Perfect Forward Simulation via Simulated Tempering

Stephen P. Brooks†

University of Cambridge, U.K.

Yanan Fan

University of New South Wales, Australia

Jeffrey S. Rosenthal

University of Toronto, Canada

Summary. Several authors discuss how the simulated tempering scheme provides a very simple mechanism for introducing regenerations within a Markov chain. In this paper we explain how regenerative simulated tempering schemes provide a very natural mechanism for perfect simulation. We use this to provide a perfect simulation algorithm, which uses a single-sweep forward-simulation without the need for recursively searching through negative times. We demonstrate this algorithm in the context of several examples.

Keywords: Small sets; Regeneration; Coupling from the past; Reversible jump MCMC; Band-return data; Autoregressive time series

AMS Classification Numbers: primary 62F99, secondary 68W20.

Running Head: Perfect Forward Simulation

1. Introduction

MCMC methods have enjoyed wide-ranging interest in an enormous variety of applications over the past few years (Brooks 1998). It is well known that multi-modal target distributions pose particular problems for the MCMC method (much as they do for the numerical optimisation routines required for the corresponding maximum likelihood calculations) in that it is difficult to devise a transition rule which is able to efficiently explore both between and within modes. Partially to deal with these problems, various modifications of the basic MCMC algorithms have been proposed, such as the Langevin algorithm (Roberts and Tweedie 1996; Roberts and Rosenthal 1996; Neal 1993), the slice sampler (Neal 2000; Roberts and Rosenthal 2002a); simulated tempering (Marinari and Parisi 1992;

†*Address for correspondence:* Steve Brooks, Statistical Laboratory, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WB, U.K.

E-mail: steve@statslab.cam.ac.uk

Geyer and Thompson 1995; Liu and Sabatti 1999), and transdimensional MCMC methods (Green 1995; Brooks *et al* 2003).

In this paper, we examine the applications of regenerative simulated tempering schemes. We begin, in Section 2, by introducing the concepts behind simulated annealing and the "coupling from the past" (CFTP; Propp and Wilson 1996) perfect simulation scheme, as well as introducing the notions of small sets, coupling and regeneration. In Section 3 we revisit the idea of small sets within the context of the general simulated tempering scheme and describe two separate regenerative schemes. In Section 4.1 we discuss how separate tempering simulations can be coupled, through a particular reformulation of the simulated tempering algorithm. In Section 2.4 we discuss a very general result which can be applied beyond the tempering scheme developed here and which allows us to reformulate the usual CFTP scheme as a single-sweep forward-time simulation algorithm. This reduces somewhat the complexity of the CFTP scheme, in that we no longer need to repeatedly simulate the Markov chain forwards from increasingly further back in time, while keeping track of the updates from previous runs.

In Section 4.2 we construct a generic perfect simulation scheme based upon the CFTP scheme and using the regenerative properties of a simulated tempering scheme. The scheme is similar to that of Møller and Nicholls (1999) who also use simulated tempering to develop a perfect simulation algorithm. However, there are some significant points of departure between our two approaches. Firstly, we show how the perfect simulation scheme can be applied when the hot distribution is not an atom (which it always is in Møller and Nicholls 1999). In particular, we describe a new set of distributions which we call *i.i.d.-like* and explain how such distributions commonly arise in the context of simulated tempering schemes. Second, we ensure that our backward-coalescence time T will have a known (geometric) distribution, so that we can sample it directly, rather than repeatedly searching larger and larger values of T. This simplifies the algorithm, and somewhat improves its computational efficiency.

In Section 5 we extend all of these ideas to the trans-dimensional context by examining the strong links between the simulated tempering and reversible jump MCMC schemes. We illustrate these ideas in the context of several examples, including a simple example in Section 6.1, a trans-dimensional auto-regressive example in Section 5, and a real data analysis in Section 6.2. We close (Section 7) with some general discussion of the implications and general context of these ideas.

2. Background

Our approach will use ideas from simulated tempering, from coupling from the past (CFTP), and from the theory of small sets. Thus, we review these ideas here.

2.1. Simulated tempering

Metropolis-Hastings algorithms can become "stuck" in local modes (see e.g. Chib and Greenberg 1995). To avoid this problem, MCMC methods based upon the introduction of transition kernels with "flatter" stationary distributions have been proposed. One such method is that of Simulated tempering (Marinari and Parisi 1992; Geyer and Thompson 1995; Liu and Sabatti 1999) which uses a series of transition kernels $\mathcal{K}_\tau$ for $\tau \in \mathcal{T}$, with corresponding stationary densities $\pi_\tau(\boldsymbol{x})$ which link up two extremes: the "cold" distribution π_1 which is the distribution of interest, and a "hot" distribution π_T, where rapid mixing takes place. Typically, inference is then based upon observations which were drawn from the cold distribution, and the remaining observations are discarded.

