ISSN 1398-599X MPS-RR 1999-11

Perfect Metropolis-Hastings simulation of locally stable point processes

Wilfrid S. Kendall,
Department of Statistics, University of Warwick,
Coventry CV4 7AL, UK

Jesper Møller,
Department of Mathematical Sciences, Aalborg University,
Fredrik Bajers Vej 7E, DK-9220 Aalborg, DENMARK

March 25, 1999

Abstract

In this paper we investigate the application of perfect simulation, in particular Coupling from The Past (CFTP), to the simulation of random point processes. We give a general formulation of the method of dominated CFTP and apply it to the problem of perfect simulation of general locally stable point processes as equilibrium distributions of spatial birth-and-death processes. We then investigate discrete-time Metropolis-Hastings samplers for point processes, and show how a variant which samples systematically from cells can be converted into a perfect version. An application is given to the Strauss point process.

Keywords: COUPLING FROM THE PAST (CFTP); DOMINATED CFTP; EXACT SIMULATION; LOCAL STABILITY; MARKOV CHAIN MONTE CARLO; METROPOLIS-HASTINGS; PERFECT SIMULATION; PAPANGELOU CONDITIONAL INTENSITY; REALIZABLE MONOTONICITY; SPATIAL BIRTH-AND-DEATH PROCESS; SPATIAL POINT PROCESS; STOCHASTIC MONOTONICITY; STRAUSS PROCESS.

AMS Subject Classification: 62M30, 60G55, 60K35

1 Introduction

Many useful simulation algorithms for spatial processes are indirect and deliver only approximate results, in the sense that the desired distribution is the equilibrium distribution of a Markov chain which the algorithm simulates for a long but not infinite time; therefore the algorithm delivers a sample which is drawn from a distribution not equal to but only approximating the desired distribution. Recently attention has been given to the possibility of exact simulation of spatial processes as used in statistical physics, spatial statistics and stochastic geometry [16, 19, 20, 21, 23, 31, 32, 41], following on from the seminal work of Propp and Wilson [37]. A Markov chain simulation algorithm is said to be *exact* if it is the case that exact equilibrium has in fact been attained when the

algorithm completes; usually the algorithm running time is then random but is still finite. For several reasons simulations cannot be "exact" in the precise sense: useable random number generators always have defects (even if not yet discovered!), while (at least for the methods we describe here) one should admit into one's analysis the possibility that the algorithm fails to deliver an answer within practical constraints of time (but compare [9], which provides an exact simulation algorithm which overcomes this particular defect). We therefore prefer to use the term *perfect simulation* for these "exact" simulation methods.

Perfect simulation is obviously appealing and potentially very useful (given that a good random number generator is used). There is no need to worry about whether one has used an appropriate "burn in" period before beginning sampling; independent and identically distributed sampling is available, so that (for example) asymptotic variances of Monte Carlo estimates can be calculated very directly; and one can assess the approximation error incurred by a "non-perfect" algorithm *via* comparison with an algorithm which is a perfect variation of the original (a useful point if the perfect version of the algorithm is very costly in computational terms). Note also that Murdoch and Rosenthal [35] discuss ways of using information from perfect simulation as efficiently as possible.

It is notable that the idea of perfect or exact simulation represents a substantial and practical contribution of modern probability theory to simulation; a striking contrast with analytic estimates of rates of convergence which (while of great foundational importance) can induce excessive pessimism (a recent survey is given in [7]). Perfect simulation does not estimate convergence rates, but uses the ideas of coupling theory to deliver (in favourable cases) exactly what is required by the practitioner: a certificate that the sample has indeed been drawn from the desired distribution.

The present paper follows up the development of perfect simulation algorithms for spatial point processes described in [16, 21]; while [21] uses spatial birth-and-death processes, and [16] develops a particular Gibbs' sampler, here we investigate the possibility of using Metropolis-Hastings algorithms for rather general finite point processes satisfying a certain local stability condition. In §2 we describe a general theoretical set-up for point processes and introduce the local stability condition. In §3 we present the first general formulation of *dominated CFTP*. In §4 we summarize the spatial birth-and-death process setup in [19, 21] and detail its extension to the case of locally stable point processes. §3 and §4 also act as an introduction to the ideas used in §5, the main section of the paper where perfect Metropolis-Hastings sampling is considered. Finally, §6 is concerned with an application of these ideas to the particular example of a Strauss point process, using an implementation of the algorithm in `C`.

Acknowledgements:

WSK's research was supported by EPSRC grant GR/L56831 and JM's by MaPhySto (Centre for *Ma*thematical *Phy*sics and *Sto*chastics, funded by a grant from the Danish National Research Foundation) and by the Danish Informatics Network in the Agricultural Sciences, funded by a grant from the Danish Research Councils. The collaboration producing this paper was supported by the European Union's research network "Statistical and Computational Methods for the Analysis of Spatial Data. ERB-FMRX-CT96-0095".

2 Locally stable point processes

In this section we summarize the basic notation and assumptions of point process theory, and introduce the local stability assumption which we require for the purposes of our exact simulation algorithm.

Let $(S, \mathcal{B}, \lambda)$ be a measure space endowed with a nonzero finite diffuse measure λ. (In fact the simple case of the unit interval with Lebesgue measure is sufficient to illustrate the details which we will consider; most applications concern the case when S is a compact subset of the plane and λ is an absolutely continuous finite measure.) Following the now-standard construction of [5], we consider the "Carter-Prenter exponential space" $\Omega = \{x \subseteq S : \#(x) < \infty\}$ of finite subsets (point configurations) of S; here $\#(x)$ is the cardinality of the set $x \subseteq S$. This is endowed with the σ-field $\mathcal{F}$ generated by sets $\{x : \#(x \cap B) = n\}$ with $B \in \mathcal{B},\ n = 0, 1, 2, \ldots$. For technical reasons (that is in order to establish ergodicity results for Markov chains considered later on in the paper) we assume that $\mathcal{F}$ is separable. This holds if for example S is a metric separable space and $\mathcal{B}$ is its Borel σ-field. For ease of presentation we restrict ourselves to the case of point configurations without multiple points, by requiring that the measure λ is diffuse; however the ideas in the sequel translate across to the case of multiple points and also to the case of lattice processes with a 'background colour' (see, for example, [33]). We furnish $(\Omega, \mathcal{F})$ with the probability measure μ which corresponds to a Poisson point process on S with intensity measure λ. Hence if the Ω-valued random variable Y has the distribution of μ then $\#(Y)$ follows a Poisson distribution with mean $\lambda(S)$. Moreover, conditional on $\#(Y) = k$, the k points in Y form a binomial process; under this conditioning they are independent and identically distributed with common distribution $\lambda(\cdot)/\lambda(S)$.

Consider the general problem of simulating a point process X on S, whose distribution on the Carter-Prenter exponential space $(\Omega, \mathcal{F})$ has density f with respect to μ. It is assumed that f satisfies a *local stability condition* [12]: there is a constant $K > 0$ such that

$$f(x \cup \{\xi\}) \quad \leq \quad K\, f(x) \tag{2.1}$$

for all finite point configurations $x \subseteq S$ and all points $\xi \in S \setminus x$.Note that this is stronger than Ruelle stability (which would read, $f(x) \leq C \cdot K^{\#(x)}$, and which in turn implies that f is well defined, *i.e.* integrable with respect to μ). The local stability condition Eq. (2.1) implies that f is hereditary: $f(y) > 0$ whenever $f(x) > 0$ and $y \subset x$. Local stability is equivalent to the hereditary condition and a uniform bound on the *Papangelou conditional intensity*

$$\ell^*(x; \xi) \quad = \quad f(x \cup \{\xi\})/f(x) \tag{2.2}$$

(defined to equal 0 if $f(x) = 0$). These conditions are satisfied by most Markov point processes and nearest-neighbour Markov point processes [1, 12, 31, 39] including for example the Strauss process [18, 40], the continuum random-cluster model [16, 25, 30, 31], the saturation and triplets processes discussed in [12] and the area-interaction point process [2]. They are also satisfied by the perimeter interaction point process introduced in [24] in the special case of fixed-size disk or rectangle grains ([19] supplies a proof), but *not* by the Euler interaction point process discussed in [24].

The major issue for perfect simulation of point processes is that we must deal with an infinite state-space, so that there is typically no maximal element,

and (crucially) uniform ergodicity of associated Markov chains is the exception rather than the rule. As pointed out in [11], the failure of uniform ergodicity implies that coalescence of the associated flow or stochastic recursive sequence cannot happen in finite time, so the standard CFTP recipe of the finite state-space work of [37] needs modification. It turns out that we need to introduce a special "dominating process" which acts as a kind of stochastic maximum: the resulting *dominated CFTP* algorithm will provide viable perfect simulation if the dominating process can itself be simulated in statistical equilibrium *and* in reverse-time. In the language of [11], *vertical CFTP* (coupling of realizations started at a fixed time for all possible initial states) is not available if uniform ergodicity fails, but the dominating process and associated constructs provide a way to establish when *horizontal CFTP* (coupling of realizations started at the minimal state at all sufficiently early initial times) has taken place.

This is how the perfect simulation procedure of [21] differs from the finite state-space work of [37]; a further difference is that, following [21], we describe perfect simulation techniques which apply when the underlying point process is not attractive.

A particular innovation of the work presented below is the use of regeneration at the empty set or vacant pattern configuration $\emptyset$; this observation simplifies the proofs of perfect simulation because typically $\emptyset$ is an ergodic atom of the underlying Markov chain. We note here that Murdoch and Green [34] use similar ideas to produce an application to Bayesian inference based on Markov chain Monte Carlo methods: however Murdoch and Green extend what is in effect the Athreya-Nummelin method of splitting the state-space, and so produce regeneration which occurs with sufficient frequency to produce effective CFTP in its own right: our use of regeneration is solely in order to produce theoretical results on the termination of our dominated CFTP algorithm. See also [17] who use similar ideas but in the context of certain Bayesian models for which regeneration is feasible.

3 Perfect simulation in a general context: dominated CFTP

We begin by describing the idea of perfect simulation for point processes in a much more general context, which will specialize both to the context of spatial birth-and-death processes as in §4 and also to the context of Metropolis-Hastings algorithms as in §5.

Our aim is to deliver perfect simulation of the equilibrium distribution of a discrete or continuous time Markov chain $X = \{X(t) : t \geq 0\}$ taking values in a partially-ordered state-space $\mathcal{X}$. We refer to X as the *target chain* started at time 0 and let $\preceq$ denote the partial order relation. We suppose that there is a minimal element $\mathbf{0}$ of $\mathcal{X}$ such that $\mathbf{0} \preceq x$ for all $x \in \mathcal{X}$, but we do *not* suppose that $\mathcal{X}$ has a maximal element. We set $X(0) = \mathbf{0}$.

In the context of this paper $\mathcal{X}$ is the Carter-Prenter exponential space Ω of finite subsets of S, described in the previous section, $x \preceq y$ means $x \subseteq y$, and $\mathbf{0}$ is the vacant pattern $\emptyset$; the simple birth-death process example in [19] sets $\mathcal{X}$ to be the set of nonnegative integers, with $x \preceq y$ meaning $x \leq y$, and $\mathbf{0} = 0$.

