BAYESIAN STATISTICS 6, pp. 000–000
J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (Eds.)

Exact sampling for Bayesian inference: towards general purpose algorithms

PETER J. GREEN and DUNCAN J. MURDOCH
University of Bristol, UK and Queen's University, Canada

SUMMARY

There are now methods for organising a Markov chain Monte Carlo simulation so that it can be guaranteed that the state of the process at a given time is *exactly* drawn from the target distribution. The question of assessing convergence totally vanishes. Such methods are known as exact or perfect sampling. The approach that has received most attention uses the protocol of *coupling from the past* devised by Propp and Wilson (*Random Structures and Algorithms*,1996), in which multiple dependent paths of the chain are run from different initial states at a sequence of initial times going backwards into the past, until they satisfy the condition of coalescence by time 0. When this is achieved the state at time 0 is distributed according to the required target. This process must be implemented very carefully to assure its validity (including appropriate re-use of random number streams), and also requires one of various tricks to enable us to follow infinitely many sample paths with a finite amount of work.

With the ultimate objective of Bayesian MCMC with guaranteed convergence, the purpose of this paper is to describe recent efforts to construct exact sampling methods for continuous-state Markov chains. We review existing methods based on gamma-coupling and rejection sampling (Murdoch and Green, *Scandinavian Journal of Statistics*, 1998), that are quite straightforward to understand, but require a closed form for the transition kernel and entail cumbersome algebraic manipulation. We then introduce two new methods based on random walk Metropolis, that offer the prospect of more automatic use, not least because the difficult, continuous, part of the transition mechanism can be coupled in a generic way, using a proposal distribution of convenience.

One of the methods is based on a neat decomposition of any unimodal (multivariate) symmetric density into pieces that may be re-assembled to construct any translated copy of itself: that allows coupling of a continuum of Metropolis proposals to a finite set, at least for a compact state space. We discuss methods for economically coupling the subsequent accept/reject decisions.

Our second new method deals with unbounded state spaces, using a trick due to W. S. Kendall of running a coupled dominating process in parallel with the sample paths of interest. The random subset of the state space below the dominating path is compact, allowing efficient coupling and coalescence.

We look towards the possibility that application of such methods could become sufficiently convenient that they could become the basis for routine Bayesian computation in the foreseeable future.

Keywords: BISECTION COUPLER; COUPLING FROM THE PAST; EXACT SAMPLING; GAMMA COUPLING; PERFECT SAMPLING; REJECTION COUPLER.

1. INTRODUCTION

Markov chain Monte Carlo has made something of an impact on Bayesian analysis. Many computations that we need to do but which were essentially impossible using available numerical methods can now be done. In consequence, Bayesian statistics has become mainstream, not only for routine analysis, but a method of choice for challenging inferential problems with large-scale datasets arising in modern science and technology. In the process of this development, sample-based computation, by Markov chain simulation or otherwise, has become respectable: we are gradually realising the vast opportunities for subtle extraction of information from posterior distributions that are now available. Sample-based computation having become acceptable as a routine approach to Bayesian computation, two aspects of Markov chain sampling remain problematical: the issues of convergence — when can I assume that my Markov chain is sampling in equilibrium? — and dependence — how precise are estimates of probabilities and expectations obtained from the simulation? Arguably, the second of these is relatively easily handled, by one or other estimator of Monte Carlo variance, once convergence is assured.

There are now methods for organising a MCMC simulation so that it can be guaranteed that the state of the process at a given time is *exactly* drawn from the target distribution. The question of assessing convergence totally vanishes. Such methods are known as exact or perfect sampling.

The purpose of this paper is to describe recent efforts to construct exact sampling methods for continuous-state Markov chains, and to look towards the possibility that application of such methods could become sufficiently convenient that they could become the basis for routine Bayesian computation in the foreseeable future. We remain open-minded about this possibility, and certainly cannot yet demonstrate its probability, but we are optimistic.

1.1. *Coupling*

Exact sampling is a recent discovery. Two papers first circulated in 1995, by Propp and Wilson and by Fill, set out two different schemes, both based on ideas of coupling, for adapting MCMC simulation to the task of producing a guaranteed target draw. Propp and Wilson's proposal of *coupling from the past* is easier to implement and has received more attention, and we shall concentrate on it here.

Coupling has been a valuable tool in probability theory for some time, but its use in statistical simulation has been minimal. Coupling refers to the idea of constructing the random variables of interest on a common probability space, so that questions can be answered by probabilistic rather than analytic methods. There are, for example, particularly elegant proofs of Poisson approximation theorems, or of the ergodic theorem for Markov chains, based on coupling arguments (see Lindvall, 1992, for an enthusiastic course on this subject). In simulation, the 'common probability space' becomes concrete as the multiple use of sequences of random numbers in order to introduce dependence between sample paths. In making this correspondence, all we are doing is being literal about what random variables and stochastic processes are, and thinking of our simulation programs as defining appropriately measurable functions: it is a natural perspective for a probabilist, but perhaps novel for the practising statistician.

In Markov chain simulation, coupling allows us to create sample paths from many — perhaps infinitely many — initial states simultaneously. By reusing the same random numbers, these paths are made dependent, and by appropriate design can be made to satisfy desired conditions, such as having high probability that their paths merge, or *coalesce*. Of course, this constructed dependence does not infringe the Markov property, which only requires that we use different random numbers at different times within one path, and is silent about the same times on different paths.

The essence of the game will already be evident — coupling seems to offer the prospects of constructing multiple paths that ultimately coalesce, and hence 'forget' their initial conditions. Forgetting initial conditions is exactly what we want our MCMC simulations to do. Various lines of enquiry prior to Propp and Wilson were concerned with using these ideas to diagnose convergence (for example, Johnson, 1996, Asmussen *et al.* , 1992).

1.2. *Coupling from the past*

The leap made by Propp and Wilson was to note that coupling could be used more explicitly still, to guarantee rather than assess convergence. Informally, the idea is to conduct the simulation from all possible states at a random (but finite) time in the past, in such a way that all paths have coalesced into a single track by time 0, and the state occupied at time 0 is a draw from the target.

The structure of algorithms based on this idea is as follows. We consider runs of length M ending at time 0, for some arbitrary value of $M > 0$, starting at every possible state of the chain at time $t = -M$. Using a sequence of random numbers, we follow the paths from each of these states forward in time; occasionally paths will coalesce, and eventually all of the paths will have coalesced into one. If this has happened when we reach time 0, then we are done. If there are still multiple possible states, we choose a larger value of M and start again, using the *same realisation* of the sequence of random numbers.

