cdot,\cdot)$ be the density of (R_j, S_j). The time reversed sequence $(\tilde{R}_n, \tilde{S}_n)_0^{2N}$ has the same density as the original sequence, which is

$$\begin{aligned} & h^N\Big((\tilde{x}_n, \tilde{y}_n)_0^{2N}\Big) \\ &= h_0(x_0, y_0)p_{y_0}(y_1|x_0)p_{x_0}(x_1|y_1)\cdots p_{y_{N-1}}(y_N|x_{N-1})p_{x_{N-1}}(x_N|y_N). \end{aligned}$$

Substituting the time reversal kernels for the transition kernels this density becomes

$$\begin{aligned} h^N\Big((\tilde{x}_n,\tilde{y}_n)_0^{2N}\Big) &= \frac{h_0(x_0,y_0)f(x_N,y_N)}{f(x_0,y_0)}\,\tilde{p}_{y_1}(y_0|x_0)\tilde{p}_{x_1}(x_0|y_1)\tilde{p}_{y_2}(y_1|x_1) \\ &\qquad\qquad \times\cdots\times\tilde{p}_{y_N}(y_{N-1}|x_{N-1})\tilde{p}_{x_N}(x_{N-1}|y_N). \end{aligned}$$

Substituting $\tilde{x}_k$ and $\tilde{y}_k$ for x_{N-k} and y_{N-k} and reordering leads to

$$\begin{aligned} h^N\Big((\tilde{x}_n,\tilde{y}_n)_0^{2N}\Big) &= \frac{h_0(\tilde{x}_N,\tilde{y}_N)f(\tilde{x}_0,\tilde{y}_0)}{f(\tilde{x}_N,\tilde{y}_N)}\,\tilde{p}_{\tilde{x}_0}(\tilde{x}_1|\tilde{y}_0)\tilde{p}_{\tilde{y}_0}(\tilde{y}_1|\tilde{x}_1)\tilde{p}_{\tilde{x}_1}(\tilde{x}_2|\tilde{y}_1) \\ &\qquad\qquad \times\cdots\times\tilde{p}_{\tilde{x}_{N-1}}(\tilde{x}_N|\tilde{y}_{N-1})\tilde{p}_{\tilde{y}_{N-1}}(\tilde{y}_N|\tilde{x}_N). \end{aligned}$$

If we now condition on $(\tilde{X}_0,\tilde{Y}_0)=(q,r)$ and $(\tilde{X}_N,\tilde{Y}_N)=(s,t)$, the conditional density of the time reversed sequence is

$$\begin{aligned} & h^N\Big((\tilde{x}_n,\tilde{y}_n)_0^{2N}\;\Big|\;(\tilde{X}_0,\tilde{Y}_0)=(q,r),(\tilde{X}_N,\tilde{Y}_N)=(s,t)\Big) \\ &\quad = \tilde{p}^N\Big((s,t)\;\Big|\;(q,r)\Big)^{-1}\,\tilde{p}_q(\tilde{x}_1|r)\tilde{p}_r(\tilde{y}_1|\tilde{x}_1)\cdots\tilde{p}_{\tilde{x}_{N-1}}(s|\tilde{y}_{N-1})\tilde{p}_{\tilde{y}_{N-1}}(t|s). \quad (9) \end{aligned}$$

□

We make some further statements, which we will need when we prove that our algorithm produces exact samples of the mixture model.
Let $p^N\Big((x_N,y_N)\;\Big|\;(x_0,y_0)\Big)$ be the N-step transition density for `Gibbs-Sampling1`, i.e.

$$\begin{aligned} p^N\Big((x_N,y_N)\;\Big|\;(x_0,y_0)\Big) = \int & p_{y_0}(y_1|x_0)\,p_{x_0}(x_1|y_1)\times\cdots\times p_{y_{N-1}}(y_N|x_{N-1}) \\ & \times p_{x_{N-1}}(x_N|y_N)d(x_1,y_1)\cdots d(x_{N-1},y_{N-1}). \end{aligned}$$

Due to detailed balance we have that

$$f(x_0,y_0)\,p^N\Big((x_N,y_N)\;\Big|\;(x_0,y_0)\Big) = f(x_N,y_N)\,\tilde{p}^N\Big((x_0,y_0)\Big|(x_N,y_N)\Big). \quad (10)$$