In fact the Markov property does not play any part in this section, and it is

convenient to consider a rather more general case. Suppose there is an $\mathcal{X}$-valued *dominating process* $D = \{D(t) : \Leftrightarrow\infty < t < \infty\}$ and an associated *mark process* $M = \{M(t) : \Leftrightarrow\infty < t < \infty\}$ such that simulations of X can be obtained as adapted functionals of (D, M) lying below D in the ordering $\preceq$: for $t \geq 0$,

$$X(t) \quad = \quad F(\{(D(s), M(s)) : 0 \leq s \leq t\}, t) \quad \preceq \quad D(t)\,. \tag{3.1}$$

Furthermore, assume that *target processes* $X_n = \{X_n(t) : \Leftrightarrow n \leq t \leq 0\}$ begun at $X_n(\Leftrightarrow n) = \mathbf{0}$ at times $\Leftrightarrow n \leq 0$ can be constructed similarly as adapted functionals of (D, M): for $\Leftrightarrow n \leq t \leq 0$,

$$X_n(t) \quad = \quad F(\{(D(s), M(s)) : \Leftrightarrow n \leq s \leq t\}, t + n) \quad \preceq \quad D(t)\,. \tag{3.2}$$

It is an exercise in probabilistic coupling techniques (see [26] for a good exposition of coupling theory) to do this for Markov chains on the integers using stochastic recursive sequences [3]. In the case of ordinary CFTP we can use the trivial case when D is constant and equal to the maximal element: in this case the marks provide information (typically independent Uniform$[0, 1]$ random variables) required to simulate the target processes and other processes introduced below. As we will see, it is not also hard to develop such a construction for those Markov chains on Carter-Prenter space Ω in which we have an interest.

To be specific, we assume in the sequel (unless otherwise stated) that we follow [37] by progressively doubling n so that

$$n \quad \in \quad \{1, 2, 4, 8, \dots\}$$

though any strictly increasing sequence of positive integers (or numbers in the continuous time case) may be used instead (see also Remark 3.2 below).

For each such n we also posit the existence of a pair of "sandwiching processes": an *upper bound process* U_n and a *lower bound process* L_n, with $U_n(\Leftrightarrow n) = D(\Leftrightarrow n)$ and $L_n(\Leftrightarrow n) = \mathbf{0}$, constructed using the randomness of (D, M) in an adapted way: for $\Leftrightarrow n \leq t \leq 0$,

$$U_n(t) \quad = \quad F^+(\{(D(s), M(s)) : \Leftrightarrow n \leq s \leq t\}, t + n)\,, \tag{3.3}$$

$$L_n(t) \quad = \quad F^-(\{(D(s), M(s)) : \Leftrightarrow n \leq s \leq t\}, t + n)\,, \tag{3.4}$$

analogously to Eq. (3.2). For notational convenience we extend L_n, X_n, U_n to be defined for all negative times by setting $L_n(t) = X_n(t) = \mathbf{0}$ and $U_n(t) = D(t)$ if $t \leq \Leftrightarrow n$, and we set

$$L_{1/2}(\cdot) = \mathbf{0}, \quad U_{1/2}(\cdot) = D(\cdot).$$

Furthermore, we require that the following *sandwiching properties* hold for all $s \leq t \leq 0$:

$$L_n(t) \quad \preceq \quad X_n(t) \quad \preceq \quad U_n(t) \quad \preceq \quad D(t)\,, \tag{3.5}$$

$$L_n(t) \quad = \quad U_n(t) \quad \text{if} \quad L_n(s) \quad = \quad U_n(s)\,. \tag{3.6}$$

We require also the following *funnelling property* to hold for all $t \leq 0$::

$$L_n(t) \quad \preceq \quad L_{2n}(t) \quad \preceq \quad U_{2n}(t) \quad \preceq \quad U_n(t)\,. \tag{3.7}$$

Certainly such sandwiching processes exist: one example (typically very inefficient for our purposes) is $L_n(\cdot) = \mathbf{0}$, $U_n(\cdot) = D(\cdot)$. Suppose that for any $\Leftarrow n \leq t \leq 0$ and $x \in \mathcal{X}$ we have defined target processes

$$X_n^x(t) \quad = \quad F(\{D(s) : \Leftarrow n \leq s \leq t\}, t+n; x)$$

with $X_n^x(\Leftarrow n) = x$ and which respect the partial order so that

$$X_n^x(t) \quad \preceq \quad X_n^y(t) \quad \text{whenever} \quad x \quad \preceq \quad y\,.$$

Then it can be highly feasible to choose $L_n = X_n^{\mathbf{0}}$, $U_n = X_n^{D(-n)}$ (corresponding to the original case of *monotonic CFTP* if there is a maximal element $\hat{\mathbf{1}}$, and D is set equal to $\hat{\mathbf{1}}$, as in [37]), but (as we will see) it is possible by making other choices to deal effectively with cases where the partial order is *not* respected. In particular, we do *not* require that L_n and U_n evolve like the target process X (typically they need not be individually Markov even if X is). However as soon as $L_n = U_n$ then the sandwiching properties described by Eqs. (3.5)–(3.6) imply that their common subsequent evolution follows a trajectory of X_n. By Eqs. (3.1)–(3.2), if (D, M) is time stationary, the process X_n evolves as a realization of the target process X, and the following theorem uses this to draw strong conclusions as to the constructive simulation of the equilibrium distribution of X, based on simple and verifiable conditions on (D, M).

Theorem 3.1 (The dominated CFTP construction): *Suppose that processes X and L_n, X_n, U_n ($n = 1, 2, 4, \ldots$) are produced from a marked dominating process (D, M) by adapted constructions Eqs. (3.1)–(3.4) so that Eqs. (3.5)–(3.7) are satisfied. Suppose further that the random process (D, M) is stationary in time, that $X(t)$ converges weakly to an equilibrium distribution π as $t \to \infty$, and that $\mathbf{0}$ is an ergodic atom for D (that is to say, the probability of D visiting $\mathbf{0}$ in the time interval $[0, t]$ converges to 1 as time t tends to ∞). Set*

$$T \quad = \quad \inf\{n \in \{1/2, 1, 2, 4, \ldots\} : L_n(0) = U_n(0)\} \tag{3.8}$$

so that $\Leftarrow T$ is the coalescence time *for simulations running up to time* 0. *Then $T < \infty$ almost surely and*

$$L_T(0) \quad = \quad U_T(0) \tag{3.9}$$

follows the equilibrium distribution π.

Proof: Since $\mathbf{0}$ is an ergodic atom, it follows immediately from Eqs. (3.5)–(3.6) that $T < \infty$ almost surely. By Eq. (3.5) and Eq. (3.7), if $n \geq T$ then

$$L_T(0) \quad \preceq \quad L_n(0) \quad \preceq \quad X_n(0) \quad \preceq \quad U_n(0) \quad \preceq \quad U_T(0)\,.$$

But $L_T(0) = U_T(0)$, so $L_T(0) = X_n(0)$ whenever $n \geq T$. Thus with probability one, $L_T(0) = \lim_{n\to\infty} X_n(0)$ exists. By stationarity of (D, M) and the constructions in Eqs. (3.1)–(3.2), $X_n(0)$ has the same distribution as $X(n)$, which itself converges weakly towards π. Consequently, by the dominated convergence theorem, $L_T(0)$ has distribution π. □

Remark 3.2 : We have deliberately formulated Theorem 3.1 in a rather general fashion, so as to apply to non-Markovian situations. To make this practical for an actual simulation context we have to:

(a) identify a stationary marked dominating process (D, M) from which we can build coupled versions of the required target processes X and X_n as indicated in Eqs. (3.1)–(3.2);

(b) identify suitable upper-and lower-sandwich processes U_n, L_n constructed from

$$(D, M)|_{[-n,0]} = \{(D(t), M(t)) : \Leftrightarrow n \leq t \leq 0\}$$

so that the required sandwiching constraints Eqs. (3.5,3.6) and the funnelling property Eq. (3.7) hold;

(c) have a practical way of extending the simulation of (D, M) *backwards* in time.

We can then generate pairs of upper-and lower-sandwiching processes

$$(L_n(0), U_n(0)), \quad n = 1/2, 1, 2, 4, \ldots, T,$$

and return $L_T(0)$ as a perfect simulation of the equilibrium of the target process X_0. This extension of CFTP is sometimes called *coupling into and then from the past*, with the cryptic abbreviation CIA-FTP. The doubling of n is a reasonably efficient way of searching for the time at which coalescence occurs, as pointed out in [37]. Note that we reuse $(D, M)|_{[-n,0]}$ in the simulation of $(D, M)|_{[-2n,0]}$.

Remark 3.3 : In the rest of this paper we show how to work this out for locally stable point processes using the relation

$$x \preceq y \Leftrightarrow x \subseteq y$$

for finite point configurations x, y, and where the minimum is the empty point configuration:

$$\mathbf{0} = \emptyset .$$

As remarked in §2, in the point process context it is typically *not* practical simply to simulate backwards till D hits the regenerative ergodic atom $\mathbf{0}$. Our dominating processes D will be finite-set-valued Markov chains (variously using continuous or discrete time) which satisfy detailed balance and whose equilibrium distributions are easy to simulate. Moreover, we consider cases where D experiences only a finite number of transitions on any bounded time interval, and a mark $M(t)$ is only non-degenerate if D experiences a transition at time t: conditional on D, the non-degenerate marks are conditionally independent and the conditional distribution of $M(t)$ depends only on the transition of D at time t.

We shall consider situations in which requirement **(b)** above will easily follow as soon as requirement **(a)** is fulfilled. For the backwards simulation of (D, M) (see requirement **(c)**) we simply first simulate $D(0)$. Then we exploit

the detailed balance (and consequent reversibility in time) of D by simulating a forwards-time version $\{\tilde{D}(t) : t \geq 0\}$ starting with the initial state $D(0)$, and setting $D(\Leftrightarrow t) = \tilde{D}(t)$. This makes it an easy matter to extend an initial simulation of a final segment of D backwards in time, as required by the CFTP algorithm. Finally, whenever needed, we attach associated simulated marks by generating $M(\Leftrightarrow t)$ from its conditional distribution given the current and previous states of D at time $\Leftrightarrow t$.

Remark 3.4 : In the discrete time case it is possible to cast Theorem 3.1 in the form of an argument concerning stochastic recursive sequences [3] (effectively a nonlinear generalization $X(n+1) = \phi(X(n), D(n+1), D(n), M(n))$ of the notion of *innovations* in time series). We have chosen not to do this because from the implementation point of view it can be preferable *not* to think of generating the entire mark $M(n)$ at each time realized by the CFTP algorithm, but rather progressively to generate those parts of it which are needed as and when the algorithm requires it. As matters turned out, the progressive method was used in the implementation described below (§5) of CFTP in a Metropolis-Hastings' context.

4 Perfect simulation using spatial birth-and-death processes.

In this section we present a general discussion of how to conduct perfect simulation of point processes by using spatial birth-and-death processes to perform dominated CFTP, as described in [19, 21] and implemented there for the particular cases of the area-interaction process and the exclusion process.

4.1 Spatial birth-and-death processes.

We first recall a few properties of spatial birth-and-death processes as introduced in [36].