To formalise this process, to see why it works, and to describe algorithms compactly, we introduce the *stochastic recursive sequence* representation of a Markov chain. Any Markov chain $\{X_t, t = \ldots, -1, 0, 1, 2, \ldots\}$ with state space χ can be represented in the way it is simulated:

$$X_{t+1} = \phi(X_t, U_{t+1}) \tag{3.4}$$

where $\{U_t, t = \ldots, -1, 0, 1, 2, \ldots\}$ is a sequence of independent random quantities from some fixed distribution, and ϕ is an appropriate deterministic *structure function*. For example, the simultaneous-update random walk Metropolis algorithm for an arbitrary target density π can be expressed via $\phi(X, U) = X + U^{(2)}$ if $U^{(1)} < \pi(X + U^{(2)})/\pi(X)$ or X otherwise, where $U^{(1)}$ is a $U(0, 1)$ random variable and $U^{(2)}$ is a random increment of the same dimension as X with a distribution symmetric about 0.

In terms of the structure function, coupling from the past (CFTP) can be expressed in pseudo-code as follows:

```
CFTP(M):
    t ← −M
    B_t ← χ
    while t < 0
        t ← t + 1
        B_t ← φ(B_{t−1}, U_t)
    if #B_0 = 1 then
        return(B_0)
    else
        CFTP(2M)
```

and is initiated by starting with an arbitrary positive integer M, often 1. We use B_t to denote the set of states occupied at time t by the paths under consideration. Coupling from the past is based on the observation that if the chain was (conceptually) run from all initial states in χ at time $-\infty$ ($B_{-\infty} = \chi$), and these paths coalesced to a singleton by time 0 ($B_0 = \{X_0\}$), then X_0 would be drawn from the stationary distribution π. This is Theorem 2 of Propp and Wilson (1996). For certain structure functions, if this occurred, then there would be a *last* time $-T$ before 0 such that the paths from $B_{-T} = \chi$ coalesced to a singleton. The `CFTP` algorithm listed above is a search for T. We do not need to find it exactly: since $B_t = \chi$ for all $t < -T$, it is sufficient to check whether $T \leq M$, for $M = 1, 2, 4, \ldots$.

The CFTP algorithm is only guaranteed to terminate when using such structure functions. Foss and Tweedie (1998) showed that this requires uniform ergodicity of the Markov chain.

However, variations on CFTP such as those described in Section 4 allow it to be applied in greater generality.

1.3. *Current implementations of exact sampling*

As in the early days of ordinary MCMC, most current applications of CFTP have been to exact sampling of large discrete physical systems and spatial processes. Many of these exploit a *monotonicity* property, or its opposite, anti-monotonicity. A monotonic MCMC sampler is one implemented by a structure function ϕ that is order-preserving with respect to some partial order $\leq$ on the state space: $x \leq y \Rightarrow \phi(x, u) \leq \phi(y, u)$ for all u, where $\leq$ has the property that there are unique minimal and maximal elements $\widehat{0}$ and $\widehat{1}$. In the case of anti-monotonicity the ranking is reversed: $\phi(x, u) \geq \phi(y, u)$. In either case, the progress forward of paths from $B_{-M} = \chi$ to $t = 0$ can be followed by tracing only the paths from $\widehat{0}$ and $\widehat{1}$.

Papers exploiting CFTP implemented in this way include Propp and Wilson's original (1996) paper, with applications to random dimer and cluster models. Haggstrom *et al.* (1996), and Haggstrom and Nelander (1997a,b) treat area-interaction point processes, Ising models with negative interaction, the random cluster model, and random q-colourings.

Kendall (1998) and Haggstrom *et al.* (1996) go beyond straight CFTP to perform exact sampling for a chain that is not uniformly ergodic; they use a coupled dominating process, and implement the method for various spatial processes. The idea of the dominating chain is borrowed for the methods we introduce in Section 4 of this paper.

Propp and Wilson (1998) derive 'generic' methods for exact sampling on Markov chains, that do not exploit any special features such as monotonicity to facilitate coalescence. Implementations are to random spanning trees and other graphical algorithms.

Bayesian uses for exact sampling have been so far rare. Fismen (1997), in a Master's thesis supervised by Havard Rue, applies both monotone CFTP and Fill's algorithm to idealised binary image analysis problems based on Ising model priors, and simple forms of degradation for which the likelihood does not destroy the monotonicity property. Murdoch and Green (1998) apply their methods (that do not use monotonicity, and are reviewed briefly in the next section) to a hierarchical gamma model and the well-known 'pump-reliability' data, and Fismen discusses how to adapt their methods to a hierarchical model for binomial data. Moller (1997) also addresses the pump data, using his approach to conditionally-specified models exploiting (anti-) monotonicity, that gives exact sampling up to an arbitrarily small numerical tolerance.

New work in the area of exact sampling is appearing almost weekly: the interested reader is directed to David Wilson's online annotated bibliography (`http://dimacs.rutgers.edu/~dbwilson/exact`) for the latest news.

2. EXACT SAMPLING IN CONTINUOUS STATE SPACES

When we started working on exact sampling, we were unsure that exact CFTP would ever be possible in continuous state spaces, and thought that some approximation to tolerance ϵ would be the best that could be achieved. Then we discovered several prototypical constructions that did allow it exactly, at the price generally of some tedious and cumbersome algebra; these were introduced and discussed in full in Murdoch and Green (1998). In sections 2.1 and 2.2 below we briefly describe the *multigamma* and *rejection* couplers.

In practice, in implementing CFTP, we have found it useful often to carry forward a larger but more tractable set of states than $\phi(B_{t-1}, U_t)$ — this will in general delay coalescence, but save computer time and storage. So we usually code the update as $B_t \leftarrow \Phi(B_{t-1}, U_t)$, where $\Phi(B, U)$ is any set containing $\{\phi(x, U), x \in B\}$. For example, we might simply take $\Phi(B, U) = \chi$ if B is infinite, or exactly $\{\phi(x, U), x \in B\}$ otherwise.