We begin with a state space $\mathcal{X}$ which is an open subset of $\mathbb{R}^d$, with associated Borel σ-algebra $\mathcal{F}$. We take the set of "temperatures" $\mathcal{T}$ to be a finite set of integers, $\mathcal{T} = \{1, ..., T\}$ (together with the complete σ-algebra, $2^{\mathcal{T}}$). We also require a fixed collection $\{w_\tau : \tau \in \mathcal{T}\}$ of (essentially arbitrary) weights (often termed the "pseudo-prior": Geyer 1990, Geyer and Thompson 1995) for the temperature τ where $w_\tau \geq 0 \ \forall \tau \in \mathcal{T}$ and $\sum_{\tau \in \mathcal{T}} w_\tau = 1$. For each $\tau \in \mathcal{T}$, we have a probability distribution $\Pi_\tau(\cdot)$ on $(\mathcal{X}, \mathcal{F})$, with corresponding density $\pi_\tau(\boldsymbol{x})$ with respect to Lebesgue measure. (Of course, more general reference measures are possible, but for clarity we restrict to Lebesgue.) Often the densities $\pi_\tau(\boldsymbol{x})$ are known only in unnormalised form $\tilde{\pi}_\tau(\boldsymbol{x})$.

We then define a Markov chain upon the augmented state space $(\mathcal{X}, \mathcal{F}) \times (\mathcal{T}, 2^{\mathcal{T}})$. The stationary distribution is given by $\Pi(A \times \{\tau\}) = w_\tau \Pi_\tau(A)$ for $A \in \mathcal{F}$ and $\tau \in \mathcal{T}$. The distribution Π thus has density (with respect to Lebesgue measure on $\mathcal{X}$, crossed with counting measure on $\mathcal{T}$) given by $\pi(\boldsymbol{x}, \tau) = c w_\tau \tilde{\pi}_\tau(\boldsymbol{x})$, where

$$c = \left(\sum_{\tau \in \mathcal{T}} \int w_\tau \tilde{\pi}_\tau(d\boldsymbol{x}) \right)^{-1}.$$

We shall assume throughout that we only have un-normalised densities, but note that if normalised densities are available and are used above, then $c = 1$.

Applications in Bayesian inference often use a specific method for "heating" the target distribution, known as "heating up", where we take $\tilde{\pi}_\tau(\boldsymbol{x}) = \pi(\boldsymbol{x})^{1/\tau}$ for $\tau \in \mathcal{T}$. Here π_∞ would correspond to a uniform distribution over the entire parameter space, within which it is easy to traverse between any modes which slow convergence in the colder distributions. Of course, many alternative choices of temperatures and stationary distributions are possible.

To run the simulated tempering algorithm, we require a Markov chain proposal distribution $Q_1(\tau, \cdot)$ on $\mathcal{T}$, with corresponding proposal probabilities $q_1(\tau, \tau')$. (Geyer and Thompson (1995) suggest that with $\mathcal{T} = \{1, ..., T\}$, we might set $q_1(\tau, \tau+1) = q_1(\tau, \tau-1) = \frac{1}{2}$, for $\tau = 2, ..., T-1$ and $q_1(1, 2) = q_1(T, T-1) = 1$, but more general schemes are also common.) We also require a Markov chain proposal distribution $Q_2(\boldsymbol{x}, \cdot)$ on $\mathcal{X}$, with corresponding proposal densities $q_2(\boldsymbol{x}, \boldsymbol{x}')$ with respect to Lebesgue measure.

In terms of the above ingredients, the simulated tempering scheme works as follows. At time t and in state $(\boldsymbol{x}_t, \tau_t)$, we first propose a new temperature $\tau' \sim Q_1(\tau_t, \cdot)$. We then compute

$$A_1(\tau, \tau'; \boldsymbol{x}) = \frac{\pi(\boldsymbol{x}, \tau') q_1(\tau', \tau)}{\pi(\boldsymbol{x}, \tau) q_1(\tau, \tau')} = \frac{\tilde{\pi}_{\tau'}(\boldsymbol{x}) w_{\tau'} q_1(\tau', \tau)}{\tilde{\pi}_\tau(\boldsymbol{x}) w_\tau q_1(\tau, \tau')} \tag{1}$$

and $\alpha_1(\tau, \tau'; \boldsymbol{x}_t) = \min[1, A_1(\tau, \tau'; \boldsymbol{x}_t)]$. Then, with probability $\alpha_1(\tau, \tau'; \boldsymbol{x}_t)$, the proposal is accepted, so that $\tau_{t+1} = \tau'$. Otherwise, with probability $1 - \alpha_1(\tau, \tau'; \boldsymbol{x}_t)$, the proposal is rejected, so that $\tau_{t+1} = \tau$.

Next, we update the parameters $\boldsymbol{x}_t$, conditioning upon this new temperature, using the Metropolis Hastings transition scheme (Chib and Greenberg 1995) to obtain $\boldsymbol{x}_{t+1}$ as follows. We begin by generating $\boldsymbol{x}' \sim Q_2(\boldsymbol{x}_t, \cdot)$. We then accept and set $\boldsymbol{x}_{t+1} = \boldsymbol{x}'$ with probability $\alpha_2(\boldsymbol{x}, \boldsymbol{x}'; \tau_{t+1}) = \min[1, A_2(\boldsymbol{x}, \boldsymbol{x}', \tau_{t+1})]$ where

$$A_2(\boldsymbol{x}, \boldsymbol{x}'; \tau) = \frac{\tilde{\pi}_\tau(\boldsymbol{x}') q_2(\boldsymbol{x}', \boldsymbol{x})}{\tilde{\pi}_\tau(\boldsymbol{x}) q_2(\boldsymbol{x}, \boldsymbol{x}')}. \tag{2}$$

Otherwise, we set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$.