Now let $z_{\hat{0}}$ be a quasiminimal pattern and $z_{\hat{1}}$ be a quasimaximal pattern, then due to Lemma 5.2

$$\begin{aligned} p^N\Big(z_{\hat{1}}\Big|z_{\hat{0}}\Big) &\le p^N\Big(z_{\hat{1}}\Big|(x_0,y_0)\Big) \quad \text{for any } (x_0,y_0)\in\mathcal{S}_2 \text{ and} \\ \tilde{p}^N\Big(z_{\hat{0}}\Big|z_{\hat{1}}\Big) &\le \tilde{p}^N\Big(z_{\hat{0}}\Big|(\tilde{x}_0,\tilde{y}_0)\Big) \quad \text{for any } (\tilde{x}_0,\tilde{y}_0)\in\mathcal{S}_2. \end{aligned} \quad (11)$$

7 The Acceptance algorithm

After obtaining the time reversed trajectory of our Markov chain, we simulate a second Markov chain, whose final state will be used as criterion for accepting or rejecting the proposed sample. The evolution of this second chain depends on the evolution of the time reversed trajectory as follows. Suppose we have the following transitions in the time reversed trajectory

$$(\tilde{x}_0, \tilde{y}_0) \to (\tilde{x}_1, \tilde{y}_0) \to (\tilde{x}_1, \tilde{y}_1).$$

Let $(\tilde{v}_0, \tilde{w}_0)$ be the starting pattern of our second Markov chain, such that $\tilde{w}_0 \subseteq \tilde{y}_0$. One iteration of the Acceptance algorithm is then defined as follows:

`Acceptance(`$(\tilde{v}_0, \tilde{w}_0), (\tilde{x}_0, \tilde{y}_0), (\tilde{x}_1, \tilde{y}_1)$ `):`
  $\tilde{r}_1 \sim$ `Poisson(`β_1`) on` $\tilde{y}_0 \oplus G$
  $\tilde{v}_1 \leftarrow [\ \tilde{x}_1 \cup \tilde{r}_1\ ] \setminus [\ \tilde{w}_0 \oplus G\ ]$
  $\tilde{w}_1 \leftarrow \tilde{y}_1 \setminus [\ \tilde{v}_1 \oplus G\ ]$
  `return` $(\tilde{v}_1, \tilde{w}_1)$

Note that this algorithm does not make use of the point pattern $\tilde{v}_0$. Also note that $\tilde{w}_1 \subseteq \tilde{y}_1$, so that we can apply the above algorithm iteratively.

We designed the Acceptance algorithm such that the first transition is made according to the conditional distribution of $f_1((\tilde{v}_0, \tilde{w}_0), Z_1)$ given $f_1((\tilde{x}_0, \tilde{y}_0), Z_1) = (\tilde{x}_1, \tilde{y}_0)$ and the second according to the conditional distribution of $f_2((\tilde{v}_1, \tilde{w}_0), Z_2)$ given $f_2((\tilde{x}_1, \tilde{y}_0), Z_2) = (\tilde{x}_1, \tilde{y}_1)$. Hence the transitions of our second chain are made according to upward kernels of the transition kernels $\tilde{p}_x(x'|y)$ and $\tilde{p}_y(y'|x)$; see section 3 in this paper and section 7.1 in Fill [3] for further comments on upward kernels.

The Acceptance algorithm was chosen such that the Rejection Sampling algorithm has the appropriate acceptance probability. To show this we will need the following result about the distribution of the output produced by N iterations of `Acceptance`. Suppose the sequence $(\tilde{x}_0, \tilde{y}_0)_0^N$ was produced by running N iterations of `GibbsSampling1` and reversing the obtained trajectory. Let s be the state at time 0 and t be the state at time N of this time-reversed sequence. Using this sequence we run N iterations of the Acceptance algorithm and obtain output $(\tilde{V}_N, \tilde{W}_N)$. The distribution of this output is given as follows, in a simple consequence of Lemma 6.2 above.