2.1. *Multigamma coupling*

Our first method applies when we know explicitly the update kernel density $f(\cdot \mid x)$ of the Markov chain, the density of X_{t+1} given that $X_t = x$. Furthermore, we suppose that f is bounded below uniformly in x: $f(y \mid x) \geq r(y) \forall x, y \in \chi$ for some nonnegative function $r(\cdot)$, for which $\rho = \int r(y)dy > 0$. Then we can express every $f(\cdot \mid x)$ as a fixed-weight mixture of two distributions, one of which does not depend on x. In the one dimensional case, for example, we can write

$$p(X_{t+1} \leq y \mid X_t = x) = \rho R(y) + (1-\rho) Q(y \mid x)$$

where

$$R(y) = \rho^{-1} \int_{-\infty}^{y} r(v)dv \quad \text{and} \quad Q(y \mid x) = (1-\rho)^{-1} \int_{-\infty}^{y} [f(v \mid x) - r(v)]dv.$$

Assuming the inverses of R and $Q(\cdot \mid x)$ are available, the update function to be used by the algorithm is

$$\phi(x, U) = \begin{cases} R^{-1}(U^{(2)}) & \text{if } U^{(1)} < \rho \\ Q^{-1}(U^{(2)} \mid x) & \text{otherwise}, \end{cases}$$

using a pair $U = (U^{(1)}, U^{(2)})$ of Uniform random numbers. Then it is easy to see that in the `CFTP` algorithm we can use

$$\Phi(B_{t-1}, U_t) = \begin{cases} R^{-1}(U^{(2)}) & \text{if } U_t^{(1)} < \rho \\ Q^{-1}(U^{(2)} \mid X_{t-1}) & \text{if } U_t^{(1)} \geq \rho \text{ and } B_{t-1} = \{X_{t-1}\} \\ \chi & \text{otherwise}, \end{cases}$$

and the algorithm evidently terminates. In fact, T has a geometric distribution with mean $1/\rho$.

While it is often difficult or impossible to find a uniform non-zero lower bound on the update densities, quite commonly the state space can be partitioned into a finite collection of disjoint cells $\mathcal{A} = \{A_i, i = 1, \ldots, m\}$, and separate lower bounds can be placed on the densities within each cell. That is,

$$f(y \mid x) \geq r_i(y) \quad \forall x \in A_i, \quad \forall y \in \chi.$$

We further assume that $\int r_i(y)dy = \rho > 0$ is constant over all cells; this assumption can be relaxed. Then a partitioned version of the multigamma coupler is available. It uses

$$\Phi(B_{t-1}, U_t) = \begin{cases} \bigcup_{i: A_i \cap B_{t-1} \neq \emptyset} \{R_i^{-1}(U_t^{(2)})\} & \text{if } U_t^{(1)} < \rho \\ \bigcup_i \{Q_i^{-1}(U_t^{(2)} \mid x) : x \in A_i \cap B_{t-1}\} & \text{if } U_t^{(1)} \geq \rho \text{ and } \#B_{t-1} < \infty \\ \chi & \text{otherwise}. \end{cases}$$

2.2. *Rejection coupling*

This is an alternative basic recipe for coupling from the past, with the attraction that it can apply where the updating densities are known only up to a multiplicative constant, as is usual in Bayesian MCMC.

We suppose that the conditional update kernel is $f(y \mid x) = k(x)g(y \mid x)$ where the unnormalised density $g(y \mid x)$ is known, but $k(x)$ is unknown, and that there exists an envelope function $h(y)$ such that $g(y \mid x) \leq h(y)$ for all $x, y \in \chi$. We assume $\nu = \int h(y)dy < \infty$, which is quite a demanding condition in an unbounded state space. Finally, we assume that we have a way of sampling from the density $h(y)/\nu$.

The idea is a simple adaptation of rejection sampling: we sample repeatedly under the graph of $h(y)$, yielding i.i.d. points $(Y^{(j)}, V^{(j)}h(Y^{(j)}))$ with $V^{(j)} \sim U(0,1)$ independent of $Y^{(j)} \sim h(\cdot)/\nu$. Let $A_j = \{x \in B_{t-1} : g(Y^{(j)}) \geq V^{(j)}h(Y^{(j)})\}$: these are the values of x such that $X_t = Y^{(j)}$ is a valid update from $X_{t-1} = x$. Continue to the smallest J such that $\cup_{j=1}^{J} A_j \supset B_{t-1}$. Then use $\Phi(B_{t-1}, U_t) = \{Y^{(j)}, j = 1, 2, \ldots J\}$. (Note that each U_t is a sequence of pairs $(Y^{(j)}, V^{(j)}), j = 1, 2, \ldots$.)

In practice, rather than accumulate the union of the A_j, we check the coverage of χ using a lower squeezing function. And again, partitioning of χ is possible, and is useful in speeding up coalescence.

3. TOWARDS AUTOMATIC METHODS

The algorithms of the previous section demonstrate the possibility of exact sampling in continuous state spaces, and the examples in Murdoch and Green (1998) show that it works in practice, on toy examples, and for a Gibbs sampler applied to the 'pump reliability' data. However, they are expensive to use in terms of algebraic and programming effort. In this section we turn to methods that seem to us to offer a real prospect of useful algorithms for general purposes.

3.1. *Random walk Metropolis*

Random walk Metropolis is a 'vanilla' MCMC method for Bayesian computation, being very easy to code, and usually offering reasonable performance at the expense of a few pilot runs to choose proposal spread parameters.

It is promising for adaptation to CFTP since we can separately couple the proposals and the acceptance decisions; the difficult continuous part can be done once and for all, in a model-independent way, for a proposal distribution chosen at our convenience.

Thus the general objective for a given symmetric proposal density q is to first find a random walk update function ψ such that $\psi(x, U) \sim q(\cdot - x)$ and such that $\psi(B, U)$ is finite with high probability, and then use

$$\phi(B, U) = \bigcup_{x \in A(U)} \psi(x, U^{(2)}) \cup (B \setminus A(U))$$

where

$$A(U) = \{x \in B : U^{(1)} < \pi(\psi(x, U^{(2)}))/\pi(x)\}$$

is the set of states for which the proposed update is accepted. In practice we work with a $\Phi(B, U)$ for which both terms of the union may be 'rounded up' for convenience.

In the next subsection, we discuss a method for coupling the proposals, and in subsection 3.3 show how the acceptance/rejection decisions might be automated.

3.2. *The bisection coupler*

The bisection coupler is a general purpose random walk coupler for a state space that is a rectangle in any number of dimensions. Suppose first that $\chi = [0, 1]$, and that $q(\cdot)$ is unimodal and symmetric. Let

$$r_{xy}(\cdot) = \min\{q(\cdot - x), q(\cdot - y)\}$$

denote the intersection between two shifted copies of q centered at x and y. Then

1. $r_{xy}(\cdot) = \min_{z\in[x,y]} q(\cdot - z)$
2. $r_{xy}(\cdot) = r_{0,y-x}(\cdot - x)$;
3. $r_{0y}(\cdot) \nearrow q(\cdot)$ as $y \searrow 0$;
4. $r_{xy}(\cdot)$ is symmetric about $(x + y)/2$.

These properties allow a very nice specialisation of the partitioned multigamma coupler. It turns out that we can dissect the area under $q(\cdot)$ into pieces according to the overlaps with $q(\cdot - 2^{-N})$, $N = 0, \ldots$, and then represent $q(\cdot - x)$, for any $x \in [0, 1]$, by rearranging these pieces. The piece created by the N^{th} intersection, $N = 0, 1, 2, \ldots$, occurs in exactly 2^N places over all these decompositions. This is illustrated in Figure 1.