We summarise the simulated tempering algorithm as follows.

Algorithm 2.1

Step 1. Begin at time $t \leftarrow 0$, in some initial state $\boldsymbol{x}_0$ and temperature τ_0.

Step 2. Generate $\tau' \sim Q_1(\tau_t, \cdot)$, and $U_{t+1} \sim Unif[0, 1]$.

Step 3. If $U_{t+1} \leq \alpha_1(\tau, \tau'; \boldsymbol{x}_t)$ let $\tau_{t+1} \leftarrow \tau'$, otherwise let $\tau_{t+1} \leftarrow \tau_t$.

Step 4. Generate $\boldsymbol{x}' \sim Q_2(\boldsymbol{x}_t, \cdot)$, and $V_{t+1} \sim Unif[0, 1]$.

Step 5. If $V_{t+1} \leq \alpha_2(\boldsymbol{x}, \boldsymbol{x}'; \tau_{t+1})$ let $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}'$, otherwise let $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$.

Geyer and Thompson (1995) (and more recently Møller and Nicholls 1999) show how the simulated tempering schemes described above provide a very simple mechanism for introducing regenerations within the chain. If we introduce some distribution $\pi_{\tau^*}(\cdot)$ from which independent sampling is possible, then whenever $\tau_t = \tau^*$, we can update $\boldsymbol{x}_t$ with an independent draw from π_{τ^*} so that the new value, $\boldsymbol{x}_{t+1}$, is independent of $\boldsymbol{x}_t$, and the future path of the chain is independent of the past. Times when $\tau_t = \tau^*$ are regeneration times, and the segments of the sample path between regeneration times (*tours*) are stochastically independent (Mykland *et al* 1995). We also note that the "practical regeneration" of Brockwell and Kadane (2002) can be viewed, in this context, as a tempering algorithm with only two temperatures.

More generally, the range of $\boldsymbol{x}$ may differ between temperatures, so that different temperatures correspond to different state spaces, of either identical or differing dimensions. The latter case corresponds to "trans-dimensional" (or "reversible jump") MCMC algorithms (Norman and Filinov 1969, Preston 1977, Green 1995) for constructing Markov chain transitions between states of differing dimensions. If the $\mathcal{X}_\tau$ differ, then in general we must update τ and $\boldsymbol{x}$ jointly. This is considered in detail in Section 5.

2.2. Coupling from the past

One of the major issues associated with the use of MCMC methods is determining the length of the burn-in (convergence) period. Many methods have been proposed to attempt to determine this, some based upon sample statistics from the output of the chain (see Raftery and Lewis 1992; Gelman and Rubin 1992; and Brooks and Roberts 1998) and others based upon analytic bounds on the convergence rate (see e.g. Meyn and Tweedie 1994; Rosenthal 1995a; Douc *et al* 2002; and Appendix A herein), but none are completely satisfactory.

Perfect (or exact) simulation was developed as a method for overcoming these problems. Intuitively, it uses the idea of starting the Markov chain infinitely far in the past ($t = -\infty$), so that by time $t = 0$, it will have already reached equilibrium. The problem, then, is how to generate chains starting infinitely far in the past or, more precisely, to determine where such chains *would* have ended up at time zero.

Propp and Wilson (1996) describe the *Coupling From The Past* (CFTP) algorithm for adapting MCMC simulation so that a draw from the target distribution can be guaranteed at time $t = 0$, and it is upon this that we shall base our algorithm. (Another perfect sampling method, Fill's algorithm, is not considered here; see Fill 1998, Fill *et al* 2000.)

The idea of CFTP is as follows. Suppose that we started a different copy of our Markov chain from every conceivable starting state, at some time $t = -M$ in the past. Suppose further that by time $t = 0$ all of the chains had coalesced, i.e. at time $t = 0$ each of these chains is in the same state, which we shall denote by $\boldsymbol{X}^*$. Clearly, if a chain which is coupled with these original chains were started at any time before $t = -M$, then it must pass through some state at time $t = -M$ and will therefore also be in state $\boldsymbol{X}^*$ at time $t = 0$. Therefore, intuitively, if a coupled chain were started infinitely far in the past it too would be in state $\boldsymbol{X}^*$ at time $t = 0$ and, since this chain will have reached its equilibrium distribution, $\boldsymbol{X}^*$ must be a draw from that distribution. See Propp and Wilson (1996) for further discussion and details.

In practice, most CFTP algorithms proceed by constructing a sequence of pseudo-random inputs $\boldsymbol{u}_t \sim Unif[0,1]$, and an *update function* $\phi : \mathcal{X} \times [0,1] \to \mathcal{X}$, such that $\mathbb{P}[\phi(x, \boldsymbol{u}_t) \in E] = \mathcal{P}(x, E)$. We can then use ϕ to update each copy of the chain, thus ensuring that chains which have become equal, remain equal ever after. More formally, the CFTP algorithm proceeds as follows (Propp and Wilson 1996).