Corollary 7.1 *Let $(\tilde{v}_0, \tilde{w}_0) = (r_1, r_2)$ be the starting pattern for the Acceptance algorithm and let $(\tilde{V}_N, \tilde{W}_N)$ be the output produced by N iterations of* `Acceptance`. *Conditional on $(\tilde{x}_0, \tilde{y}_0) = s$ and $(\tilde{x}_N, \tilde{y}_N) = t$, $(\tilde{V}_N, \tilde{W}_N)$ is distributed according to*

$$\frac{\tilde{p}^N\Big((\tilde{v}_N, \tilde{w}_N)\;\Big|\;(r_1, r_2)\Big)}{\tilde{p}^N\Big(t\;\Big|\;s\Big)}$$

Proof: Assume that the transitions in the time reversed sequence $(\tilde{x}_0, \tilde{y}_0)_0^N$ were made according to the time reversal kernels $\tilde{p}_x(x'|y)$ and $\tilde{p}_y(y'|x)$. As the transition kernels coincide with their time reversals, we have

$$\tilde{x}_{k+1} \overset{\mathcal{D}}{\sim} \text{ Poisson}(\beta_1) \text{ on } W \setminus [\;\tilde{y}_k \oplus G\;] \text{ and}$$

$$\tilde{y}_{k+1} \overset{\mathcal{D}}{\sim} \text{ Poisson}(\beta_2) \text{ on } W \setminus [\;\tilde{x}_{k+1} \oplus G\;].$$

The Acceptance algorithm chooses $\tilde{r}_{k+1}$ such that

$$\tilde{r}_{k+1} \overset{\mathcal{D}}{\sim} \text{ Poisson}(\beta_1) \text{ on } \tilde{y}_k \oplus G.$$

Due to the independence properties of the Poissson point process it follows that

$$\tilde{v}_{k+1} \sim \text{ Poisson}(\beta_1) \text{ on } W\setminus [\;\tilde{w}_k \oplus G\;].$$

As $\tilde{w}_k \subseteq \tilde{y}_k$, we have $\tilde{x}_{k+1} \subseteq \tilde{v}_{k+1}$. Thus

$$\tilde{y}_{k+1}\setminus\tilde{v}_{k+1} \;\;=\;\; \tilde{w}_{k+1} \overset{\mathcal{D}}{\sim} \text{ Poisson}(\beta_2) \text{ on } W \setminus [\;\tilde{v}_{k+1} \oplus G\;].$$

In other words, the transitions $(\tilde{v}_k, \tilde{w}_k) \to (\tilde{v}_{k+1}, \tilde{w}_k)$ are distributed according to $\tilde{p}_{\tilde{v}_k}(\tilde{v}|\tilde{w}_k)$ and the transitions $(\tilde{v}_{k+1}, \tilde{w}_k) \to (\tilde{v}_{k+1}, \tilde{w}_{k+1})$ according to $\tilde{p}_{\tilde{w}_k}(\tilde{w}|\tilde{v}_{k+1})$.

As shown in Lemma 6.2, conditional on $(\tilde{x}_0, \tilde{y}_0) = s$ and $(\tilde{x}_N, \tilde{y}_N) = t$ the transitions of the time reversed sequence are made according to the time reversal kernels, thus our initial assumption holds. Hence conditional on s and t the N-step transition kernel of the Acceptance algorithm is given by

$$\frac{\tilde{p}^N\Big((\tilde{v}_N, \tilde{w}_N)\;\Big|\;r\Big)}{\tilde{p}^N\Big(t\;\Big|\;s\Big)},$$

where the denominator is due to our condition.