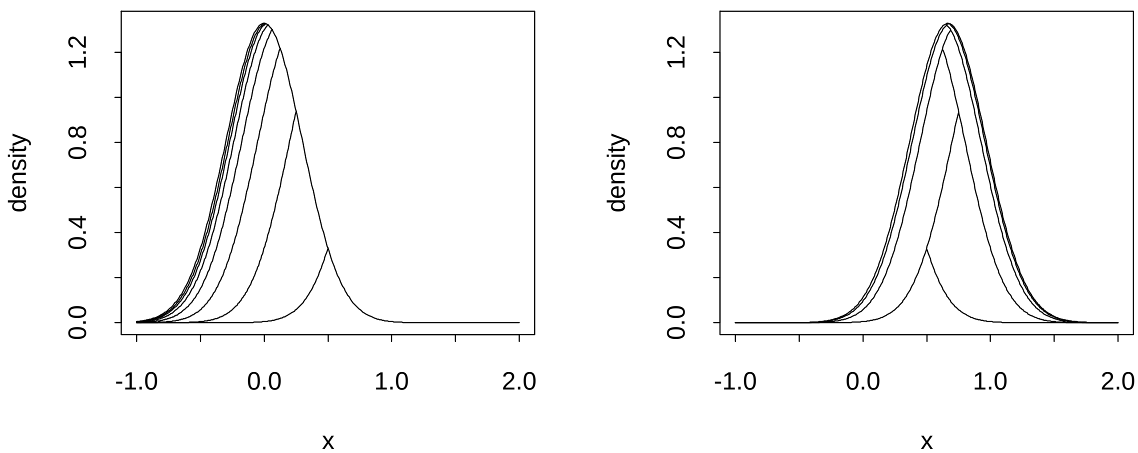

Figure 1. *Illustrating the bisection coupler: the decompositions of the* $N(0, 0.3^2)$ *and* $N(2/3, 0.3^2)$ *densities exploited by the algorithm.*

The update $\psi(B, U)$ then consists of sampling one point under the graph of q using U, finding the abscissa in the corresponding piece under $q(\cdot - x)$ for all $x \in B$ by reflections and translations, and setting $\psi(B, U)$ to be the resulting set of abscissae. There will be at most 2^N values in this set (maybe fewer as we do not need to store values outside χ, or moves from outside B), where N indexes the piece into which the random point fell.

We can write all of this as $\psi(x, U) = 2k/2^N - Y$ for $x \in [(2k - 1)/2^N, 2k/2^N)$ and $\psi(x, U) = 2k/2^N + Y$ for $x \in [2k/2^N, (2k + 1)/2^N)$. Here $U = (V, Y)$, where $V \sim U(0, 1)$ and $Y \sim q(\cdot)$, independently, N is the smallest integer such that $V q(Y) < q(Y - 2^{-N})$, and k is an integer in the range $0, \ldots, 2^{N-1}$. While N is almost surely finite, and has finite expectation, we find that, when $q(\cdot)$ is continuous at the origin, for large n we have $P(N > n) \approx 2^{-n} q(0)$ so that $E(2^N) = \infty$: the set $\psi(B, U)$ is always finite but can be large. We adopt our usual trick of storing only an interval superset if the list is too long.

If we allow N and k to be any integer values, the bisection coupler may be used on the whole real line. Provided B is bounded, $\psi(B, U)$ will be finite. To allow positive and negative values of x to coalesce, we give a random uniform shift to the coordinate system before partitioning takes place.

Bisection coupling easily generalizes to finite dimensional χ, provided $q(\cdot)$ is unimodal and symmetric about 0 in each of its coordinates: we replace the sequence $2^{-N}, N = \ldots, 0, 1, \ldots$ by a sequence of vector displacements in which each coordinate is bisected in turn.

3.3. *Automatic cell bounds*

For a real impact on the practice of Bayesian computation, it will be necessary for exact simulation to be much more routinely implemented. What are the prospects for automated coupling of the accept/reject decisions?

A minimal general class of models of interest would be the Bayesian hierarchical models based on 'standard' distributions, and built on directed acyclic graphs, such as those implemented in `BUGS` (Gilks, *et al.* , 1994). Such graphs represent conditional independence statements: given the values $x_{pa(v)}$ of its parents, each variable x_v is conditionally independent of all non-descendants, giving the factorisation:

$$\pi(x) = \prod_{v \in V} \pi(x_v \mid x_{pa(v)}).$$

In turn, this implies a 'near-separability' property that we can exploit. Taking logarithms, we can bound the log-density in arbitrary cells of the parameter space:

$$\sup_{x \in A} \log \pi(x) \leq \sum_{v \in V} \sup_{x \in A} \log \pi(x_v \mid x_{pa(v)})$$

$$\inf_{x \in A} \log \pi(x) \geq \sum_{v \in V} \inf_{x \in A} \log \pi(x_v \mid x_{pa(v)}),$$

and these bounds should not be too slack: in typical models, each variable x_v appears in only a few terms.

For rectangular cells A, and 'standard' distributions, both $\sup_{x \in A} \log \pi(x_v \mid x_{pa(v)})$ and $\inf_{x \in A} \log \pi(x_v \mid x_{pa(v)})$ are explicitly available, typically requiring some tedious one-off calculus on 2 or 3 variables. The bounds apply equally to posterior distributions: observed variables are fixed, and we can ignore the fact that the whole joint density should be re-normalised.

The bounds provide the raw ingredients for partitioned random walk Metropolis coupling, permitting economical checking of whether a proposed update is to be accepted. For example, suppose $A \cap B_{t-1}$ is proposed to be updated to $\{Y_j : j = 1, 2, \ldots, J\} = \psi(A \cap B_{t-1}, U_t^{(2)})$. Then if

$$U_t^{(1)} < \frac{\pi(Y_j)}{\inf_{x \in A \cap B_{t-1}} \pi(x)}$$

we include Y_j in B_t, and if

$$U_t^{(1)} > \frac{\pi(Y_j)}{\sup_{x \in A \cap B_{t-1}} \pi(x)}$$

for any j, we include all of $A \cap B_{t-1}$ in B_t. The set B_t assembled in this way certainly includes $\phi(x, U_t)$ for all $x \in B_{t-1}$.

3.4. *Example: Dirichlet priors and multinomial data*

To avoid problems with unbounded state spaces, we will illustrate the working of the bisection coupler on various target distributions with state space $[0,1]^d$, using exact bounds on the target density in each cell.