Algorithm 2.2

Step 1. Begin at time $t = -1$.

Step 2. Generate $\boldsymbol{u}_t \sim Unif[0,1]$.

STEP 3. FOR EACH $\boldsymbol{x} \in \mathcal{X}$, LET $\boldsymbol{X}_t^{(\boldsymbol{x})} = \boldsymbol{x}$, AND ITERATIVELY LET $\boldsymbol{X}_{s+1}^{(\boldsymbol{x})} = \phi(\boldsymbol{X}_s^{(\boldsymbol{x})}, u_s)$ FOR $s = t, t+1, \ldots, -1$.

STEP 3. CALCULATE THE SET OF END-POINTS $\{\boldsymbol{X}_0^{(\boldsymbol{x})} : \boldsymbol{x} \in \mathcal{X}\}$ OF ALL OF THE CHAINS. IF THIS SET CONTAINS ONLY A SINGLE STATE, RETURN THIS STATE AND STOP. OTHERWISE, LET $t \leftarrow t-1$ AND RETURN TO STEP 2.

The state in which the algorithm stops will then be a draw from the common stationary distribution of the chains. Of course, many variations are available. In particular, in Step 3 above, using $t \leftarrow 2t$ (after generating $u_s \sim Unif[0,1]$ for $2t \leq s \leq t-1$) in place of $t \leftarrow t-1$ is actually more efficient in general (Propp and Wilson 1996).

One obvious problem with implementing the CFTP algorithm is the need to run (perhaps repeatedly) an infinite number of chains from every conceivable starting point. One way around this problem is to use the idea of monotonicity to limit the number of chains that need to be simulated. For example, Propp and Wilson (1996) construct lower and upper chains which start at the two extremes of a finite partially-ordered state space. The lower chain always lies no higher than any other chain; similarly, the upper chain always lies no lower than any other chain. Then if these two chains coalesce, then every other chain in between must also have coalesced. Such "monotone CFTP" algorithms have also been extended to continuous and unbounded state spaces, see e.g. Green and Murdoch (1998). However, a general scheme of generic applicability to a wide range of realistic problems typical of MCMC applications has yet to be developed.

2.3. Small sets, Coupling, and Regeneration

Here we discuss small sets for Markov chains. For further background, see for example Nummelin (1984) and Meyn and Tweedie (1993).

Let $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ be a time-homogeneous Markov chain on a state space $\mathcal{X}$, with σ-algebra $\mathcal{F}$. Let $\mathcal{P}(\boldsymbol{x}, E) = \mathbb{P}[\boldsymbol{X}_{t+1} \in E \,|\, \boldsymbol{X}_t = \boldsymbol{x}]$ be the transition probabilities, and $\mathcal{P}^k(\boldsymbol{x}, E) = \mathbb{P}[\boldsymbol{X}_{t+k} \in E \,|\, \boldsymbol{X}_t = \boldsymbol{x}]$ be the higher-order transition probabilities.

A subset $\mathcal{S} \subseteq \mathcal{X}$ is *small* (or, (k_0, ϵ, ν)-small) if there exists a probability measure $\nu(\cdot)$ on $(\mathcal{X}, \mathcal{F})$, a positive integer k_0, and $\epsilon > 0$ such that we have the *minorisation condition*

$$\mathcal{P}^{k_0}(\boldsymbol{x}, E) \geq \epsilon\nu(E), \qquad \forall\, \boldsymbol{x} \in \mathcal{S},\ E \in \mathcal{F} \tag{3}$$

i.e., if we always have

$$\mathbb{P}(\boldsymbol{X}_{t+k_0} \in E \,|\, \boldsymbol{X}_t = \boldsymbol{x}) \;\geq\; \epsilon\,\nu(E)$$

for all $\boldsymbol{x} \in \mathcal{S}$, all time indices t, and all measurable subsets E of the state space. (Of course, if we instead have $\mathcal{P}^{k_0}(\boldsymbol{x}, E) \geq \delta\mu(E)$ for some *unnormalised* measure μ, then (3) still holds with $\nu(E) = \mu(E) / \mu(\mathcal{X})$ and $\epsilon = \delta\mu(\mathcal{X})$.)

Small sets are useful in many ways. Roughly speaking, they say that once the chain is in the set $\mathcal{S}$, then with probability ϵ it will "forget" its current state and simply jump to the distribution $\nu(\cdot)$, ignoring its current state. Such a jump is referred to as a "regeneration" since the chain "begins again" from the distribution $\nu(\cdot)$.

Small sets are also useful for coupling. If we are running multiple copies of the chain, which all happen to be in the small set $\mathcal{S}$ at time t, then we can construct the copies jointly so that with probability ϵ, at time $t+k$ they are all in the same exact state (which is itself chosen randomly according to $\nu(\cdot)$). It is this property that will be useful for the perfect simulation and convergence rate estimation procedures of Section 4.2 and Appendix A.