Consider a spatial birth-and-death process $\{X(t) : t \in [t_0, \infty)\}$ with birth rate b and death rate d, where $b \geq 0$ and $d > 0$ are measurable functions defined on $\Omega \times S$ so that the integral $B(x) = \int b(x, \xi)\, \mathrm{d}\lambda(\xi)$ is finite for all $x \in \Omega$ (the hereditary condition allow us to assume that d is strictly positive). The process X is a stationary jump process which develops as follows. Suppose that we condition on the event that $X(t) = x = \{x_1, \ldots, x_n\}$. Let $E_0, E_1, \ldots, E_n$ be random variables which are mutually independent (and independent of past history), exponentially distributed with means $1/B(x), 1/d((x \setminus x_1), x_1), \ldots, 1/d((x \setminus x_n), x_n)$, respectively (for $B(x) = 0$ we set $E_0 = \infty$). The next transition happens at time $t + E$, where $E = \min\{E_0, E_1, \ldots, E_n\}$. If $E = E_0$ then $X(t + E) = x \cup \{\xi\}$ represents "birth" of a point ξ with density $b(x, \cdot)/B(x)$. If $E = E_i$ with $i > 0$ then $X(t) = x \setminus \{x_i\}$ represents "death" of the point x_i.

Preston [36] gives sufficient conditions for existence, uniqueness and convergence to equilibrium of such a spatial birth-and-death process; conditions ensuring geometrical fast convergence are given in [29]. If detailed balance

[36, 38] is satisfied for a particular density $f : \Omega \to [0, \infty)$,

$$f(x)\, b(x, \xi) \quad = \quad f(x \cup \{\xi\})\, d(x, \xi) \quad > 0 \quad \text{whenever} \quad f(x \cup \{\xi\}) > 0\,, \tag{4.1}$$

then the spatial birth-and-death process is time-reversible and f is the density of the unique stationary distribution π.

In [19, 21] (as well as in many other studies of specific models of spatial birth-and-death processes in the literature) the birth rate is given by the Papangelou conditional intensity of the target point process, while the death rate is constant at 1, corresponding to uniform deletion of existing points:

$$b \quad = \quad \ell^*, \qquad d \quad = \quad 1\,. \tag{4.2}$$

Then local stability of the target point process, as described by Eq. (2.1), ensures the existence and uniqueness of the spatial birth-and-death process, and its distribution converges (in fact geometrically fast) towards the equilibrium π. This follows easily from the fact that the process regenerates at the state $\mathbf{0}$. Indeed this state is an ergodic atom; with probability one, $X(t) = \mathbf{0}$ for infinitely many transition times t. For details, see [36] or [29].

4.2 Coupling construction

Perfect simulation can now be arranged using a construction of the form leading to Theorem 3.1 above; similar remarks can be found in [19, 21], but the formulation and arguments given here become more useful when we want to describe perfect simulation for a wide class of locally stable point processes, whether we use spatial birth-and-death processes as in this section, or whether we use more general Metropolis-Hastings algorithms as studied in §5.

The dominating process D in this section is a spatial birth-and-death process with birth rate $b = K$ and death rate $d = 1$. At each birth or death time t of D we attach as a mark a random number $M(t)$ uniformly distributed on the interval $(0, 1)$. These marks are independent of each other and all other aspects of the construction of D. In fact we ignore the marks attached to death times t of D; these are added to the construction for a purely technical reason, simply to ensure that (D, M) will inherit a time-reversibility property from D. Now consideration of detailed balance shows that D is indeed time-reversible and in equilibrium if we set $D(0)$ to be distributed as a Poisson point process of intensity measure $K\lambda$: thus requirement **(c)** of Remark 3.2 is fulfilled.

As sketched in §4.1, theory shows that this dominating process has ergodic atom $\mathbf{0}$ as required by Theorem 3.1. Assuming that $D(0)$ has the equilibrium distribution, the process (D, M) is stationary by virtue of the reversibility criterion for D and the assignation of independent marks $M(t)$ at the transition times t for D.

We now have to describe how to construct target processes. Here is how a generic target process X with equilibrium density f evolves. Suppose that for a given time t subsequent to the time at which X is started, $y \supseteq x$ are the states in the dominating and target processes just before time t. Suppose that in the dominating process a point ξ is born at time t, so that $D(t) = y \cup \{\xi\}$. Then $X(t) = x \cup \{\xi\}$ if $M(t) \leq \ell^*(x; \xi)$; otherwise there is no change and we set $X(t) = x$. Suppose that in the dominating process a point $\eta \in y$ is deleted

at time t. In that case $X(t) = x \setminus \{\eta\}$ (where η may or may not be included in x). This specifies the construction in Eq. (3.1). Similarly we obtain target spatial birth-and-death processes X_n begun at time $\Leftrightarrow n$ and lying below D in the manner set out in Eq. (3.2), with birth rate $b = \ell^*$, death rate $d = 1$.

All we now need in order to fulfill the conditions of Theorem 3.1 is to exhibit pairs (L_n, U_n) of lower- and upper-sandwich processes obeying Eqs. (3.5)–(3.7). In each case individual points are removed at unit exponential rate, but points are added to L_n at the minimal rate, and to U_n at the maximal rate, which is attainable for realizations of a spatial birth-and-death process with $(b, d) = (\ell^*, 1)$ and which is sandwiched between L_n, U_n. If in the dominating process a point ξ is born at time t and $x^{\min}, x^{\max}$ are the states of L_n, U_n just before time $t \in (\Leftrightarrow n, 0]$, then these rates are $K\alpha^{\min}(x^{\min}, x^{\max}, \xi), K\alpha^{\max}(x^{\min}, x^{\max}, \xi)$, respectively, where

$$\alpha^{\min}(x^{\min}, x^{\max}, \xi) \quad = \quad \min\{\ell^*(x; \xi)/K : x^{\min} \subseteq x \subseteq x^{\max}\} \tag{4.3}$$

$$\alpha^{\max}(x^{\min}, x^{\max}, \xi) \quad = \quad \max\{\ell^*(x; \xi)/K : x^{\min} \subseteq x \subseteq x^{\max}\}. \tag{4.4}$$

Hence, we set $L_n(t) = x^{\min} \cup \{\xi\}$ if $M(t) \leq \alpha^{\min}(x^{\min}, x^{\max}, \xi)$ and $L_n(t) = x^{\min}$ otherwise, while $U_n(t) = x^{\max} \cup \{\xi\}$ if $M(t) \leq \alpha^{\max}(x^{\min}, x^{\max}, \xi)$ and $U_n(t) = x^{\max}$ otherwise. Finally, in the case of a death $D(t) \setminus D(t\Leftrightarrow) = \{\eta\}$ in the dominating process, we set $L_n(t) = L_n(t\Leftrightarrow) \setminus \{\eta\}$ and $U_n(t) = U_n(t\Leftrightarrow) \setminus \{\eta\}$ (here $t\Leftrightarrow$ refers to the "time just before time" t).

Thus birth-times are censored so that a birth occurs for L_n if and only if births occur for all target processes lying between L_n and U_n, and a birth occurs for U_n if and only if a birth occurs for at least one target process lying between L_n and U_n, and in each case the potential birth is of the individual actually born in D. On the other hand death occurs in L_n, U_n exactly when a D-death is a death of an individual in the respective sandwiching population. In summary the births and deaths of L_n, U_n are exactly those forced by the requirements of the sandwiching equations Eqs. (3.5)–(3.6); the funnelling property Eq. (3.7) then follows from the construction.

Remark 4.1 : It is often the case that a spatial point process model is either repulsive or attractive in the following sense:

$$\textit{repulsive case:} \qquad \ell^*(x; \xi) \quad \geq \ell^*(y; \xi) \quad \text{whenever} \quad \xi \notin x \subseteq y\,; \tag{4.5}$$

$$\textit{attractive case:} \qquad \ell^*(x; \xi) \quad \leq \ell^*(y; \xi) \quad \text{whenever} \quad \xi \notin x \subseteq y\,. \tag{4.6}$$

Examples of repulsive models include most pairwise interaction processes such as the Strauss process, and indeed also most of the Ripley-Kelly Markov point processes studied in the literature to date, see [31]. The continuum random-cluster model, the saturation and triplets processes, and the area-interaction process (see §2 for references) are all models which include both repulsive and attractive cases. In the attractive case we have *monotonicity*: for $x^{\min} \subseteq x \subseteq x^{\max}$ and $\xi \notin x^{\max}$, Eq. (4.6) implies that

$$\alpha^{\min}(x^{\max}, x^{\min}, \xi) \;=\; \ell^*(x^{\min}; \xi) \;\leq\; \ell^*(x; \xi) \;\leq\; \ell^*(x^{\max}; \xi) \;=\; \alpha^{\max}(x^{\max}, x^{\min}, \xi)\,; \tag{4.7}$$

whilst the complete opposite holds in the repulsive case Eq. (4.6) as then

$$\alpha^{\min}(x^{\max}, x^{\min}, \xi) \;=\; \ell^*(x^{\max}; \xi) \leq \ell^*(x; \xi) \;\leq\; \ell^*(x^{\min}; \xi) \;=\; \alpha^{\max}(x^{\max}, x^{\min}, \xi)\,. \tag{4.8}$$

The lower and upper processes are therefore individual spatial birth-and-death processes only in the attractive case.

Note that if a locally stable point process is neither attractive nor repulsive then the calculation of $\alpha^{\min}$ and $\alpha^{\max}$ in Eqs. (4.3)–(4.4) may be computationally expensive.

4.3 Algorithm for perfect simulation using spatial birth-and-death processes

In this section we use pseudo-code to sketch out algorithms for perfect simulation using spatial birth-and-death processes based on Remark 3.3 and the coupling construction in §4.2.

The basic algorithm `SBDperfect(`$\Leftarrow n$`)` runs as follows:

`SBDperfect(`$\Leftarrow n$`):`
    $(D, M)|_{[-n,0]} \leftarrow$`SBDDextend(`$D, M, \Leftarrow n$`)`
    $(L_n(0), U_n(0)) \leftarrow$ `SBDevolve(`$D, M, \Leftarrow n$`)`
    `if` $L_n(0) = U_n(0)$ `then`
        `return` $L_n(0)$
    `else`
        `SBDperfect(`$\Leftarrow 2n$`)`

The pseudo-code notation is best explained by discussing this algorithm in words. We begin by extending the current construction of the marked dominating process (D, M) backwards in time to the range $[\Leftarrow n, 0]$ using a sub-procedure `SBDDextend`; we then construct lower and upper processes started at time $\Leftarrow n$ using sub-procedure `SBDevolve`; finally we test for equality of the terminal values $L_n(0)$, $U_n(0)$. If they are equal then the algorithm returns the common value, otherwise it calls itself again recursively using the extended range $[\Leftarrow 2n, 0]$.

Theorem 3.1 assures us that if this algorithm is carried through to completion then the returned value will in fact be distributed according to the target equilibrium distribution. (Notice however that no such guarantee applies to early termination of the algorithm: for this one needs to apply an entirely different method of perfect simulation, namely Fill's User-Interruptible algorithm [9]; Fill's algorithm was first applied to point processes by [41]; see also [32].)

Notice that if `SBDperfect(`$\Leftarrow n$`)` terminates without invoking itself recursively then it requires only the path $(D, M)|_{[-n,0]}$; otherwise an earlier portion of the path is required by the recursive call `SBDperfect(`$\Leftarrow 2n$`)`. Conceptually it is convenient to pretend that (D, M) has already been simulated over the entire range $(\Leftarrow\infty, 0]$. In practice of course we need only generate (D, M) a segment at a time as signalled by `SBDDextend(`$D, M, \Leftarrow n$`)`, extending backwards in time as required; the *time-reversed* simulation is of the same spatial birth-and-death process, started at its equilibrium distribution of a Poisson($K\lambda$) point pattern, and with *deaths* independently marked by independent uniform random marks.

The implementation of `SBDDextend` is a straightforward matter, since it can be achieved by extending the time-reversed dominating process forwards in time as indicated above, and we therefore omit it.