We begin with a one-dimensional example, with target Beta$(25,75)$. Figure 2 shows a single run of the random walk Metropolis bisection coupler, using 16 equal-sized cells, and a N$(0,0.3^2)$ proposal distribution. Coalescence to a single state (the value 0.2589) was achieved with $M = 16$. The figure should be self-explanatory: grey rectangles show where a proposal was rejected for at least one state in a cell, triangles where such a proposal was accepted, and lines denote simple state-to-state updates after the 'live' states in a cell have been reduced to a finite list; solid lines are used for the successful pass with $M = 16$. In Figure 3, we show a histogram and PP-plot for the final states from 1000 independent replicates of this coupler. The P-value for a Kolmogorov-Smirnov goodness of fit test to the specified target is 0.359. The frequency distribution for M over the 1000 replicates was $(49,74,166,280,273,140,18)$ for the values $(4,8,16,32,64,128,256)$.

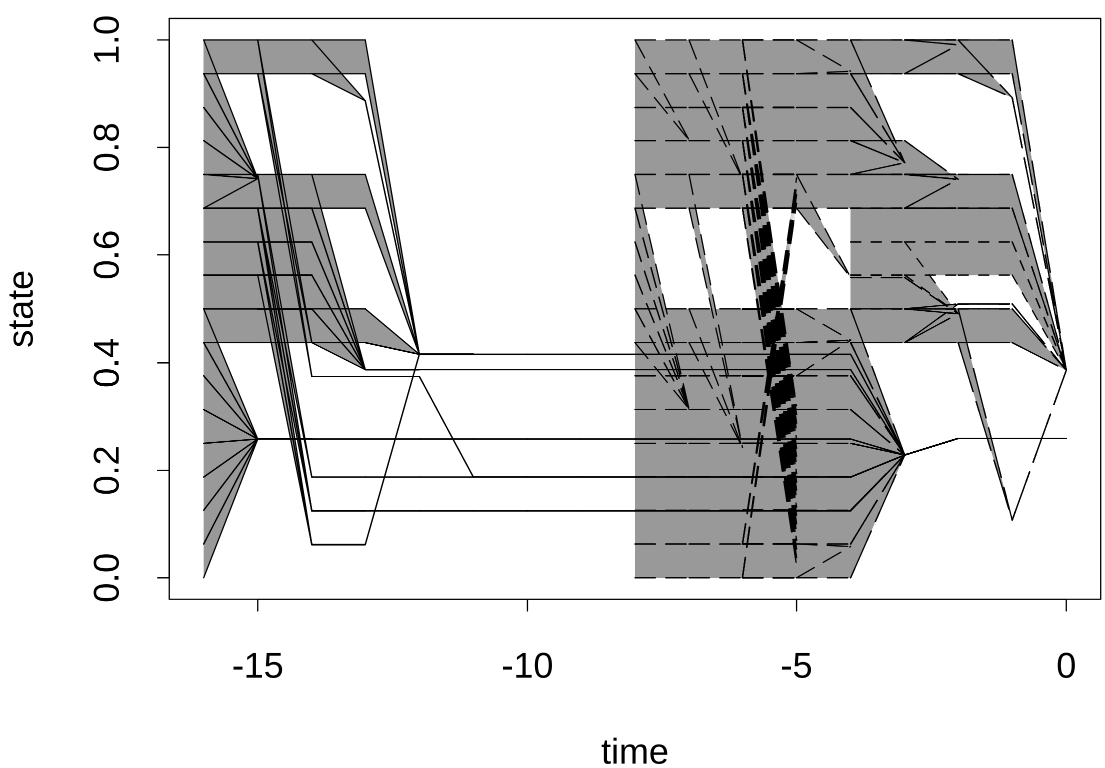

Figure 2. *A single run of the random walk Metropolis bisection coupler with a Beta$(25,75)$ target.*

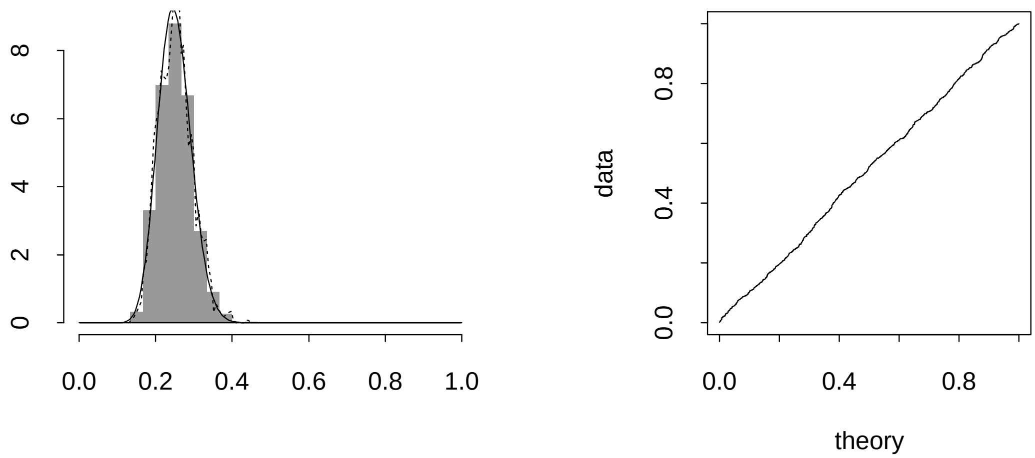

Figure 3. *Histogram and PP-plot of 1000 replicates of the random walk Metropolis bisection coupler with a Beta*$(25, 75)$ *target.*

Now we turn to a small Bayesian example. In the ABO blood group system, the alleles A and B are co-dominant to O, so under Hardy-Weinberg equilibrium, the phenotypes (A,B,AB,O) occur with probabilities $(p^2 + 2pr, q^2 + 2qr, 2pq, r^2)$ when the gene frequencies for (A,B,O) are (p, q, r) $(p + q + r = 1)$. Given observed phenotype frequencies (9,3,1,10) respectively, we compute the posterior for (p, q, r) under a Dirichlet$(1, 1, 1)$ prior. Note that although we assume multinomial sampling, this is not a conjugate set-up. Figure 4 shows a barycentric plot for the final states from 1000 replicates of a random walk Metropolis bisection coupler for this target. We used 64 equal-sized cells in the (p, q) plane, and the proposal distribution was bivariate circular normal, with standard deviation 0.2.