More formally, for $\boldsymbol{x} \in \mathcal{S}$ where S satisfies (3), define the "residual kernel"

$$\mathcal{R}^{k_0}(\boldsymbol{x}, E) = (1-\epsilon)^{-1}[\mathcal{P}^{k_0}(\boldsymbol{x}, E) - \epsilon\nu(E)]\,, \qquad \forall\, E \in \mathcal{F}\,. \tag{4}$$

Then $\mathcal{R}^{k_0}(\boldsymbol{x}, \cdot)$ is a probability measure on $\mathcal{X}$. Furthermore,

$$\mathcal{P}^{k_0}(\boldsymbol{x}, E) = \epsilon\nu(E) + (1-\epsilon)\mathcal{R}^{k_0}(\boldsymbol{x}, E)\,, \qquad \forall\, E \in \mathcal{F}\,,\ x \in S\,. \tag{5}$$

Hence, given $\boldsymbol{X}_t = \boldsymbol{x} \in \mathcal{S}$, to determine the value of $\boldsymbol{X}_{t+k_0}$, we may proceed by flipping an ϵ-coin (i.e., a coin whose probability of heads equals ϵ), and then choosing $\boldsymbol{X}_{t+k_0} \sim \nu(\cdot)$ or $\boldsymbol{X}_{t+k_0} \sim \mathcal{R}^{k_0}(\boldsymbol{x}, \cdot)$ if the coin is heads or tails respectively. (This corresponds to the *splitting construction* of Nummelin 1984.) We may use this to keep track of "regeneration times" $T_1, T_2, \ldots$, as follows. For simplicity we take the case $S = \mathcal{X}$, so that the entire state space is small. This corresponds to uniform ergodicity, which does not always hold, however it does sometimes hold for realistic models (see e.g. Rosenthal 1993).

Algorithm 2.3

Step 0. Assume (3) holds with $S = \mathcal{X}$.

Step 1. Start with $t = 0$, and some (possibly random) initial value X_0.

Step 2. Given X_t, choose X_{t+k_0} as follows.

Step 2A. Flip an ϵ-coin.

Step 2B. If the coin is heads, choose $\boldsymbol{X}_{t+k_0} \sim \nu(\cdot)$, and declare $t + k_0$ to be the next regeneration time.

Step 2C. If the coin is tails, choose $\boldsymbol{X}_{t+k_0} \sim \mathcal{R}^{k_0}(\boldsymbol{x}^t, \cdot)$.

Step 3. If $k_0 > 1$, then fill in the missing values $X_{t+1}, \ldots, X_{t+k_0-1}$ according to the Markov transition probabilities $\mathcal{P}(\boldsymbol{x}, \cdot)$, conditional on the already-chosen values of X_t and X_{t+k_0}.

Step 4. Set $t = t + k_0$, and return to Step 2.

This algorithm thus simultaneously constructs both the Markov chain values $X_1, X_2, \ldots$, and the regeneration times $T_1, T_2, \ldots$ as specified in Step 2B. (By convention, we set $T_0 = 0$.) Because of (5), this algorithm ensures that

$$\mathbb{P}(\boldsymbol{X}_{t+k_0} \in E \,|\, \boldsymbol{X}_t = \boldsymbol{x}) = \mathcal{P}^{k_0}(\boldsymbol{x}, E)\,, \qquad \forall\, \boldsymbol{x} \in \mathcal{X},\ E \in \mathcal{F}\,,$$

as it must. Furthermore, by construction, we have $\mathbb{P}[\boldsymbol{X}_{T_i} \in E] = \nu(E)$ for each regeneration time T_i. Finally, we define the i^{th} *tour* of the chain to be the sequence of random values $(X_{T_i}, X_{T_i+1}, \ldots, X_{T_{i+1}-1})$. We note that by construction, these tours are independent and identically distributed for $i = 1, 2, 3, \ldots$, both in terms of the length of the tour and the distribution of the values themselves. (If we start with $X_0 \sim \nu(\cdot)$, then we can include the $i = 0$ tour in this as well.)

Algorithm 2.3 is also useful for *coupling*. Formally, chains $\{\boldsymbol{X}^{(i)}\}_{i \in I}$, indexed by an index set I, are *coupled* if the corresponding random variables $\{\boldsymbol{X}_k^{(i)}\}_{k \in \mathbf{N}, i \in I}$ are all defined on a common probability space; see Lindvall (1992) for further discussion. Now, suppose the chains all have different initial values $\boldsymbol{X}_0^{(i)}$, but are all constructed simultaneously using Algorithm 2.3. This ensures that $\mathbb{P}(\boldsymbol{X}_{t+k_0}^{(i)} \in E \,|\, \boldsymbol{X}_t^{(i)} = \boldsymbol{x}) = \mathcal{P}^{k_0}(\boldsymbol{x}, E)$ for each $i \in I$. Suppose we specify in addition that the same coin (from Step 2A above) is used for each chain, and if the coin is heads, then the same value of $\boldsymbol{X}_{t+k_0}$ chosen from $\nu(\cdot)$ (in Step 2B above) is used for each chain. (If the coin is tails, then the values chosen in Step 2C above may be selected from any joint law, typically conditionally independently.) This will ensure that when $\{t \geq T_1\}$, we have $\boldsymbol{X}_t^{(i)} = \boldsymbol{X}_t^{(j)}$ for all $i, j \in I$. That is, the chains have all *coalesced* by time T_1. A formal definition is as follows.