To complete the specification of this algorithm we must define `SBDevolve`. This conducts the coupled evolution of L_n, U_n, implementing the marks-based

construction which was described in words just after Eqs. (4.3)–(4.4) above.

```
SBDevolve(D, ⇔n):
    (x^min, x^max) ← (0, D(⇔n))
    t ← first birth-death incident of D after ⇔n
    while t < 0
        if D(t) = D(t⇔) ∪ {ξ} then
            m ← M(t)
            (x^min, x^max) ← SBDadd(x^min, x^max, ξ, m)
        else
            {η} ← D(t⇔) \ D(t)
            (x^min, x^max) ← (x^min \ {η}, x^max \ {η})
        t ← first birth-death incident of D after t
    return (x^min, x^max)
```

Here $\texttt{SBDadd}(x^{\min}, x^{\max}, \xi, m)$ uses the mark m to compute whether or not to add ξ to each of the patterns $x^{\min}$, $x^{\max}$:

```
SBDadd(x^min, x^max, ξ, u):
    if u ≤ α^min(x^min, x^max, ξ) then
        return (x^min ∪ {ξ}, x^max ∪ {ξ})
    else if u ≤ α^max(x^min, x^max, ξ) then
        return (x^min, x^max ∪ {ξ})
    else
        return (x^min, x^max)
```

Remark 4.2 : Sometimes it is useful to finesse the initialization of $L_n(\Leftrightarrow n) = \mathbf{0}$ in the procedure $\texttt{SBDevolve}(D, M, \Leftrightarrow n)$ (in the line reading $(x^{\min}, x^{\max}) \leftarrow (\mathbf{0}, D(\Leftrightarrow n))$ above), for example by starting the lower-sandwich process in a larger point configuration contained in $D(\Leftrightarrow n)$, selecting those points ξ in $D(\Leftrightarrow n)$ whose birth-marks are lower than the lowest possible value of $\ell^*(x, \xi)/K$ as x ranges over all possible finite subsets of S. These points are exactly those which would have been born in any target process X evolving at the time of their potential birth. As described in [21], in order to obtain these birth-marks we then need to generate the dominating process further back in time up to the occasion when the 'oldest' point in $D(\Leftrightarrow n)$ was born.

Remark 4.3 : Notice that the ergodic nature of $\mathbf{0}$ is relevant to theory rather than practice; as noted in §1, the time needed for D to return to the vacant pattern may be very much larger than the time T we actually need to go back in order to obtain coalescence. In fact, although births arise at a faster rate in the upper process than in the lower process, coalescence can in favourable circumstances happen relatively quickly as a consequence of the coupling between the sandwiching processes L_n and U_n (for example this was the case for experiments reported on in [19, 21]).

Remark 4.4 : The birth-decisions taken in the evolution step `SBDevolve` and specifically `SBDadd` are monotonic in the Papangelou conditional intensity $\ell^*(x; \xi)$. Consequently it is plain that the resulting perfect sample can be viewed

as a complicated dependent random thinning of a realization of the Poisson process corresponding to the case of constant conditional intensity $\ell^*(x;\ell) = K$. Accordingly it is apparent that locally stable point processes (those with conditional intensity satisfying Eq. (2.2)) are all obtained as thinnings of Poisson point processes.

5 Perfect Metropolis-Hastings simulation

We now consider perfect *Metropolis-Hastings* simulation of locally stable point processes, using discrete-time Markov chains rather than spatial birth-and-death processes. For background material on Markov chain Monte Carlo for spatial point processes (including the ideas of Metropolis-Hastings algorithms) the reader is referred to [12, 13, 31] and the references therein. An excellent general account of Markov chain theory is given in [27]. Notice that perfect simulation for a random walk Metropolis-Hastings sampler for use in a Bayesian context is described in [14]; another perfect Metropolis-Hastings algorithm for the autogamma model is discussed in [31]. However to the best of our knowledge the following is the first general account applying the technique of Metropolis-Hastings discrete-time Markov chains to perfect sampling of point processes; the perfect Gibbs' sampling algorithm in [16] applies only on the Widom-Rowlinson multitype point process and related models; the perfect Gibbs samplers (and Fill type algorithms) in [32] apply only on random field approximations of certain spatial point processes; and the perfect slice samplers in [28] apply only when the number of points is fixed in certain spatial point process models. Simulated tempering algorithms introduced in [17] are also based on a general technique for perfect Metropolis-Hastings simulation but for use in multivariate (Bayesian) models (possibly of varying dimension) which in a certain sense can be dominated by a random walk defined on a finite state space.

5.1 Updating schemes

The simplest Metropolis-Hastings version of the algorithm in §4 translates the time continuous approach into discrete time and allows for more than just one point to be added or removed. "Death" corresponds to a proposal for independent p-thinning, using a fixed probability $p \in (0,1)$ for deletion of existing points, while "birth" corresponds to a proposal to add the points from a Poisson point process of intensity measure $pK\lambda$. However, the problem with such a Metropolis-Hastings algorithm (and indeed with a large number of variations which we have investigated) is that it is intrinsically non-monotonic.

This arises because of non-monotonicities in the acceptance probability for proposals, and presents considerable obstacles for efficient "sandwiching" arguments as presented in §4. In particular we have not discovered a variant which uses a dominating chain with a Poisson equilibrium distribution. We illustrate this point by describing an example which does *not* work, but which leads on to the variant that we adopt in this section.

Suppose for simplicity that the *proposal* stage of the Metropolis-Hastings algorithm offers alternatives of birth or death incidents (each involving varying numbers of points as specified below), with probabilities β, $1 \Leftrightarrow \beta$ respectively. The death proposals are for p-thinning of the current configuration x, while the

birth proposals employ superpositioning of a Poisson point process z of intensity measure $pK\lambda$ (recall K is the local stability parameter defined by Eq. (2.1)).

The *acceptance* stage runs as follows. The probability for accepting a birth proposal $x \to x \cup z$ is

$$\alpha(x,z) \quad = \quad \min\left\{1, \frac{f(x \cup z)}{f(x)} \times \frac{(1 \Leftrightarrow p)^{\#(x)}}{K^{\#(z)}}\right\} \tag{5.1}$$

while its "reversed" counterpart, the death proposal $x \cup z \to x$, is always accepted (this convenient simplification is a consequence of the local stability bound). Detailed balance considerations show that if we choose birth proposal probability $\beta = 1/(1 + \exp(\Leftrightarrow pK\lambda(S)))$ then the Metropolis-Hastings algorithm generates a reversible Markov chain with equilibrium density f.

The difficulty for perfect simulation here is that $(1 \Leftrightarrow p)^{\#(x)}$ in Eq. (5.1) is a *decreasing* function of $\#(x)$. Dealing with point processes, we naturally use a partial order $\preceq$ given by set-inclusion $\subseteq$, and this leads to the requirement that at least in the upper- and lower-sandwiching processes we should have monotonicity (in terms of set-inclusion) for *accepted* birth proposals. While we can in principle use obvious extensions of Eqs. (4.3)–(4.4) to assure this, it is to be expected that the resulting dominating process will then be very large. In particular the equilibrium distribution will be much larger than the Poisson process of intensity measure $K\lambda$ (which is a reasonable comparison because of the local stability condition, and also because the continuous-time methods of § 4 use a dominating process which has this equilibrium).

We have considered other variations on Metropolis-Hastings dominated by processes with Poisson equilibrium distributions, all of which run into the same difficulty. Of course, it is possible to generate an algorithm by sampling from a dominating spatial birth-and-death process at regular intervals, leading to a joint equilibrium density $g(x,y)$ with respect to the measure (in configuration variables x, y) given by

$$\mathbf{1}[x \subseteq y] \times (\text{counting measure for } x) \times \mathrm{d}\mu(y)$$

such that the marginal density for the x-component is the required target point process density and the marginal distribution for the y-component is a Poisson process of intensity measure $K\lambda$. However this joint density will in general be very complicated. To derive a suitable Metropolis-Hastings scheme we would need to find a closed-form expression for $g(x,y)$ up to constant of proportionality, so this approach seems infeasible (except by implicit and crude discretization of the spatial birth-and-death approach of §4, which offers no computational advantage).

The problem with the first approach of this subsection is that the dominating process is too large to lead to a feasible algorithm, resulting from the combined effect of non-monotone acceptance probabilities for birth proposals *together with* frequent birth proposals. We can substantially alleviate this problem if we explore a further variation on the Metropolis-Hastings theme, by subdividing the space S into a finite collection of sufficiently small disjoint $\mathcal{B}$-measurable "cells" $C_1, \ldots, C_k \in \mathcal{B}$ with positive cell-contents $\lambda(C_i) > 0$ and such that

- the probability of having more than one point in a cell under the target distribution is small,

- a transition within a cell consists either of a birth or a death of a single point.

The algorithms considered below can then be viewed as "single-cell Metropolis-Hastings".

To implement this we need to devise *different* (but coupled) Metropolis-Hastings algorithms: one algorithm to generate the target Markov chains and another to generate the dominating Markov chain. The dominating Markov chain is effectively the combination of independent random walks one for each cell, negatively biased and reflected at zero. The equilibrium distribution for the dominating chain is no longer Poisson; as we shall see it is "independent Geometric" (in equilibrium the point configurations in different cells are independent, with Geometric numbers of points in each cell C_i, with individual points conditionally distributed in C_i according to the "λ-uniform distribution on C_i" given by $\lambda_i(\cdot) = \lambda(\cdot \cap C_i)/\lambda(C_i)$).

The point configurations within cells are updated by visiting the cells either at random (*i.e.* the next cell is selected uniformly at random) or systematically, either visiting each cell once per turn in some fixed order (in which case the Markov chains are not reversible) or visiting each cell in order and then revisiting cells in reverse order (in which case the Markov chains are reversible); we use the latter *reversible systematic updating scheme* below and in our examples in §6. So one cycle consists of $2k$ updates in the order

$$C_1 \to C_2 \to \ldots \to C_{k-1} \to C_k \to C_k \to C_{k-1} \to \ldots \to C_2 \to C_1 \,. \tag{5.2}$$

(It is an exercise for the reader to modify the following to deal with irreversible systematic updating, or with reversible random updating. The reader is warned that the perfect variant of reversible random updating requires a record to be kept of the random order in which sites are visited.)

If f is a spatially Markov density with a finite range of interaction R then in principle we could obtain a speed-up by implementing a parallel processing scheme in which cells more than R units apart are updated simultaneously (so-called "coding sets" of cells).

5.2 Metropolis-Hastings chains

We first describe the *forward-time* (conventional) simulation algorithms for the dominating chain D respective target chain X which we will then convert to CFTP in §5.4.

We begin with the dominating chain D since it will eventually control everything which goes on. We keep track of the individual updates of D in cycle t by using the extended notation

$$D(t,i) \text{ is the } i^{\text{th}} \text{ update in cycle } t \tag{5.3}$$

so that $D(t,1)$ is obtained from $D(t) = D(t,0)$ by updating in cell C_1, $D(t,2)$ is obtained from $D(t,1)$ by updating in cell C_2, *etc*, and $D(t,2k) = D(t+1) = D(t+1,0)$. In other words, to obtain the configuration at time (t,i) (for $i = 1, \ldots\ 2k$) we update the configuration at time $(t, i \Leftrightarrow 1)$ within the j^{th} cell, where

$$j = \min\{i, 2k \Leftrightarrow i+1\} \,.$$

The single-cell update employs a draw $V(t,i)$ from a Uniform$[0,1]$ distribution to determine whether birth or death is to take place, based on a parameter $p_j \in (0,1)$ as follows:

```
draw V(t,i) from Uniform[0,1]
if V(t,i) ≤ p_j/(1+p_j) then
    arrange for birth in D at time (t,i)
else
    arrange for death in D at time (t,i)
```

Here the $V(t,i)$ draw is independent of all other sources of randomness, and in particular is independent of everything in the *simulation past* (that is, times before (t,i), as opposed to the *algorithm past*, which would involve times later than (t,i) treated in previous recursive passes of the CFTP algorithm).