□

8 The Rejection Sampling algorithm

Following Fill's program we combine the algorithms of section 5-7 to a Rejection Sampling algorithm as follows. We first determine a quasiminimal pattern $(z_1, z_2) = z_{\hat{0}}$, i.e. we set $z_1 = \emptyset$ and choose z_2 deterministically such that $z_2 \oplus G \supseteq W$. We will use $z_{\hat{0}}$ as the starting pattern for the Proposal algorithm. We then set $z_{\hat{1}} = (z_2, z_1)$, which leads to a quasimaximal pattern. This will be the starting pattern for the Acceptance algorithm. Throughout the following we keep the above patterns fixed. Note that the Rejection Sampling algorithm will not actually depend on the choice of z_2 as the Proposal algorithm only makes use of the first component of the starting pattern and the Acceptance algorithm only of the second one, see the remarks made when describing the algorithms. The Rejection Sampling algorithm is then defined as follows:

```
RejectionSampling(N):
   (x_0, y_0) ← z_0̂
   (ṽ_0, w̃_0) ← z_1̂
   for k = 1 to N
      (x_k, y_k) ← GibbsSampling1(x_{k-1}, y_{k-1})
   for k = 0 to N
      (x̃_k, ỹ_k) ← (x_{N-k}, y_{N-k})
   for k = 1 to N
      (ṽ_k, w̃_k) ← Acceptance((ṽ_{k-1}, w̃_{k-1}), (x̃_{k-1}, ỹ_{k-1}), (x̃_k, ỹ_k))
   if ((ṽ_N, w̃_N) = z_0̂)
      return((x_N, y_N))
   else
      RejectionSampling(2N)
```

Note that if a sample is not accepted, the following run of the Markov chain is independent of the previous runs.

If we use conventional MCMC, we just iterate `GibbsSampling1` (say) N times and assume that the produced sample has the stationary distribution. The reversal and acceptance step in our algorithm ensures that the sample actually has the stationary distribution. How does this work?

This section follows closely Fill's argument in section 7.3 of [3]. The algorithm is a form of rejection sampling: the Proposal algorithm proposes a sample and the Acceptance algorithm accepts it with an appropriate probability. The Proposal algorithm generates a sample according to the N-step transition density $p^N(\cdot \mid z_{\hat{0}})$ Having obtained a sample (x, y) from $p^N(\cdot \mid z_{\hat{0}})$, we wish to accept it as a sample from $f(\cdot, \cdot)$ with probability $f(x,y) \setminus (c\, p^N((x,y) \mid z_{\hat{0}}))$, where c is an upper bound on the ratio $f(x,y) \setminus p^N((x,y) \mid z_{\hat{0}})$. Due to detailed balance (10) and the monotonicity of the Gibbs sampler (11)

$$\frac{f(x,y)}{p^N((x,y)|z_{\hat{0}})} = \frac{f(z_{\hat{0}})}{\tilde{p}^N(z_{\hat{0}}|(x,y))} \leq \frac{f(z_{\hat{0}})}{\tilde{p}^N(z_{\hat{0}}|z_{\hat{1}})}.$$

So we can use $c = f(z_{\hat{0}}) \setminus \tilde{p}^N(z_{\hat{0}}|z_{\hat{1}})$ and hence we want to accept (x, y) with probability

$$\frac{f(x,y)}{p^N((x,y)|z_{\hat{0}})} \times \frac{1}{c} = \frac{f(z_{\hat{0}})}{\tilde{p}^N(z_{\hat{0}}|(x,y))} \times \frac{\tilde{p}^N(z_{\hat{0}}|z_{\hat{1}})}{f(z_{\hat{0}})} = \frac{\tilde{p}^N(z_{\hat{0}}|z_{\hat{1}})}{\tilde{p}^N(z_{\hat{0}}|(x,y))}.$$

But the Acceptance algorithm accepts the sample exactly with this probability as the following Lemma shows:

Lemma 8.1 *The probability that the Acceptance algorithm accepts a proposed sample* (x, y)

$$I\!P\Big((\tilde{v}_N, \tilde{w}_N) = z_{\hat{0}} \;\Big|\; (\tilde{x}_0, \tilde{y}_0) = (x,y),\ (\tilde{x}_N, \tilde{y}_N) = z_{\hat{0}},\ (v_0, w_0) = z_{\hat{1}}\Big)$$

is equal to

$$\frac{\tilde{p}^N(z_{\hat{0}}|z_{\hat{1}})}{\tilde{p}^N(z_{\hat{0}}|(x,y))}.$$

Proof: This follows directly from Corollary 7.1 using $(\tilde{x}_0, \tilde{y}_0) = (x, y)$, $(\tilde{x}_n, \tilde{y}_n) = z_{\hat{0}}$, $(\tilde{v}_0, \tilde{w}_0) = z_{\hat{1}}$ and $(\tilde{v}_N, \tilde{w}_N) = z_{\hat{0}}$.