DEFINITION 2.1. *A collection $\{\boldsymbol{X}^{(i)}\}_{i \in I}$ of coupled chains have coalesced at time t if $\boldsymbol{X}_t^{(i)} = \boldsymbol{X}_t^{(j)}$ for all $i, j \in I$.*

One useful property of small sets is summarised in the following simple lemma (taken from Lemma 6 of Rosenthal 1995a).

LEMMA 2.1. *Suppose a Markov transition kernel $\mathcal{P}$ on a state space $\mathcal{X}$ satisfies*

$$\mathcal{P}^{k_1}(\boldsymbol{x}, \mathcal{S}_2) \geq \epsilon_1 \qquad \forall\, \boldsymbol{x} \in \mathcal{S}_1$$

and

$$\mathcal{P}^{k_2}(\boldsymbol{x}, \cdot) \geq \epsilon_2\, \nu(\cdot) \qquad \forall\, \boldsymbol{x} \in \mathcal{S}_2\,,$$

for some probability measure $\nu(\cdot)$ on $\mathcal{X}$. Then the subset $\mathcal{S}_1$ is small with parameters $k_0 = k_1 + k_2$, and $\epsilon = \epsilon_1 \epsilon_2$.

Remark 1. Obviously, if a subset is (k_0, ϵ, ν)-small, then it is also (k_0, ϵ', ν)-small for any $\epsilon' < \epsilon$. This means that if we have an *underestimate* of ϵ, then we can still apply small set ideas (though less efficiently). This provides a certain flexibility when designing algorithms which make use of small sets. We shall return to this point in Sections 3.3 and 6.

Remark 2. We note that for coupling just *two* copies of a chain, the weaker property of a *pseudo-small set* suffices (Roberts and Rosenthal 2002b). However, since we are interested in coupling more than two chains, we use the small set property herein.

2.4. Converting CFTP to a Forward-Time Algorithm

Consider any MCMC algorithm (not necessarily tempering), which is uniformly ergodic (i.e., for which the entire state space is small). For any such algorithm, it is possible to convert CFTP into a forward-time perfect simulation algorithm which involves running the algorithm for a geometrically-distributed number of iterations. This idea was first suggested by Murdoch and Green (1998) in the context of chains whose transition distributions are completely known and subsequently extended by Wilson (2000) and Breyer and Roberts (2000). Here, we present and prove the underlying theorem in its most general form so as to be applicable to any uniformly ergodic simulation scheme.

THEOREM 2.1. *Consider a Markov chain* $\{\boldsymbol{X}_n\}$ *on a state space* $(\mathcal{X}, \mathcal{F})$*, with stationary distribution* $\pi(\cdot)$*. Suppose the chain's transition probabilities* $\mathcal{P}(\boldsymbol{x}, \cdot)$ *satisfy the uniform minorisation condition* $\mathcal{P}^{k_0}(\boldsymbol{x}, \cdot) \geq \epsilon\nu(\cdot)$ *for all* $\boldsymbol{x} \in \mathcal{X}$*, some positive integer* k_0*, some* $\epsilon > 0$*, and some probability measure* $\nu(\cdot)$ *on* $(\mathcal{X}, \mathcal{F})$*. Let* $\mathcal{R}^{k_0}(\boldsymbol{x}, \cdot) = (1-\epsilon)^{-1}[\mathcal{P}^{k_0}(\boldsymbol{x}, \cdot) - \epsilon\nu(\cdot)]$ *be the residual kernel. Let* $\{\boldsymbol{Y}_n\}$ *be a Markov chain with initial distribution* ν *i.e.,* $\boldsymbol{Y}_0 \sim \nu(\cdot)$*, and transition probabilities* $\mathcal{R}^{k_0}(\boldsymbol{x}, \cdot)$*. If we generate* $T \sim$ *Geometric*(ϵ) *independently of the chain* $\{\boldsymbol{Y}_n\}$*, then*

$$\boldsymbol{Y}_T \sim \pi(\cdot)\,.$$

That is, the chain $\{\boldsymbol{Y}_n\}$*, when started in* $\nu(\cdot)$ *and run for a random number* T *of iterations, produces a draw exactly from the stationary distribution* $\pi(\cdot)$*.*

Proof. Consider running a CFTP algorithm using the chain $\{\boldsymbol{X}_n\}$, with coupling and regenerations defined as in Algorithm 2.3. Let

$$T \;=\; \min\{t \geq 0 : \; \{\boldsymbol{X}_n\} \text{ regenerates at time } -k_0 t\}\,.$$

Then by construction, $T \sim \textit{Geometric}(\epsilon)$, since regenerations occur when Step 2B is used and this occurs with probability ϵ independently at each iteration. Furthermore, T is conditionally independent of $\{\boldsymbol{X}_n\}_{n=-k_0T+1}^{0}$, conditional on $\boldsymbol{X}_{-k_0T}$ and upon there being no further regenerations from time $-k_0T+1$ to time 0. Hence the use of $\mathcal{R}^{k_0}(\boldsymbol{x}, \cdot)$ rather than $\mathcal{P}^{k_0}(\boldsymbol{x}, \cdot)$.