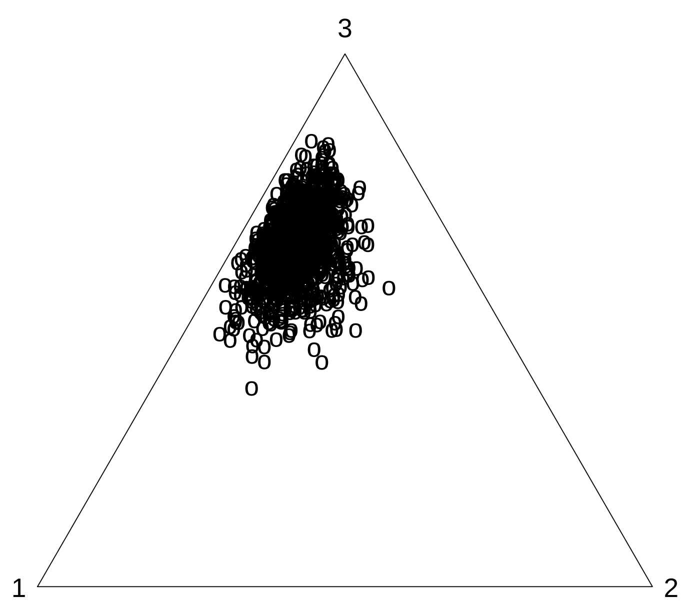

Figure 4. *Barycentric scatter plot of 1000 replicates of the random walk Metropolis bisection coupler for the posterior in the small ABO dataset.*

Figure 5 shows results from a similar analysis, but based on the larger data set (179,35,6,202). $\blacksquare$

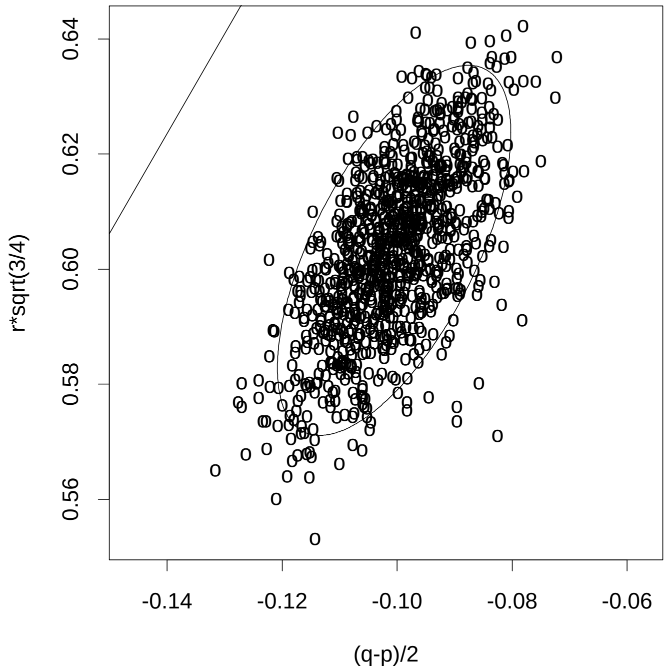

Figure 5. *Magnified view of part of barycentric scatter plot of 1000 replicates of the random walk Metropolis bisection coupler for the posterior in the large ABO dataset, with an elliptical 90% credible region.*

Finally, in Figure 6, we show PP-plots of the results of 100 replicates of the coupler for a Dirichlet$(4, 4, 4, 4)$ target: this was implemented using 64 cells in the 3-dimensional parameter space, with $N(0, 0.5^2)$ proposals.

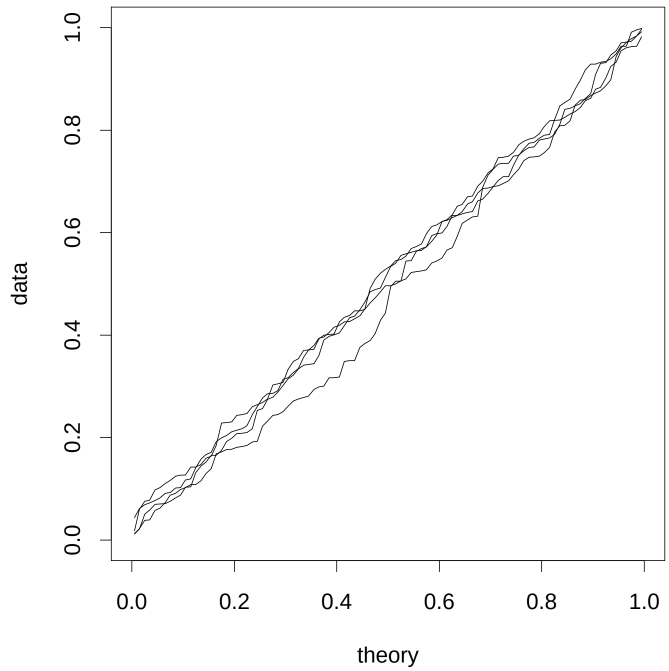

Figure 6. *Superimposed PP-plots of each margin of the output of a random walk Metropolis bisection coupler for a Dirichlet*$(4, 4, 4, 4)$ *target, based on 100 replicates.*

4. DOMINATED CFTP

4.1. *Random bounds on couplers*

Kendall (1998) described a trick which allows CFTP to be used on Markov chains which are not uniformly ergodic. The idea is based on the fact that a stationary realization of the chain will tend to stay in regions of high density. We quantify this by constructing a coupled *dominating* chain which has a stationary distribution with heavier tails than the target, and bound the possible values of the target by the (more easily simulated) values of the dominating chain, thereby reducing the state space to a manageable size.

To make this concrete, consider a one dimensional real-valued chain X_t taking positive values. We construct a dominating chain C_t with the following properties:

Coupling: Both chains are updated together, using the same random inputs. The updating rule can be written as $(X_{t+1}, C_{t+1}) = [\phi_1(X_t, U_{t+1}), \phi_2(C_t, U_{t+1})]$.
Dominance: If $x \leq c$ then $\phi_1(x, u) \leq \phi_2(c, u)$ for all u.
Convenience: C_t has a known limiting distribution from which we can sample directly.
Practical reversibility: We know how to run the bounding chain in reverse, i.e. when the dominating chain is in equilibrium, we know the conditional distribution of (C_{t-1}, U_t) given C_t and can sample from it.

In addition to these conditions, we need to ensure that the chain X_t eventually falls below C_t regardless of its starting value. For the cases of interest to us, coupled X_t chains will eventually coalesce, so this is automatic.

We then proceed as follows. To obtain the set B_{-M} of possible values of X_t at time $t = -M$, we sample C_0 from its stationary distribution, and then generate pairs (C_{t-1}, U_t),

$t = 0, \ldots, -M+1$. Since we know that any X_t path, if run from far enough back in time, would be caught below C_t by time $-M$, we set B_{-M} equal to $[0, C_{-M}]$. Using ϕ_1 and the just-generated U_t values, we carry it forward to time 0 for the usual CFTP test of coalescence.