□

Lemma 8.2 *The (random) number of iterations I performed by the Rejection Sampling algorithm until producing output is finite with probability one.*

Proof: The unconditional probability of acceptance is

$$c^{-1} = \frac{\tilde{p}^N(z_{\hat{0}}|z_{\hat{1}})}{f(z_{\hat{0}})} = \frac{p^N(z_{\hat{1}}|z_{\hat{0}})}{f(z_{\hat{1}})}.$$

By ergodicity the above expression converges to one as $N \to \infty$. Therefore the probability of output at the j^{th} iteration given no prior output converges to one for $j \to \infty$.

□

9 Simulation

We implemented the Rejection Sampling algorithm in a C program to demonstrate that the exact sampling of the penetrable spheres mixture model with this algorithm is feasible in practice. Figure 1 shows some samples of the penetrable spheres mixture model obtained through the new algorithm. We use the same display as Häggström et al. in [5]. The first type of points is pictured as a dot, the second type is pictured as a cross. The circles are of radius $R/2$ and show the area around a point in which no point of the other type is allowed. Note that for a larger value of β phase transitions seem to begin to occur and one of the components dominates. The implementation of our algorithm also enables us to compare it to the analogous algorithm using CFTP in [5] with respect to memory and expected number of transitions needed.

Fill [3] shows that for a finite Markov Chain a naive implementation of the algorithm needs more memory than the analogous algorithm using CFTP. It is clear that similarly our algorithm will also need more memory than the CFTP algorithm in [5].

Fill [3] also derives some properties of the number of iterations used by the algorithm in terms of the separation. The proofs also hold for an uncountable state space with a fixed quasimaximal and quasiminimal state, we therefore just summarize the results given in section 8 of [3].

Definition 9.1 *Let $z_{\hat{1}}$ and $z_{\hat{0}}$ be a fixed quasimaximal respectively a fixed quasiminimal state. We define the separation at time N of the Markov chain obtained through* `GibbsSampling1` *started in $z_{\hat{0}}$ as*

$$\mathrm{sep}_{z_{\hat{0}}}^{z_{\hat{1}}}(N) := 1 - \frac{p^N(\ z_{\hat{1}} \mid z_{\hat{0}}\ )}{f(z_{\hat{1}})},$$

where $f(\cdot)$ is the density of the penetrable spheres mixture model.

The above separation coincides with the unconditional probability of rejection. Note that due to detailed balance (10) and the monotonicity of the Gibbs Sampler (11) we have

$$\mathrm{sep}_{z_{\hat{0}}}^{z_{\hat{1}}}(N) = \max_{y \in \mathcal{S}_2} \Big\{ 1 - \frac{p^N(y \mid z_{\hat{0}})}{f(y)} \Big\}.$$