A birth is arranged as follows: we add $\{\xi(t,i)\}$ to the current dominating population, where the new-born point $\xi(t,i)$ is drawn using the distribution $\lambda_j(\cdot)$ which is the normalization of the restriction to the i^{th} cell of the measure λ:

```
arrange for birth ...
    draw ξ(t,i) from λ_j(·)
    set D(t,i) = D(t, i ⇔1) ∪ {ξ(t,i)}
```

Here again the new-born point $\xi(t,i)$ is independent of all other sources of randomness, and in particular of everything in the simulation past.

A death is arranged by deleting a point $\eta(t,i)$ uniformly at random from the current point pattern:

```
arrange for death ...
    draw η(t,i) from FiniteUniform(D(t, i ⇔1) ∩ C_j)
    set D(t,i) = D(t, i ⇔1) \ {η(t,i)}
```

Here the random death $\eta(t,i)$ is conditionally independent of the simulation past given $D(t, i \Leftrightarrow 1)$ and in fact depends only on $D(t, i \Leftrightarrow 1) \cap C_j$. Furthermore, we adopt the convention that

$$\texttt{draw } \eta \texttt{ from FiniteUniform}(\mathbf{0}) \tag{5.4}$$

returns $\{\eta\} = \mathbf{0}$.

This construction can be viewed as a Metropolis-Hastings algorithm where the proposal kernel is the same as the transition kernel for the resulting Markov chain D (in other words the acceptance probabilities of this Metropolis-Hastings algorithm are set to be equal to 1). A straightforward analysis of detailed balance (see [13]) shows that the equilibrium distribution for D is "independent Geometric": for the counts in cell C_j at equilibrium we have

$$\mathbf{P}\,[\,n \text{ points in } C_j\,] \quad = \quad p_j^n\,(1 \Leftrightarrow p_j) \qquad \text{for } n = 0, 1, 2, \ldots$$

independently in each cell.

We turn next to the Metropolis-Hastings algorithm for the target chain X. Here we reuse the random variables $V(t,i)$ and, in case of births, the newborn

points $\xi(t,i)$; the two chains will be further coupled in §5.3. Using a notation as above and considering the update of cell j at time (t,i) we make the following assignments:

```
setting N(t, i⇔1) = #(X(t, i⇔1) ∩ C_j)

if V(t,i) ≤ p_j ℓ*(X(t, i⇔1); ξ(t,i)) / ((N(t, i⇔1) + p_j + 1)K) then

    arrange for birth in X at time (t,i)

else if V(t,i) ≥ p_j / (N(t, i⇔1) + p_j) then

    arrange for death in X at time (t,i)

else
    do nothing
```

It is crucial for detailed balance here to have the "do-nothing" option. We arrange births and deaths much as before, but coupled to what is happening in the dominating process. For a birth we use

```
arrange for birth ...
    set X(t,i) = X(t, i⇔1) ∪ {ξ(t,i)}
```

(coupling by re-using the birth candidate $\xi(t,i)$; because of the way in which we are re-using the mark $V(t,i)$ the availability of $\xi(t,i)$ is guaranteed) while for a death

```
arrange for death ...
    draw η*(t,i) from FiniteUniform(X(t, i⇔1) ∩ C_j)
    set X(t,i) = X(t, i⇔1) \ {η*(t,i)}                    (5.5)
```

We describe the coupling of $\eta^*(t,i)$ below. In the "do-nothing" option, of course, $X(t,i) = X(t, i \Leftrightarrow 1)$ is left unchanged.

This construction has to be coupled carefully and explicitly to the single-cell update in D. The main issue is to ensure that if $X(t, i\Leftrightarrow 1) = x \subseteq D(t, i\Leftrightarrow 1) = \delta$ then an x-birth can occur only if a δ-birth occurs, and further to ensure that the death of a δ-point forces death of the corresponding x point should that point be alive in X at that time. Notice in particular that x-points die in C_j at rate $\#(x \cap C_j)/(\#(x \cap C_j) + p_j)$ while in the dominating chain they die at the rate $1/(1+p_j)$ (note that if $x \cap C_j = \delta \cap C_j = \mathbf{0}$ is the empty pattern then the two chains stay in $\mathbf{0}$ with the same probability $1/(1+p_j)$). However the death rate *per point* of x is $1/(\#(x \cap C_j) + p_j)$, which (so long as $x \subseteq \delta$) is always lower than the corresponding death rate per point of δ, namely $1/(\#(\delta \cap C_j)(1+p_j))$.

The X chain can be viewed as the result of a Metropolis-Hastings algorithm where at time (t,i) we propose with probability

$$p_j/(N(t, i \Leftrightarrow 1) + p_j) \tag{5.6}$$

to add the $\lambda_j(\cdot)$-uniform point $\xi(t,i)$, and otherwise we delete a point chosen at

random from $X(t, i \Leftrightarrow 1)$. Note that we always accept in case of a death, while we only accept the birth proposal with probability

$$\frac{N(t, i \Leftrightarrow 1) + p_j}{N(t, i \Leftrightarrow 1) + p_j + 1} \times \frac{\ell^*(X(t, i \Leftrightarrow 1); \xi(t, i))}{K}. \tag{5.7}$$

Finally note that the "do-nothing" case arises when the birth proposal is made but rejected.

Analysis of detailed balance (see [13]) shows that X has f as its equilibrium density if the parameters are fixed to obtain

$$p_j \quad = \quad \lambda(C_j)\, K\,, \quad j = 1, \ldots, k\,. \tag{5.8}$$

In the special case $\ell^* = K$ we find that f is the density of a Poisson process of intensity measure $K\lambda$ and the last term in Eq. (5.7) cancels; in the general case, this last term is never greater than 1. However, in contrast to the case of spatial birth-and-death processes described in §4, we cannot obtain the general construction by thinning the construction used to produce the Poisson process of intensity measure $K\lambda$: if $x \subseteq y$ are the current states of two such Metropolis-Hastings chains, then by Eq. (5.6) the chain coming from x has a larger probability for proposing a birth than the chain coming from y.

In the sequel we assume the constraint in Eq. (5.8) to hold.

Remark 5.1 Convergence properties: If we consider the random updating scheme as described in §5.1, then it can be shown that both Metropolis-Hastings algorithms generate Markov chains which are aperiodic, Harris recurrent, and geometrically ergodic. Moreover (useful here principally for theoretical reasons) $\mathbf{0}$ is an ergodic atom. The techniques in [13] and especially [12] apply directly to prove these properties. Incidentally, the target chain is uniformly ergodic if and only if the total number of points is almost surely bounded by a constant under the equilibrium density f; in that case one is tempted to try to truncate in the MH-algorithm for the dominating chain (using the almost sure bound on the number of points) to speed up convergence. However naïve implementations of this for CFTP will fail, since the dominating chain must be large enough to dominate *all* of a class of sample realizations started from $\mathbf{0}$ at times $\Leftrightarrow t$ for sufficiently large t, and for this reason will typically fail to satisfy the upper bound. Note however that in [11] it is shown that in the case of uniform ergodicity it is always *theoretically* possible to implement perfect simulations based on *classic* CFTP, without using a dominating process.

Remark 5.2 Choice of parameters p_j: In our experiments we have used constant $p_j = p$ for all j. Being a probability, the parameter p is constrained to lie within $(0, 1)$, and should not be too close to 1 since otherwise $\#D(t)$ will take a very large range of values when the dominating process is in statistical equilibrium. In § 6 we discuss some empirical findings about useful values for p.

If the local stability condition is replaced by the condition $f(x \cup \{\xi\}) \leq K_j f(x)$ for all x,ξ, j with $\xi \in C_j$, then we can replace K by K_j everywhere above and in the following when updating the j^{th} cell; in particular, Eq. (5.8) is replaced by the constraint

$$p_j \quad = \quad \lambda(C_j)\, K_j\,, \quad j = 1, \ldots, k\,.$$

This modification may prove to be useful when considering perfect simulation for inhomogeneous point processes.

5.3 Coupling construction

In this section we use the notation of §5.2 and consider in more detail what should happen when updating the j^{th} cell at time (t,i) in the dominating marked process and in its subprocesses.

The step requiring real care is the coupling of death incidents. To conceptualize the actual algorithm we imagine that if $D(t,i \Leftrightarrow 1) = \{\delta_1, \ldots, \delta_m\}$ with $m \geq 1$, then (no matter if D experiences a death or not) at time (t,i) there is also attached a mark $\Sigma(t,i)$ to $D(t,i)$ which is a random permutation used to specify the order $\bar{D}(t,i \Leftrightarrow 1) = (\delta_1, \ldots, \delta_m)$ in which the points can be deleted at time (t,i) when considering subprocesses of D (such as X_n, L_n, U_n and D itself). Formally, conditional on $D(t,i)$, $\Sigma(t,i)$ is a uniformly distributed permutation of the points in $D(t,i \Leftrightarrow 1)$, and $\Sigma(t,i)$ is conditionally independent of the simulation past. From an implementation point of view this risks a heavy computational demand; however, as discussed in more detail in §5.4, in the perfect Metropolis-Hastings algorithm we need only implement the simulation of $\Sigma(t,i)$ in an implicit and partial manner, since we will only require to know *some* aspects of the ordering $\bar{D}$.

The dominating chain evolves as described in §5.2. If D experiences a death at time (t,i) then the point to be deleted is

$$\eta(t,i) \quad = \quad \text{first point in } \bar{D}(t,i \Leftrightarrow 1)$$

with the convention $\{\eta(t,i)\} = \mathbf{0}$ if $D(t,i \Leftrightarrow 1) = \mathbf{0}$. This is in accordance with the specification of Eq. (5.4). The marks for the dominating process are now specified by setting $M(t,i) = (V(t,i), \Sigma(t,i))$. Under the assumption that $D(0)$ has the equilibrium distribution, we see that (D,M) is stationary. Further, $\mathbf{0}$ is an ergodic atom of D, as required by Theorem 3.1.

A generic target Metropolis-Hastings chain X with equilibrium density f evolves as described in §5.2 but with death candidates specified as follows. If X experiences a death at time (t,i), the point to be deleted is

$$\eta^*(t,i) \quad = \quad \text{first point in } \bar{D}(t,i \Leftrightarrow 1) \text{ which is contained in } X(t,i \Leftrightarrow 1)$$

(with $\{\eta^*(t,i)\} = \mathbf{0}$ if $X(t,i \Leftrightarrow 1) = \mathbf{0}$). This is in accordance with the distribution in Eq. (5.5). Note that $\eta^*(t,i) = \eta(t,i)$ if D experiences a death at time (t,i) and $\eta(t,i) \in X(t,i \Leftrightarrow 1)$. Thereby we maintain the requirement that $X(\cdot) \subseteq D(\cdot)$ and the construction of X is completed as an adapted functional of D (see Eq. (3.1)).