The beauty of this dominated CFTP algorithm is that it does not require uniform ergodicity of the Markov chain on the whole state space. We only need to coalesce values on the random sets below C_t. Most chains that would arise in practice are uniformly ergodic on bounded sets, so coalescence may be achievable in practice.

4.2. *Applying dominated CFTP to random walk Metropolis*

Random walk Metropolis chains on unbounded state spaces are not uniformly ergodic, so standard CFTP will not work for them. Here we describe some details of implementing dominated CFTP for these chains.

Suppose that our target distribution has density proportional to $\pi(\cdot)$, and that π has positive support. Furthermore, π has a right tail that is stochastically dominated by a known distribution; for illustration we'll assume an exponential tail. Specifically, there exist constants $C > 0$ and $D > 0$ such that for $0 < d < D$, $C + D \leq x_1$, and $C \leq x \leq x_1$,

$$\frac{\pi(x+d)}{\pi(x)} \leq \frac{\pi_c(x_1+d)}{\pi_c(x_1)}$$

where $\pi_c(x) = \exp(-x)$ for $x > C + D$ is proportional to the density of an Exponential(1) random variable shifted $C + D$ units to the right.

What this condition is designed to achieve is the following. We replace U_t in the previous section with the pair (D_t, U_t), where $D_t \sim U(-D, D)$ and $U_t \sim U(0,1)$ are independent. Then random walk Metropolis for the target process may be written as $X_{t+1} = \phi_1(X_t, D_{t+1}, U_{t+1})$ where $\phi_1(x, d, u) = x + d$ if $\pi(x+d)/\pi(x) > u$ and x otherwise. Similarly, we implement the coupled dominating process as $C_{t+1} = \phi_2(C_t, D_{t+1}, U_{t+1})$, where we accept a move if $\pi_c(C_t + D_{t+1})/\pi_c(C_t) > U_{t+1}$.

With these definitions, the required ordering of X_t and C_t holds. Provided $X_t < C_t$, if X_t accepts a move upwards, so will C_t (unless X_t is so far below C_t that the paths could not cross), and similarly moves of C_t downwards imply corresponding moves of (close enough) X_t values. The other requirements of dominated CFTP are also met. In particular, to simulate the dominating chain backwards, we sample $\tilde{U} \sim U(0,1)$ and $\tilde{D} \sim U(-D, D)$; then if $\tilde{U} > \exp(\tilde{D})$ or $C_{t+1} - \tilde{D} < C + D$, we set $C_t = C_{t+1}$, $D_t = -\tilde{D}$ and $U_t = \tilde{U}$, and otherwise set $C_t = C_{t+1} - \tilde{D}$, $D_t = \tilde{D}$ and $U_t = \tilde{U}/\exp(\tilde{D})$.

One problem with this formulation is that it couples chains by proposing the same D_t value for every state. But this coupler does not coalesce, so the dominated CFTP algorithm would never terminate. We need to incorporate a coalescing update rule, such as the bisection coupler, but the bisection coupler generally will not respect the ordering: it uses different steps D_t from different states, and often leads to crossing paths.

A solution to this problem is to use a method like that of Corcoran and Tweedie (1998): construct an update formula which coalesces for some states but not all. They used a multigamma coupler for central states and achieved global monotonicity, and we could do the same. However, global monotonicity is not needed: provided $X_t \leq C$ and $D_t \in [-D, D]$, we do not care whether $\phi_1(X_t, D_{t+1}, U_{t+1})$ is monotonic. We only need to preserve the condition $X_{t+1} \leq C_{t+1}$, and it is automatically preserved by the bound on D_t. Thus we can use the bisection coupler on the states below C, and use common D_t proposals on other states.

4.3. *Extensions to more general targets*

The extension of dominated CFTP to two-tailed distributions is straightforward. We can put bounding processes on both the upper and lower tails, i.e. define coupled processes C_t^0, X_t and C_t^1 by $(C_{t+1}^0, X_{t+1}, C_{t+1}^1) = (\phi_0(C_t^0, U_{t+1}), \phi_1(X_t, U_{t+1}), \phi_2(C_t^1, U_{t+1}))$ such that if $c_0 \leq x \leq c_1$ then $\phi_0(c_0, u) \leq \phi_1(x, u) \leq \phi_1(c_1, u)$ for all u. The only new twist is that we need to sample from the joint limiting distribution of the pair of bounds at time 0, and update both backwards.

However, if we can use the same bounds on each tail then this is very easy. Provided we choose our random walk coupler to make jump proposals $-D_t$ for states less than $-C$ when it makes proposals D_t for states greater than C, we will have an equilibrium distribution with $C_t^0 = -C_t^1$ and only C_t^1 need be simulated.

The extension to p-dimensional processes is not so easy. If we could put separate bounds on each coordinate and these bounds applied uniformly across the values of the other coordinates, then we could bound each coordinate independently and we would not need anything new to sample at time 0 or to evolve the bounds backwards in time. However, most interesting p-dimensional distributions do not allow for uniform bounds. For instance, among the multivariate normals, uniform bounds imply zero correlation between the coordinates: not a very interesting case! It may turn out that we can transform the space to allow for uniform bounds as we can in the multivariate normal case, but it remains to be seen whether these transformations will be widely available.

4.4 *Example: dominated CFTP for a univariate case*

To illustrate the dominated CFTP algorithm, we use it to simulate a value from a standard Gaussian distribution. Ratios of the tails of the target density are bounded by ratios of e^{-x} for $x > C = 1$; we chose to use $D_t \sim U(-2, 2)$ random step proposals. This gave an Exponential(1), shifted 3 units to the right, as the upper bound, and its negative as the lower bound. We used the modified bisection coupler on the range $[-1, 1]$ to induce coalescence in the target paths.

We made one other modification to the algorithm. Instead of testing whether the whole interval $[-C_{-M}, C_{-M}]$ coalesces, we do a first pass test of just the two points $-C_{-M}$ and C_{-M}. If those two do not coalesce, it is unnecessary to test any others. Simulating just two paths makes the calculations much faster than simulating all paths.

A single run of this coupler is illustrated in Figure 7. It may be seen that in this run, the bounding points first coalesced when $M = 8$. However, when the full interval was run, there were still 4 possible values of the target process at time 0, so $M = 16$ was attempted. All values from that run coalesced at time $t = -12$; this path was carried forward to time 0, where it yielded the simulated $N(0, 1)$ value -0.755.