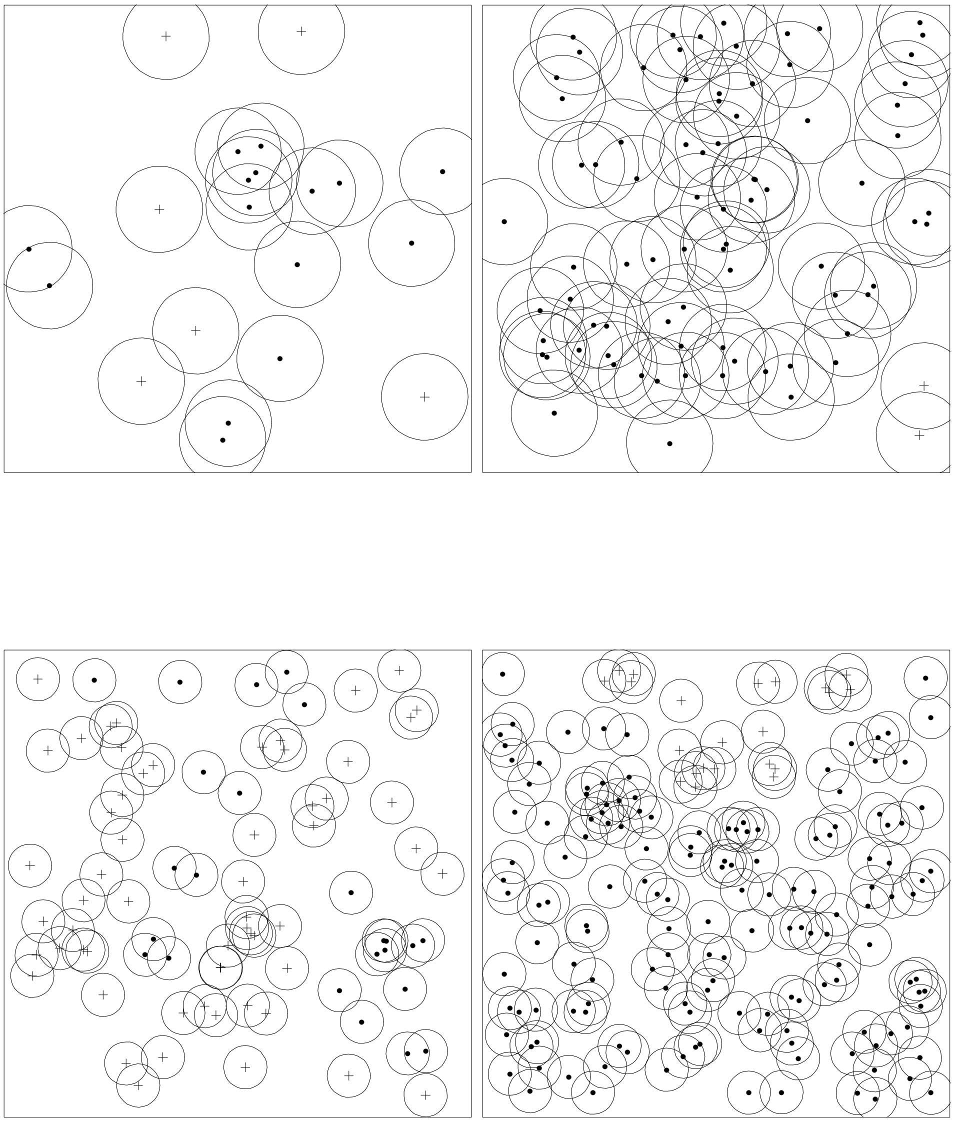

Figure 1: Realisation of the symmetric mixture model with $\beta = 30, R = 0.2$ (top left), $\beta = 80, R = 0.2$ (top right), $\beta = 100, R = 0.1$ (bottom left), $\beta = 200, R = 0.1$ (bottom right).

Proposition 9.2 *Let I denote the number of iterations used by the algorithm (i.e. the number of runs needed to produce output) and F be the number of forward transitions generated by the algorithm, i.e.*

$$F = \sum_{i=0}^{I-1} 2^i = 2^I - 1.$$

Then

$$E(F) = \sum_{i=0}^{\infty} 2^i \prod_{j=0}^{i-1} sep_{z_{\hat{0}}}^{z_{\hat{1}}}(2^j) \quad and$$

$$P(F > 2^i - 1) = \prod_{j=0}^{i-1} sep_{z_{\hat{0}}}^{z_{\hat{1}}}(2^j),$$

In the case of a reversible finite state space Markov chain, Fill derives an upper bound on the expected number of Markov transitions, which is smaller than the bound on the expected number of transitions given by Propp and Wilson [14] for CFTP. Both bounds rely strongly on the assumption of having a finite state space. At present there are no bounds on the number of expected transitions for the algorithm in [5]. Thus in order to compare the number of iterations I needed for our algorithm with the number of iterations needed for CFTP we performed some simulation experiments. As with CFTP we started each simulation with one iteration of the Gibbs Sampler and increased the number of Gibbs Sampler iterations by a factor of two if the proposed sample was not accepted. Figure 2 shows the Monte Carlo estimates of the expected number of iterations I of the Rejection Sampling alogrithm. To enable the comparison with the Monte Carlo estimates in [5] we plotted the average number of iterations for a symmetric penetrable spheres mixture model versus the parameter $\theta = \log(\beta_1) = \log(\beta_2)$. We found that Rejection Sampling used about the same number of iterations as CFTP.