It follows that the distribution of the CFTP output is the same as that of Y_T above. But the CFTP output is known to be distributed as $\pi(\cdot)$, so the result follows. □

Theorem 2.1 thus says that, whenever we have a uniformly ergodic Markov chain, we can generate a perfect draw from stationarity simply by running the residual chain for a $Geometric(\epsilon)$ time, where ϵ is the uniform minorisation parameter. We emphasise that Theorem 2.1 applies to *any* uniformly ergodic chain, whether or not it is related to simulated tempering.

Recall now that $Geometric(\epsilon)$ has mean $1/\epsilon$. Hence, for this Theorem to be useful, we require certain conditions:

(1) The distribution $\nu(\cdot)$ is feasible to sample from;

(2) The residual kernel $\mathcal{R}(x,\cdot)$ is feasible to run; and

(3) The value of ϵ is non-negligible.

Condition (1) will often hold, since $\nu(\cdot)$ is often constructed explicitly to facilitate the verification of the minorisation condition. If $k_0 > 1$ then condition (2) will usually fail; however, if $k_0 = 1$, then condition (2) may well hold, and we shall see in Section 4.2 that it is straightforward in our context. Finally, condition (3) is related to the convergence time of the underlying Markov chain, and will often hold for sufficiently rapidly mixing chains.

We shall see that, for the simulated tempering chains that we shall consider, conditions (1), (2), and (3) will indeed hold. Hence, Theorem 2.1 will indeed be very useful in this case.

3. Small Sets for Simulated Tempering

Here we consider the availability of small sets and minorisation conditions for simulated tempering. (Here the state at time t is $(\boldsymbol{X}_t, \tau_t) \in \mathcal{X} \times \mathcal{T}$, rather than just $\boldsymbol{X}_t \in \mathcal{X}$.) We shall provide several different results, depending upon just what is known about the "hot" distribution.

In the context of simulated tempering, Lemma 2.1 may be reformulated as follows.

Proposition 3.1. *Suppose that our simulated tempering chain has one temperature τ^* such that for some $k_2 \in \mathbb{N}$, $\epsilon_2 > 0$, and probability measure $\nu(\cdot)$ on $\mathcal{T} \times \mathcal{X}$,*

$$\mathbb{P}((\tau_{k_2}, \boldsymbol{X}_{k_2}) \in E \,|\, \boldsymbol{X}_0 = \boldsymbol{x},\ \tau_0 = \tau,\ \tau_1 = \tau^*) \ \geq\ \epsilon_2\, \nu(E)\,, \qquad \forall\, \boldsymbol{x} \in \mathcal{X}\,,\ \tau \in \mathcal{T}\,,\ E \in \mathcal{F} \times 2^{\mathcal{T}}\,.$$

Let $\mathcal{S}^* = \mathcal{X} \times \{\tau^*\}$. Suppose further that

$$\mathbb{P}(\tau_{k_1} = \tau^* \,|\, \boldsymbol{X}_0 = \boldsymbol{x},\ \tau_0 = \tau) \ \geq\ \epsilon_1\,, \quad \forall\, \boldsymbol{x} \in \mathcal{X}\,,\ \tau \in \mathcal{T}\,.$$

Then the entire state space $\mathcal{X} \times \mathcal{T}$ is (k_0, ϵ, ν)-small, where $k_0 = k_1 + k_2$, and $\epsilon = \epsilon_1 \epsilon_2$. That is, $\mathcal{P}^{k_0}[(\boldsymbol{x}, \tau), E] \geq \epsilon\, \nu(E)$ for all $(\boldsymbol{x}, \tau) \in \mathcal{X} \times \mathcal{T}$ and $E \in \mathcal{F} \times 2^{\mathcal{T}}$.

The set S^* will generally correspond to the "hot" distribution, which mixes very rapidly. In particular, we consider two extreme cases of this, namely S^* being an *atom* (see Definition 3.1) or S^* being *i.i.d.-like* (see Definition 3.3). Note that, if S^* is a *singleton*, i.e. $|S^*| = 1$, then S^* is both an atom and i.i.d.-like.

3.1. Case I: S^* is an atom

DEFINITION 3.1. *The set $\mathcal{S}^* = \{(\boldsymbol{x}, \tau^*) : \boldsymbol{x} \in \mathcal{X}\}$ is an atom if $\mathcal{P}[(\boldsymbol{x}, \tau^*), E] = \mu(E)$ for all $\boldsymbol{x} \in \mathcal{X}$ and all $E \in \mathcal{F} \times 2^{\mathcal{T}}$, i.e. the transition probabilities from S^* do not depend upon $\boldsymbol{x}$.*

If S^* is an atom, then we may take $k_2 = 1$ and $\epsilon_2 = 1$ and $\nu = \mu$, to obtain the following result.