In order to see how to construct target Metropolis-Hastings chains X_n begun with value $\mathbf{0}$ at time $\Leftrightarrow n$, we have first to specify the conditional distribution of $M(t,i)$ given $(D(t,i \Leftrightarrow 1), D(t,i))$ (Remark 3.3). Then $\Sigma(t,i)$ is still a uniformly distributed permutation of the points in $D(t,i \Leftrightarrow 1)$, and $\Sigma(t,i)$ is (conditionally) independent of $V(t,i)$ which is uniformly distributed in accordance to whether D experiences a birth or a death at time (t,i). So let

$$(\{\xi(t,i)\}, \{\eta(t,i)\}) \quad = \quad (D(t,i) \setminus D(t,i \Leftrightarrow 1), D(t,i \Leftrightarrow 1) \setminus D(t,i))$$

specify which point has been added or which point has been deleted in the dominating chain considered forwards in time. Then, conditional on $(D(t,i \Leftrightarrow 1), D(t,i))$, we require

$$V(t,i) \quad \text{is distributed as } \texttt{Uniform}[0, 1/(1+p_j)] \qquad \text{if } \{\xi(t,i)\} \neq \mathbf{0}\,, \quad (5.9)$$

$$V(t,i) \quad \text{is distributed as } \texttt{Uniform}[1/(1+p_j), 1] \qquad \text{if } \{\xi(t,i)\} = \mathbf{0}\,. \quad (5.10)$$

Notice that it is convenient to interpret the case $\{\eta(t,i)\} = \{\xi(t,i)\} = \mathbf{0}$ as a death. Furthermore, for each target process X_n $(n = 1, 2, 4, \dots)$,

$$(\{\xi_n(t,i)\}, \{\eta_n(t,i)\}) \quad = \quad (X_n(t,i) \setminus X_n(t, i \Leftrightarrow 1), X_n(t, i \Leftrightarrow 1) \setminus X_n(t,i)) \,.$$

Then for $\Leftrightarrow n < t \leq 0$, X_n experiences either "do-nothing" (more precisely the case $X_n(t, i \Leftrightarrow 1) = X_n(t,i) \neq \mathbf{0}$) birth or death:

$$\texttt{setting } N_n(t, i \Leftrightarrow 1) = \#(X_n(t, i \Leftrightarrow 1) \cap C_j)$$

$$\texttt{if} \quad V(t,i) \leq \frac{p_j \, l^*(X_n(t, i \Leftrightarrow 1), \xi(t,i))}{(N_n(t, i \Leftrightarrow 1) + p_j + 1)K} \quad \texttt{then}$$

$$\texttt{arrange birth in } X_n \texttt{ at time } (t,i)$$

$$\texttt{else if } V(t,i) \geq \frac{p_j}{N_n(t, i \Leftrightarrow 1) + p_j} \texttt{then}$$

$$\texttt{arrange death in } X_n \texttt{ at time } (t,i)$$

$$\texttt{else}$$
$$\texttt{do nothing}$$

If X_n experiences birth at time (t,i), then D experiences birth at time (t,i), and so we set $\xi_n(t,i) = \xi(t,i)$ and $\{\eta_n(t,i)\} = \mathbf{0}$. If X_n experiences death at time (t,i), then $\{\xi_n(t,i)\} = \mathbf{0}$ and $\eta_n(t,i)$ is the first point in $\bar{D}(t, i \Leftrightarrow 1)$ which is contained in $X_n(t, i \Leftrightarrow 1)$ (setting $\{\eta_n(t,i)\} = \mathbf{0}$ if $X_n(t, i \Leftrightarrow 1) = \mathbf{0}$). Finally, in the case of "do-nothing", $\{\eta_n(t,i)\} = \{\xi_n(t,i)\} = \mathbf{0}$. Thereby the construction in Eq. (3.2) is completed.

Monotonicity of birth and death incidents in lower and upper processes (L_n, U_n) $(n = 1, 2, 4, \dots)$ follows from the following assignments. Using the notation of Eqs. (4.3)–(4.4), for births we have the following:

$$\texttt{if} \quad V(t,i) \leq \frac{p_j \, \alpha^{\min}(U_n(t, i \Leftrightarrow 1), L_n(t, i \Leftrightarrow 1), \xi(t,i))}{\#(U_n(t, i \Leftrightarrow 1) \cap C_j) + p_j + 1} \quad \texttt{then}$$

$$\texttt{arrange birth in } L_n \texttt{ and } U_n \texttt{ at time } (t,i)$$

$$\texttt{else if} \quad V(t,i) \leq \frac{p_j \, \alpha^{\max}(U_n(t, i \Leftrightarrow 1), L_n(t, i \Leftrightarrow 1), \xi(t,i))}{\#(L_n(t, i \Leftrightarrow 1) \cap C_j) + p_j + 1} \quad \texttt{then}$$

$$\texttt{arrange birth in } U_n \texttt{ alone at time } (t,i)$$

$$\texttt{else}$$
$$\texttt{arrange no births}$$

and for deaths we have

```
if        V(t,i) ≥ p_j / (#(L_n(t, i⇔1) ∩ C_j) + p_j)        then
    arrange deaths in L_n and U_n at time (t,i)
else if   V(t,i) ≥ p_j / (#(U_n(t, i⇔1) ∩ C_j) + p_j)        then
    arrange death in L_n alone at time (t,i)
else
    arrange no deaths
```

Births and deaths are implemented as follows: if we set

$$\begin{aligned}(\{\xi_n^{\min}(t,i)\},\{\eta_n^{\min}(t,i)\}) &= (L_n(t,i)\setminus L_n(t,i\Leftrightarrow 1), L_n(t,i\Leftrightarrow 1)\setminus L_n(t,i))\,,\\ (\{\xi_n^{\max}(t,i)\},\{\eta_n^{\max}(t,i)\}) &= (U_n(t,i)\setminus U_n(t,i\Leftrightarrow 1), U_n(t,i\Leftrightarrow 1)\setminus U_n(t,i))\end{aligned}$$

then $\xi_n^{\min}(t,i)=\xi(t,i)$ (respectively, $\xi_n^{\max}(t,i)=\xi(t,i)$) in case of birth incidence in the lower (respectively, upper) process. Further, in case of death in U_n, $\eta_n^{\max}(t,i)$ is the first point in $\bar{D}(t,i\Leftrightarrow 1)$ which is contained in $U_n(t,i\Leftrightarrow 1)$. Similarly, in case of death in L_n, $\eta_n^{\min}(t,i)$ is the first point in $\bar{D}(t,i\Leftrightarrow 1)$ which is contained in $L_n(t,i\Leftrightarrow 1)$.

A simple induction argument shows that Eqs. (3.5)–(3.7) are satisfied, and so Theorem 3.1 applies.

Remark 5.3 : Notice that we allow a slight inefficiency in the way birth probabilities for lower and upper processes are specified. For example, for birth in L_n at time (t,i), we could use the sharper upper bound given by

$$\frac{p_j}{K}\min\left\{\frac{\ell^*(x;\xi(t,i))}{\#(x\cap C_j)+p_j+1}\;:\;L_n(t,i\Leftrightarrow 1)\subseteq x\subseteq U_n(t,i\Leftrightarrow 1)\right\}$$

and similarly for U_n (where min is replaced by max).

5.4 Algorithms for perfect Metropolis-Hastings simulation

The general form of the CFTP variation follows that of the algorithm for perfect simulation using spatial birth-and-death processes:

```
MHperfect(⇔n):
    D|_[−n,0] ←MHDextend(D, ⇔n)
    (L_n(0), U_n(0)) ← MHevolve(D, ⇔n)
    if L_n(0) = U_n(0) then
        return U_n(0)
    else
        MHperfect(⇔2n)
```

---

where the sub-procedures act as follows:

- `MHDextend`$(D, \Leftarrow\!\!n)$ exploits reversibility to extend D backwards in time to be defined over $[\Leftarrow\!\!n, 0]$, and also generates the marks $V(t,i)$ in accordance to Eqs. (5.9)–(5.10)
- the procedure `MHevolve`$(D, \Leftarrow\!\!n)$ then generates upper- and lower-sandwich processes working forwards in time over the range $[\Leftarrow\!\!n, 0]$.

Once again the definition of `MHDextend` is straightforward, based on the reversibility of D and the ease with which one can simulate from its equilibrium distribution and from Eqs. (5.9)–(5.10). Note however that we also require `MHDextend` to set the lower- and upper-sandwich processes $L_n(t)$ and $U_n(t)$ to their default values (respectively, $\mathbf{0}$ and $D(t)$) where they have not already been defined by previous invocations of `MHDextend`, i.e. when $\Leftarrow\!\!n \leq t < \Leftarrow\!\!n/2$ and $n \geq 2$.

Instead of generating the entire mark $\Sigma(t,i)$, we choose instead to generate the death candidates of upper and lower processes "on the fly", compatibly with the above prescription but only generating that part of the random permutation $\Sigma(t,i)$ required at the time. Essentially this means we have to select the required death candidates η with the correct probabilities conditional on previous choices for earlier sandwiching processes, so as to maintain the funnelling property Eq. (3.7). We do this largely for convenience of implementation: note however there is also an underlying issue of *realizable monotonicity* which should be born in mind. Suppose we have *stochastic monotonicity* of the kind observed above; death rates per individual decrease as the population increases, but the overall death probability increases. In general it is not possible to choose a death candidate consistently for *all* possible sub-populations, using the correct conditional probabilities. Consider for example a grand population of 3 individuals, such that individual i has probability $1/2$ of being chosen in the population $\{i\}$, probability $1/2$ of being chosen in $\{i,j\}$, and probability $1/3$ of being chosen in the grand population $\{i,j,k\}$. As an exercise for the reader, it is not possible to make choices of individuals in the ${}^3C_2$ different populations such that at most one individual is chosen in each population, if i is chosen in $\{i,j,k\}$ then it is chosen in $\{i,j\}$, and if it is chosen in $\{i,j\}$ then it is chosen in $\{i\}$. This relates to recent work on realizable monotonicity due to Fill and Machida [10].

Consider first any three point configurations $x \subseteq y \subseteq z$ with (possible) candidates $\{\eta_x\} \subseteq x$, $\{\eta_y\} \subseteq y$, $\{\eta_z\} \subseteq z$ for deaths, where we interpret $\{\eta_w\} = \mathbf{0}$ as the case where there is no death-nomination for the "population" w (with $w = x$, y or z). Assume that

$$\begin{aligned} \eta_z \in y \quad &\Rightarrow \quad \eta_y = \eta_z \text{ assuming } \eta_z \text{ exists,} \\ \eta_y \in x \quad &\Rightarrow \quad \eta_x = \eta_y \text{ assuming } \eta_y \text{ exists,} \\ &\qquad\quad \eta_z \text{ is drawn from } \texttt{FiniteUniform}(z)\,, \\ x \text{ is nominated for death} \quad &\Rightarrow \quad \eta_x \text{ is drawn from } \texttt{FiniteUniform}(x)\,. \end{aligned}$$

Conditional on the values of $(x, y, \{\eta_x\}, \{\eta_z\})$ and the fact that y is nominated for death, we can generate η_y as drawn from a `FiniteUniform`(y) distribution by using

```
MHdeath(x, y, η_x, η_z):
    if η_z ∈ y then
        η_y ← η_z
    else
        draw W from Uniform[0,1]
        if W ≤ #(x)/#(y) then
            η_y ← η_x
        else
            draw η_y from FiniteUniform(y \ x)
    return η_y
```

where we set $\#(x)/\#(y) = 0$ if $\#(y) = 0$.