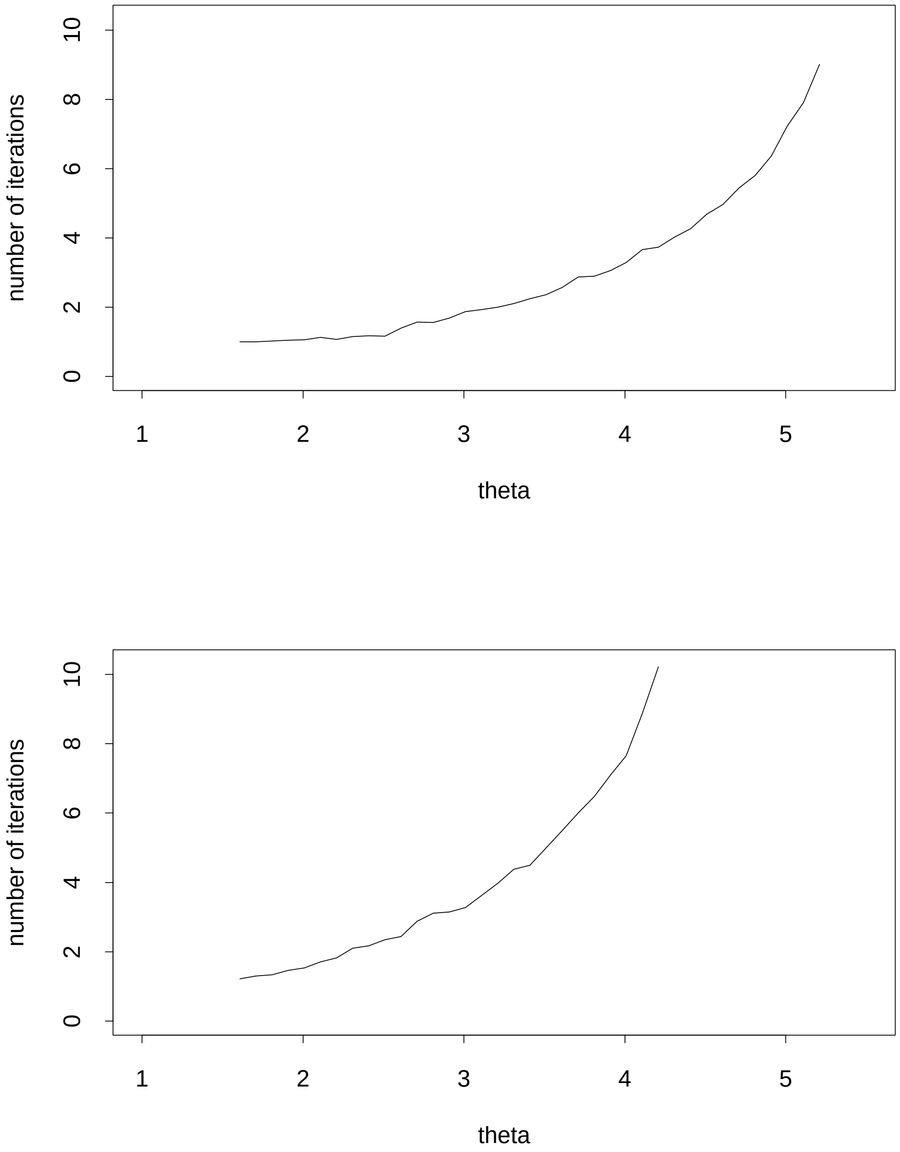

Figure 2: Monte Carlo estimates of the expected number of iterations needed for the symmetric mixture model as a function of θ for interaction radius $R = 0.1$ (top) and $R = 0.2$ (bottom).

10 Conclusion

This article shows that, as with CFTP, Fill's Rejection Sampling algorithm is available and practical for some point processes. We have illustrated how Fill's algorithm can be combined with the Gibbs sampler in [5] to produce exact samples of the penetrable spheres mixture process and associates.
The Gibbs sampler used is restricted to attractive versions of the penetrable spheres mixture process and related models. It would be useful to develop Fill's version of Rejection Sampling for algorithms of greater generality as in [7] and [8] which can e.g. also sample the repulsive area interaction. In fact we examined these methods for a possible application of Rejection sampling. As in [5] the methods in [7] and [8] define quasimaximal and quasiminimal states, but in contrast to [5] these states can not be fixed but are stochastically varying. At present we cannot see how to overcome this barrier.
As we have shown, for some Markov chains CFTP protects against bias caused by user impatience. It would be helpful to find criteria to determine when user impatience bias cannot occur in a CFTP algorithm.
Note that this application of Fill's algorithm provides a *new* perfect simulation algorithm for point processes. Simulation suggests that this new algorithm is approximately as efficient as the analogous algorithm using CFTP. Our algorithm needs more memory than CFTP, but protects against bias caused by user impatience. This is a significant virtue: however the main contribution of this article is to show how one can perform perfect simulation for various point processes in quite a different manner from CFTP.