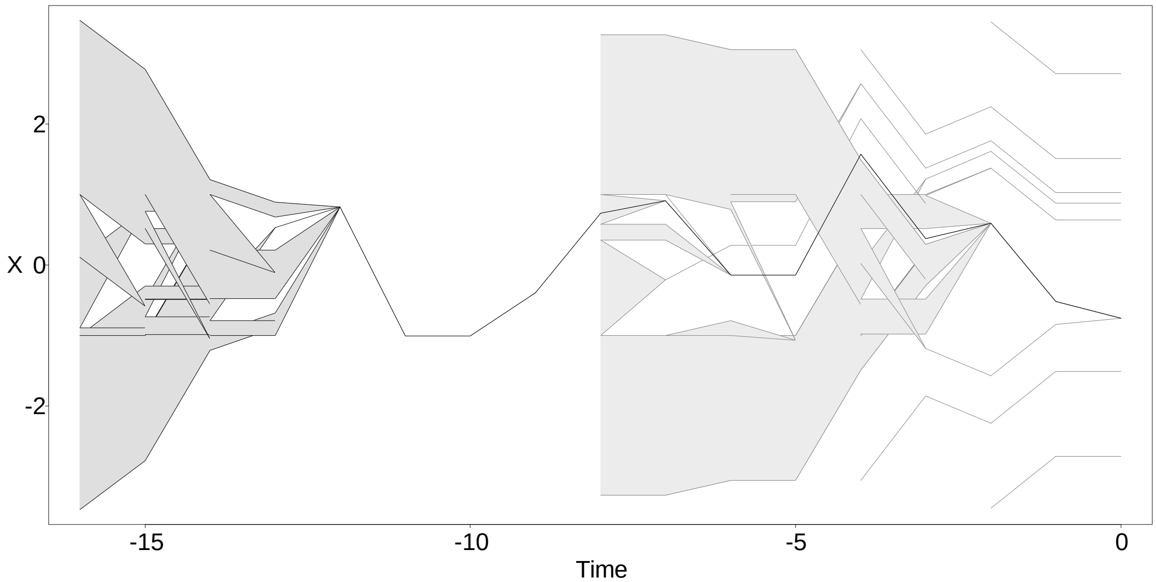

Figure 7. *A single run of the dominated CFTP algorithm with a $N(0, 1)$ target.*

5. CONCLUSIONS

At the start of this paper we expressed optimism about the eventual routine adoption of exact sampling methods in Bayesian computation, and we hope we have communicated some of this to the reader. In this section we will outline some of the remaining work to be done.

To date, exact sampling has been most spectacularly successful in its applications to models of very particular structure in many dimensions, where convergence diagnostics for MCMC methods are unavailable or unreliable. On the other hand, the methods for rather general models described in this paper work best in few dimensions: even the three dimensional Dirichlet example described in section 3.4 was a stretch, taking much longer to run than a standard simulation from a Dirichlet distribution. To merit routine use, methods are needed that work in the middle case, from a few to a few dozen dimensions.

Efficiency of use of the calculations is an important issue. A great deal of effort goes into producing just one sample from the target distribution. In some of the statistical physics and spatial process applications of exact sampling, one sample is already highly informative due to spatial ergodicity, but that perspective is of no use in Bayesian statistics. We have to decide how to use the opportunity to draw an exact sample in the MCMC context: do we make one (expensive) exact draw as a starting point for one long MCMC run, accepting that our ultimate sample will be dependent but is guaranteed to have the correct distribution? Or do we repeatedly draw exact samples, avoid ordinary MCMC and escape dependence as well as convergence detection? Presumably, the optimal choice will lie somewhere in the middle, and will depend on the mixing time of the chain, the theoretical coupling time of the structure function ϕ, any inefficiency in computer time in replacing ϕ by a convenient Φ, and the cost of evaluating the functionals of interest from the generated states.

If the time spent on exact draws is an appreciable fraction of the total simulation time, then surely better use of the information gleaned during the search for coalescence could be made than to throw it all away; it must be of some value, at least in tuning the CFTP implementation, and possibly more directly. If such information can be exploited, then it will certainly make the effort of exact sampling more attractive.

There is a certain aesthetic appeal to truly exact sampling, but when these methods are not feasible, simulations could still be improved by using ideas from CFTP. For example, Johnson's (1996) idea of coupling a finite set of paths forward in time to identify burn-in times might be improved if it were used to specify the initial set B_{-M} in a CFTP algorithm: then the bias induced by stopping based on a coalescence time would be removed.

ACKNOWLEDGEMENTS

We are grateful to Laird Breyer and Jeffrey Rosenthal for helpful discussions about bounded CFTP. This research was supported in part by EPSRC and NSERC Research Grants to the respective authors.

REFERENCES

Asmussen, S., Glynn, P. W. and Thorisson, H. (1992). Stationary detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation*, **2**,130–157.

Corcoran, J. N. and Tweedie, R. L. (1998). Perfect sampling of Harris recurrent Markov chains. Unpublished.

Fill, J. A. (1998) An interruptible algorithm for exact sampling via Markov Chains. *Ann. Appl. Prob.* (to appear).

Fismen, M. (1997). Exact simulation using Markov chains. *Tech. Rep.* **6/98**, Department of Mathematics, Norwegian University of Science and Technology. (available via WWW at `http://www.math.ntnu.no/preprint/statistics/1998/`)

Foss, S. G. and Tweedie, R. L. (1998). Perfect simulation and backward coupling. *Stochastic Models* (to appear).

Gilks, W. R., Thomas, A. and Spiegelhalter, D. J. (1994). A language and program for complex Bayesian modelling. *The Statistician* **43**, 169–178.

Haggstrom, O., van Lieshout, M. N. M., and Moller, J. (1996). Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Tech. Rep.* **R-96-2040**, Department of Mathematics, Aalborg University. *Bernoulli* (to appear).

Haggstrom, O. and Nelander, K. (1997a). Exact sampling from anti-monotone systems. *Statistica Neerlandica* (to appear).

Haggstrom, O. and Nelander, K. (1997b). On exact simulation of Markov random fields using coupling from the past. *Scandinavian J. Statist.* (to appear).

Johnson, V. E. (1996). Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *J. Amer. Statist. Assoc.* **91**, 154–166.

Kendall, W. (1998). Perfect simulation for the area-interaction point process. In *Probability towards 2000*, L. Accardi and C. C. Heyde, eds., pp. 218–234. New York: Springer.

Lindvall, T. (1992). *Lectures on the Coupling Method*. Chichester: Wiley.

Moller, J. (1997). Perfect simulation of conditionally specified models. *Tech. Rep.* **R-97-2006**, Department of Mathematics, Aalborg University. *J. Roy. Statist. Soc. B* (to appear).

Murdoch, D. J. and Green, P. J. (1998) Exact sampling from a continuous state space. *Scandinavian J. Statist.* (to appear).

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.

Propp, J. G. and Wilson, D. B. (1998). How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms* **27**, 170–217.