Now, set

$$(\{\xi_{1/2}(t,i)\}, \{\eta_{1/2}(t,i)\}) \quad = \quad (\{\xi(t,i)\}, \{\eta(t,i)\})\,,$$

and consider any $n \in \{1, 2, 4, \dots\}$. Suppose we condition on

(i) U_n experiences a death at time (t,i),

(ii) the realisation of $(L_{1/2}(s,l), U_{1/2}(s,l), \dots, L_{n/2}(s,l), U_{n/2}(s,l))$ for $\Leftrightarrow n \le s \le t$ and $1 \le l \le 2k$ with $l \le i$ when $s = t$,

(iii) the realisation of $(L_n(s,l), U_n(s,l))$ for $\Leftrightarrow n \le s \le t$ and $1 \le l \le 2k$ with $l < i$ when $s = t$.

Then $\eta_n^{\max}(t,i)$ (the death candidate in U_n at time (t,i)) follows the same conditional distribution as

$$\texttt{MHdeath}(L_{n/2}(t, i \Leftrightarrow 1) \cap C_j, U_n(t, i \Leftrightarrow 1) \cap C_j, \eta_{n/2}^{\min}(t,i), \eta_{n/2}^{\max}(t,i))$$

Suppose we in addition condition on

(iv) L_n experiences a death at time (t,i),

(v) the realisation of $U_n(t,i)$.

Note that, by construction, **(iv)** implies **(i)**. Then $\eta_n^{\min}(t,i)$ (the death candidate in L_n at time (t,i)) follows the same conditional distribution as

$$\texttt{MHdeath}(L_{n/2}(t, i \Leftrightarrow 1) \cap C_j, L_n(t, i \Leftrightarrow 1) \cap C_j, \eta_{n/2}^{\min}(t,i), \eta_n^{\max}(t,i))$$

The final form of `MHevolve` is as given in Table 1. Notice that, as was made plain in Theorem 3.1, there is no need to simulate the actual processes X_n. The upper- and lower-sandwich processes suffice for perfect simulation of the equilibrium distribution at time 0.

Remark 5.4 : The actual implementation in [22] differs from the above algorithm in detail, in that the marks are used in a different way. The implementation systematically exploits ordering relations in the coupling (for example, deaths in upper-sandwich processes force the existence of deaths in lower-sandwich processes) so as where possible to avoid computation of conditions such as whether or not

$$V(t,i) \quad \ge \quad \frac{p_j}{\#(L_n(t, i \Leftrightarrow 1) \cap C_j) + p_j}\,.$$

`MHevolve`(D, n) :

$(L_n(\Leftrightarrow n), U_n(\Leftrightarrow n)) \leftarrow (\mathbf{0}, D(\Leftrightarrow n))$

$t \leftarrow 1 \Leftrightarrow n$

`while` $t \leq 0$

`for` $i = 1 \ldots 2k$

$j \leftarrow \min\{i, 2k+1 \Leftrightarrow i\}$

`if` $V(t,i) \leq \dfrac{p_j\, \alpha^{\max}(U_n(t, i \Leftrightarrow 1), L_n(t, i \Leftrightarrow 1), \xi(t,i))}{\#(L_n(t, i \Leftrightarrow 1) \cap C_j) + p_j + 1}$

`then` $(\eta_n^{\max}(t,i), \xi_n^{\max}(t,i)) \leftarrow (\mathbf{0}, \xi(t,i))$

`else if` $V(t,i) \geq \dfrac{p_j}{\#(U_n(t, i \Leftrightarrow 1) \cap C_j) + p_j}$

`then` $\eta_n^{\max}(t,i) \leftarrow$
`MHdeath`$(L_{n/2}(t, i \Leftrightarrow 1) \cap C_j, U_n(t, i \Leftrightarrow 1) \cap C_j, \eta_{n/2}^{\min}(t,i), \eta_{n/2}^{\max}(t,i))$

`else` $(\eta_n^{\min}(t,i), \xi_n^{\min}(t,i)) \leftarrow (\mathbf{0}, \mathbf{0})$

`if` $V(t,i) \leq \dfrac{p_j\, \alpha^{\min}(U_n(t, i \Leftrightarrow 1), L_n(t, i \Leftrightarrow 1), \xi(t,i))}{\#(U_n(t, i \Leftrightarrow 1) \cap C_j) + p_j + 1}$

`then` $(\eta_n^{\min}(t,i), \xi_n^{\min}(t,i)) \leftarrow (\mathbf{0}, \xi(t,i))$

`else if` $V(t,i) \geq \dfrac{p_j}{\#(L_n(t, i \Leftrightarrow 1) \cap C_j) + p_j}$

`then` $\eta_n^{\min}(t,i) \leftarrow$
`MHdeath`$(L_{n/2}(t, i \Leftrightarrow 1) \cap C_j, L_n(t, i \Leftrightarrow 1) \cap C_j, \{\eta_{n/2}^{\min}(t,i)\}, \{\eta_n^{\max}(t,i)\})$

`else` $(\eta_n^{\min}(t,i), \xi_n^{\min}(t,i)) \leftarrow (\mathbf{0}, \mathbf{0})$

$L_n(t,i) \leftarrow (L_n(t,i) \setminus \{\eta_n^{\min}\}) \cup \{\xi_n^{\min}\}$
$U_n(t,i) \leftarrow (U_n(t,i) \setminus \{\eta_n^{\max}\}) \cup \{\xi_n^{\max}\}$

Table 1: Summary of subprocedure `MHevolve` for perfect Metropolis-Hastings simulation of a locally stable point process.

Consequently the Uniform$[0,1]$ marks are used to implement events under various conditionings. However the end-result is as given above.

Remark 5.5 : In order to reduce the cost of calculations, `MHevolve` may be slightly modified. For instance, to avoid unnecessary calculations of $\alpha^{\max}$ ($\alpha^{\min}$) for the birth probability in the upper (lower) process, we can first check in the upper (lower) process if a death takes place; and if not, whether the

dominating chain experiences a birth (since birth in the upper (lower) process implies birth in the dominating chain); and if it does, then we finally check if a birth or "do-nothing" takes place in the upper (lower) process.

6 Discussion and application to the Strauss process

In the above we have described a new algorithm which uses discrete-time Metropolis-Hastings methods to conduct perfect simulation of locally stable point processes, and we have given a general formulation and theoretical framework, and described how the continuous-time birth-and-death methods of [19, 21] fit into this framework. The Metropolis-Hastings approach may be categorized as "variable-cell Metropolis-Hastings" applied to independent asymmetric random walks reflected at the origin, one for each cell of an array of cells dividing up the region in which the point process is being simulated.

In Remark 5.2 we commented on the issue of choice of parameters for our Metropolis-Hastings algorithm. For a given point process in a specified window, cell-size is balanced against the parameter $p = p_j$ (which we hold constant over all cells for the sake of simplicity) controlling the probability of proposing birth or death in the dominating process. Here we report on numerical experiments concerning the Poisson and Strauss point processes. A typical invocation of the program implementing the Metropolis-Hastings algorithm to generate a Poisson point process might be

$$\texttt{mh-cftp -i "poisson" -n } m^2 \texttt{ -m } m \texttt{ -z } z \texttt{ -p } p \texttt{ -s } s\,,$$

which would produce a point pattern over a square region S divided into $n = m^2$ square cells of side-length z, using random number seed s, the mean number of points in S being np. Table 2 shows mean times (in seconds, measured on a Sun UltraSparc) taken to attain perfect simulation using various seeds, working with various invocations of `mh-cftp` using a Strauss process with density proportional to $\gamma^{s_r(x)}$ with reference to the Poisson process considered above. Here $r = 1.5$ is the interaction radius, $\gamma = 0.5$ is the interaction parameter, and $s_r(x)$ denotes the number of pairs of points in x which are closer to each other than the interaction radius. Note that Eqs. (2.1,5.8) hold with $K = p/z^2$.

As can be seen in Table 2, there is weak evidence that efficiency is maximized for p in the region of 0.4. However it should be noted that standard deviations based on 30 replicates were of the order of ± 0.3. The particular form of the CFTP algorithm, especially its search for coalescence using doubling, means that this analysis is indicative only. A full analysis would be more complicated than is really justified at this exploratory stage.

The rewards of perfect simulation carry with them a number of costs. In joint work with Burdzy one of us has begun to delineate the costs in terms of simulation time [4]; a further cost is that considerable care is required to make sure that the CFTP algorithm really is coalescing towards the correct process, and hence that one really is sampling perfectly from the target distribution rather than some other. In our implementation we have taken a number of precautions, further details of which may be found in the associated technical report [22]. These include: debugging code which dumps a complete trace of the

n	m	p (3dp)	z (3dp)	mean time
121	11	0.826	0.909	2.20
144	12	0.694	0.833	1.10
169	13	0.592	0.769	0.99
196	14	0.510	0.714	0.86
225	15	0.444	0.667	0.83
256	16	0.391	0.625	0.71
289	17	0.346	0.588	0.93
324	18	0.309	0.556	0.94
361	19	0.277	0.526	0.93
400	20	0.25	0.5	0.99

Table 2: Mean coalescence times in seconds for perfect Metropolis-Hastings simulation of a Strauss process using `mh-cftp` with varying cell-width z and a 10×10 window. Note np is held constant at 100.

lower- and upper-sandwich simulations, liberal use of the `C assert` facility, and also (perhaps most important) statistical tests to examine whether the output appears to be drawn from the correct distribution. Details of this last are given in [22, §7], and also include an `Splus` analysis based on a partial likelihood argument applied to marginal transition rates of the upper- and lower-sandwich processes, aimed at checking that individual birth-death transitions occur at the correct rates. The whole subject of verification of simulation needs to be taken very seriously when one is trying to attain perfect simulation!

In the present paper we have concentrated on presenting a conceptual framework for dominated CFTP *via* Theorem 3.1, reinterpreting the construction of [19, 21] in this framework, and describing a new implementation of dominated CFTP for point processes using Metropolis-Hastings ideas. We have chosen not to give general guidance on how to construct new dominated CFTP algorithms for two reasons. Firstly, a clear conceptual framework together with two explicit examples conveys the actual flavour of the technique, and the good effect could be diluted if accompanied by a number of generalities. We believe it will be more useful for readers to examine for example our discussion of the spatial birth-and-death process in §4, and our brief report in §5.1 of the way in which we were led to the above implementation of perfect simulation for an MH algorithm. Secondly, and more importantly, the area is still in its early phase of very rapid development, and well-intended attempts at guidance might actually serve to stifle development. For example: at an earlier stage assertions were in circulation that CFTP could only ever apply to uniformly ergodic situations: the technique of dominated CFTP exactly refutes this. New application of CFTP frequently require new conceptual ideas: for a minor example see [23], which enlarges the state-space by adding a transient component so as to deliver CFTP for a conditioned Boolean model. There are in fact several general frameworks now available to "explain" CFTP (for example, the notion of *stochastic recursive sequences*, which has been around since well before CFTP and is surveyed in [3]; [21] hints at a continuous-time approach based on coalescing stochastic flows as in [8]): in Theorem 3.1 we work towards a minimalist framework in order to make opportunity for innovative algorithm design and implementation. Note in particular that while *in principle* one could force the §5 MH-CFTP im-

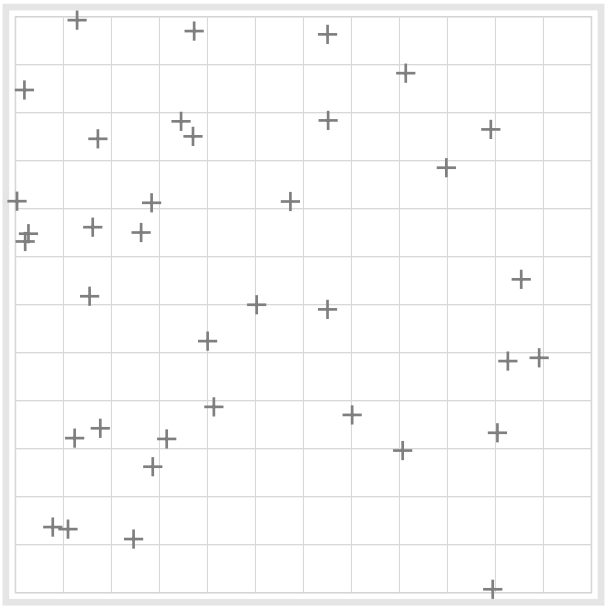

Figure 1: Output of `C` program `mh-cftp`: Strauss point process (37 points).

plementation into the framework of stochastic recursive sequences (as already mentioned in Remark 3.4), constructing marks to record the whole possible pattern of nested deaths as described in §5.3, in practice it may be conceptually simpler, and easier from an implementation point of view, to work with the framework of Theorem 3.1.