Acknowledgement I would like to kindly thank my supervisors Prof. W.S. Kendall and Dr. M.N.M van Lieshout for their assistance and many helpful discussions.

References

[1] Baddeley, A.J. and Van Lieshout, M.N.M. (1995) Area-interaction Point Processes, *Annals of the Institute of Statistical Mathematics* **47**, 601 - 619

[2] Chayes, J.T., Chayes, L. and Kotecký, R. (1995) The Analysis of the Widom-Rowlinson Model by Stochastic Geometric Methods, *Communications in Mathematical Physics* **172**, 551 - 569

[3] Fill, J.A. (1997) An Interruptible Algorithm for Perfect Sampling via Markov Chains, *Annals of Applied Probability*, to appear

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

[5] Häggström, O., Van Lieshout, M.N.M. and Møller, J. (1996) Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes, Aalborg Mathematics Department Research Report R-96-2040

[6] Hammersley, J.M., Lewis, J.W.E. and Rowlinson, J.S. (1975) Relationships between the multinomial and Poisson models of stochastic processes, and between the canonical and grand canonical ensembles in statistical mechanics, with illustrations and Monte Carlo methods for the penetrable spheres model of liquid-vapour equilibrium, *Sankhya: The Indian Journal of Statistics*, Series A **37** 457-491

[7] Kendall, W.S. (1997) Perfect Simulation for the Area-Interaction Point Process, *Probability Perspective,* ed. by L. Accardi and C.C Heyde, World Scientific Press, Singapore

[8] Kendall, W.S. and Møller, J. (1997) Perfect Simulation of Point Processes using Metropolis-Hastings algorithms, article in preparation

[9] Klein, W. (1982) Potts-model Formulation of Continuum Percolation, *Physical Review* **B 26**, 2677 - 2678

[10] Lindvall, T. (1992) *Lectures on the Coupling Method,* John Wiley & Sons, New York

[11] Meyn, S.P. and Tweedie, R.L. (1994) Computable Bounds For Geometric Convergence Of Markov Chains, *The Annals of Applied Probability*, **Vol. 4**, No. 4, 981-1011

[12] Møller, J. (1997) Markov Chain Monte Carlo and Spatial Point Processes, in *Stochastic Geometry: likelihood and computation,* edited by Barndorff-Nielsen, O., Kendall, W.S. and Van Lieshout, M.N.M, Chapman & Hall, London, to appear

[13] Propp, J. G (1997) Coupling from the past: a user's guide. Research Report, MIT.

[14] 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

[15] Roberts, G.O. and Polson, N.G. (1994) On the Geometric Convergence of the Gibbs Sampler, *Journal of the Royal Statistical Society* Series B, **56**, 377-384

[16] Rowlinson, J.S (1980) Penetrable spheres model of liquid vapour equilibrium, *Advances in Chemical Physics* **41**, 1-57

[17] Rowlinson, J.S. (1990) Probability densities for some one-dimensional problems in statistical mechanics, in *Disorder in physical systems*, edited by Grimmet, G.R. and Welsh, D.J.A., pages 261 - 276, Clarendon Press, Oxford

[18] Saloff-Coste, L. (1997) Simple examples of the Use of Nash Inequalities for finite Markov Chains, in *Stochastic Geometry: likelihood and computation*, edited by Barndorff-Nielsen, O., Kendall, W.S. and Van Lieshout, M.N.M, Chapman & Hall, London, to appear

[19] Widom, B. and Rowlinson, J.S. (1970) New model for the study of liquid-vapor phase transitions, *Journal of Chemical Physics*, **52**, 1670 - 1684