To illustrate the Metropolis-Hastings algorithm of §5 we describe here the results of perfect simulation from the Strauss point process. Figure 1 is a simulation of a Strauss process with density proportional to $\gamma^{s_r(x)}$ with reference to a Poisson point process of intensity 2 within a square region of side length 6 units, and where $r = 1$ and $\gamma = 0.5$. The square region is divided into 12×12 square cells of size 0.5×0.5, and we set all $p_j = 1/2$ so that Eqs. (2.1,5.8) hold with $K = 2$. The simulation was modest in computational demands: it took 30 seconds (inclusive of producing `Postscript` output) within $640k$ RAM on an old 25MHz Acorn A440/1 with `ARM3` *cpu* (it should be noted that the same `C` program also compiled and ran on Sun workstations under `Solaris` and a Pentium 230 MHz machine running `Windows 95`, both using the `gcc` compiler).

Figure 2 was produced by the same program, using the same seed for its random-number generator, with no interaction ($\gamma = 1$), in which case the result is a Poisson process. Note that the Poisson realization need *not* in general include corresponding Strauss realizations as subsets, in contrast to what would arise from a perfect simulation using spatial birth-and-death processes as in §4.

More ambitious simulations can certainly be produced by this program, whether by increasing the amount of interaction between points, or by increasing the interaction radius, or by enlarging the region or increasing the intensity of the underlying Poisson point process, or indeed by writing new `C` functions to

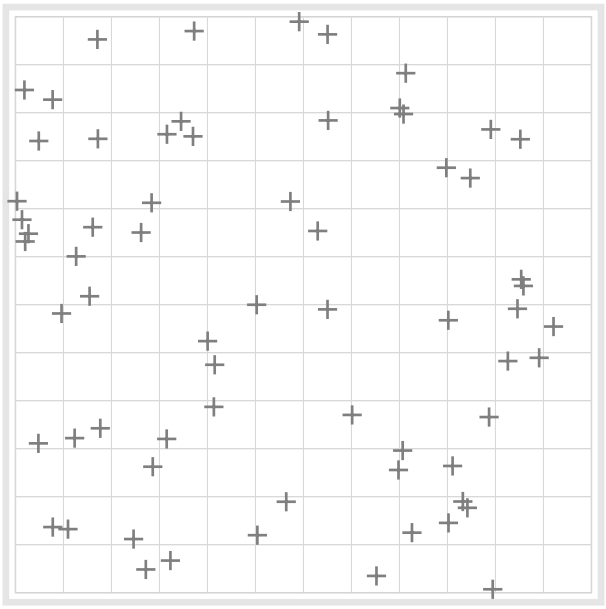

Figure 2: Output of `C` program `mh-cftp`: Poisson point process (66 points) produced using the seed which was used for the Strauss point process illustrated above.

implement other locally-stable point processes. One should however note here an intrinsic limitation of this particular method of perfect simulation, noted already in [21]. If the interaction becomes too intense, and the radius of interaction relative to the underlying intensity becomes too large, then percolation effects may be expected to set in. This will lead to extremely slow convergence (and, incidentally, heavy dependence on edge-effects; see [20] for a discussion of how to apply perfect simulation to deal with edge-effects in a non-percolative situation, and see [15, 42] for extremely interesting results in a Markov random field context).

It would be interesting to compare the efficiency of the various approaches for perfect simulation of locally stable point processes based on running spatial birth-and-death processes as in §4.3 and Metropolis-Hastings chains as in §5.4. This would give an interesting perspective on unpublished results of Clifford and Nicholls [6]. Their results include a simulation study which indicates that neither method dominates the other in terms of efficiency.

Finally, note that the perfect simulation algorithms presented here are general algorithms for locally stable point processes, and pay a price for this generality. For example, in the particular case of the Widom-Rowlinson model we expect the Gibbs' sampler in [16] to be more efficient than any of the algorithms considered in the present paper.

References

[1] A.J. Baddeley and J. Møller. Nearest-neighbour Markov point processes and random sets. *International Statistical Review*, 57:89–121, 1989.

[2] A.J. Baddeley and M.N.M. van Lieshout. Area-interaction point processes. *Ann. Inst. Statist. Math.*, 47:601–619, 1995.

[3] A.A. Borovkov and S.G. Foss. Stochastically recursive sequences and their generalizations. *Siberian Advances in Mathematics*, 2(1):16–81, 1992. Translated from PRIM.

[4] K. Burdzy and W.S. Kendall. Efficient Markovian couplings: examples and counterexamples. Research report 331, Department of Statistics, University of Warwick, 1998. Submitted for publication.

[5] D.S. Carter and P.M. Prenter. Exponential spaces and counting processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwe Gebiete*, 21:1–19, 1972.

[6] P. Clifford and G. Nicholls. Comparison of birth-and-death and Metropolis-Hastings Markov chain Monte Carlo for the Strauss process. `http://www.math.auckland.ac.nz/~nicholls`, 1994.

[7] L. Saloff Coste. Nash inequalities. In O.E. Barndorff Nielsen, W.S. Kendall, and M.N.M. van Lieshout, editors, *Stochastic Geometry: Likelihood and Computation*, pages 365–400, Boca Raton, 1999. Chapman and Hall/CRC.

[8] R.W.R. Darling. *Constructing nonhomeomorphic stochastic flows*, volume 376. American Mathematical Society, 1987.

[9] J. Fill. An interruptible algorithm for exact sampling via Markov Chains. *Annals of Applied Probability*, 8:131–162, 1998.

[10] J.A. Fill and M. Machida. Stochastic monotonicity and realizable monotonicity. preprint, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, 1998.

[11] S.G. Foss and R.L. Tweedie. Perfect simulation and backward coupling. *Stochastic Models*, 14:187–203, 1998.

[12] C.J. Geyer. Likelihood inference for spatial point processes. In O.E. Barndorff Nielsen, W.S. Kendall, and M.N.M. van Lieshout, editors, *Stochastic Geometry: Likelihood and Computation*, pages 79–140, Boca Raton, 1999. Chapman and Hall/CRC.

[13] C.J. Geyer and J. Møller. Simulation and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21:359–373, 1994.

[14] P.J. Green and D.J. Murdoch. Exact sampling for Bayesian inference: towards general purpose algorithms. In J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics 6*. Oxford University Press, 1999. Presented as an invited paper at the 6th Valencia International Meeting on Bayesian Statistics, Alcossebre, Spain, June 1998.

[15] O. Häggström and J.E. Steif. Propp-Wilson algorithms and finitary codings for high noise Markov random fields, 1999. Preprint.

[16] O. Häggström, M.N.M. van Lieshout, and J. Møller. Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. Technical Report R-96-2040, Aalborg Mathematics Department Research Report, 1996. To appear in *Bernoulli*.

[17] M.B. Hansen, J. Møller, and G. Nicholls. Perfect sampling for Bayesian inference. (In preparation).

[18] F.P. Kelly and B.D. Ripley. A note on Strauss's model for clustering. *Biometrika*, 63:357–360, 1976.

[19] W.S. Kendall. On some weighted Boolean models. In D. Jeulin, editor, *Advances in Theory and Applications of Random Sets*, pages 105–120, Singapore, 1997. World Scientific Publishing Company.

[20] W.S. Kendall. Perfect simulation for spatial point processes. In *Proc. ISI* 51^{st} *session, Istanbul (August 1997)*, volume 3, pages 163–166, 1997.

[21] W.S. Kendall. Perfect simulation for the area-interaction point process. In L. Accardi and C.C. Heyde, editors, *Probability Towards 2000*, pages 218–234, New York, 1998. Springer.

[22] W.S. Kendall and J. Møller. Perfect implementation of a Metropolis-Hastings simulation of Markov point processes. Research Report 348, Department of Statistics, University of Warwick, 1999.

[23] W.S. Kendall and E. Thönnes. Perfect simulation in stochastic geometry. Research report 323, Department of Statistics, University of Warwick, 1997. To appear in *J. Pattern Recognition*.

[24] W.S. Kendall, M.N.M. van Lieshout, and A.J. Baddeley. Quermass-interaction processes: Stability properties. *Advances in Applied Probability*, 1999. To appear.

[25] W. Klein. Potts-model formulation of continuum percolation. *Phys. Review B (3)*, 26:2677–2678, 1982.

[26] T. Lindvall. *Lectures on the Coupling Method*. John Wiley and Sons, Chichester, 1992.

[27] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Springer Verlag, New York, 1993.

[28] A. Mira, J. Møller, and G.O. Roberts. Perfect slice samplers. In preparation, 1999.

[29] J. Møller. On the rate of convergence of spatial birth-and-death processes. *Ann. Inst. Statist. Math.*, 41:565–581, 1989.

[30] J. Møller. Discussion contribution. *Scandinavian Journal of Statistics*, 21:346–349, 1994.

[31] J. Møller. Markov chain Monte Carlo and spatial point processes. In O.E. Barndorff Nielsen, W.S. Kendall, and M.N.M. van Lieshout, editors, *Stochastic Geometry: Likelihood and Computation*, pages 141–172, Boca Raton, 1999. Chapman and Hall/CRC.

[32] J. Møller and K. Schladitz. Extensions of Fill's algorithm for perfect simulation. *Journal of the Royal Statistical Society*, B 61, 1999. To appear.

[33] J. Møller and R. Waagepetersen. Markov connected component fields. *Advances in Applied Probability*, 30:1–35, 1998.

[34] D.J Murdoch and P.J. Green. Exact sampling from a continuous state space. *Scand. J. Stat.*, 25:483–502, 1998.

[35] D.J. Murdoch and J.S. Rosenthal. Efficient use of exact samples, 1999. To appear.

[36] C.J. Preston. Spatial birth-and-death processes. *Bull. Inst. Internat. Statist.*, 46 (2):371–391, 1977.

[37] J.G. Propp and D.B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[38] B.D. Ripley. Modelling spatial patterns (with discussion). *J. Roy. Statist. Soc.*, 39:172–212, 1977.

[39] B.D. Ripley and F.P. Kelly. Markov point processes. *Journal of the London Mathematical Society*, 15:188–192, 1977.

[40] D.J. Strauss. A model for clustering. *Biometrika*, 63:467–475, 1975.

[41] E. Thönnes. Perfect simulation of some point processes for the impatient user. *Advances in Applied Probability*, 1999. To appear.

[42] J. van den Berg and J.E. Steif. On the existence and non-existence of finitary codings for a class of random fields, 1999. To appear.