PROPOSITION 3.2. *Suppose our simulated tempering chain has one constituent temperature τ^* which is an atom, with $\mu(\cdot)$ as above, and that*

$$\mathbb{P}(\tau_{k_1} = \tau^* \,|\, \boldsymbol{X}_0 = \boldsymbol{x},\ \tau_0 = \tau) \ \geq\ \epsilon_1\,, \quad \forall\, \boldsymbol{x} \in \mathcal{X}\,,\ \tau \in \mathcal{T}\,.$$

Then $\mathcal{P}^{k_1+1}[(\boldsymbol{x}, \tau), E] \geq \epsilon_1\, \mu(E)$ for all $(\boldsymbol{x}, \tau) \in \mathcal{X} \times \mathcal{T}$ and all $E \in \mathcal{F} \times 2^{\mathcal{T}}$, i.e. $\mathcal{X} \times \mathcal{T}$ is (k_1+1, ϵ_1, μ)-small.

Møller and Nicholls (1999) essentially combine this result with Algorithm 2.3 to obtain a regenerative simulated tempering algorithm which they then use as a basis for their perfect simulation scheme. We shall extend their result to the case where S^* is i.i.d.-like, before demonstrating how easily such distributions occur and then constructing a forward-time perfect simulation scheme which removes the need to iteratively restart the algorithm further and further back in time.

3.2. Case II: S^* is δ-uniform or i.i.d.-like

DEFINITION 3.2. *The set $\mathcal{S}^* = \{(\boldsymbol{x}, \tau^*) : \boldsymbol{x} \in \mathcal{X}\}$ is δ-uniform (for $0 < \delta \leq 1$) if there is a probability distribution μ on $\mathcal{X}$ such that*

$$\mathbb{P}(\boldsymbol{X}_1 \in E \,|\, \boldsymbol{X}_0 = \boldsymbol{x},\ \tau_0 = \tau,\ \tau_1 = \tau^*) \ \geq\ \delta\, \mu(E)\,, \quad \forall\, E \in \mathcal{F}\,,\ \boldsymbol{x} \in \mathcal{X}\,,\ \tau \in \mathcal{T}\,.$$

Thus, S^* is δ-uniform if the entry distributions into S^* all have in common a component of (uniform) size δ. If S^* is δ-uniform, and if also

$$\mathbb{P}(\tau_{k_1} = \tau^* \,|\, \boldsymbol{X}_0 = \boldsymbol{x},\ \tau_0 = \tau) \ \geq\ \epsilon\,, \quad \forall\, \boldsymbol{x} \in \mathcal{X}\,,\ \tau \in \mathcal{T}\,, \tag{6}$$

then $\mathcal{P}^{k_1}[(\boldsymbol{x}, \tau), E] \geq \epsilon\, \delta\, \mu(E)$ whenever $E \in \mathcal{F} \times \{\tau^*\}$. We thus obtain the following result.

COROLLARY 3.1. *Suppose our simulated tempering chain has one constituent temperature τ^* so that S^* is δ-uniform, and that (6) holds. Then $\mathcal{P}^{k_1}[(\boldsymbol{x}, \tau), E] \geq \delta\, \epsilon\, \mu(E)$ for all $(\boldsymbol{x}, \tau) \in \mathcal{X} \times \mathcal{T}$ and $E \subseteq \mathcal{X} \times \{\tau^*\}$, and $\mathcal{X} \times \mathcal{T}$ is therefore $(k_1, \delta\epsilon, M)$-small, where $M(E \times \{\tau^*\}) = \mu(E)$, and $M(\mathcal{X} \times \{\tau\}) = 0$ for $\tau \neq \tau^*$.*

DEFINITION 3.3. *The set $\mathcal{S}^* = \{(\boldsymbol{x}, \tau^*) : \boldsymbol{x} \in \mathcal{X}\}$ is i.i.d.-like if it is 1-uniform i.e., if there is a probability distribution Π_{τ^*} on $\mathcal{X}$ such that*

$$\mathbb{P}(\boldsymbol{X}_1 \in E \,|\, \boldsymbol{X}_0 = \boldsymbol{x},\ \tau_0 = \tau,\ \tau_1 = \tau^*) \ =\ \Pi_{\tau^*}(E)\,, \quad \forall E \subseteq \mathcal{X}\,,\ \boldsymbol{x} \in \mathcal{X}\,,\ \tau \in \mathcal{T}\,,$$

independently of the value of $\boldsymbol{x}$ and τ.

Thus, if S^* is i.i.d.-like, then the value of $\boldsymbol{x}$ upon entering S^* is drawn independently of the previous value. We then have:

COROLLARY 3.2. *Suppose our simulated tempering chain has a constituent temperature τ^* such that S^* is i.i.d.-like, and that (6) holds. Then $\mathcal{P}^{k_1}[(\boldsymbol{x},\tau),E] \geq \epsilon\,\Pi_{\tau^*}(E)$ for all $(\boldsymbol{x},\tau) \in \mathcal{X}\times\mathcal{T}$ and $E \subseteq \mathcal{X}\times\{\tau^*\}$, and $\mathcal{X}\times\mathcal{T}$ is therefore (k_1,ϵ,M)-small, where $M(E\times\{\tau^*\}) = \Pi_{\tau^*}(E)$, and $M(\mathcal{X}\times\{\tau\}) = 0$ for $\tau \neq \tau^*$.*

Combining Corollary 3.2 (with $k_1 = 1$) with Algorithm 2.3, we see that the following simulated tempering algorithm simulates a Markov chain having stationary distribution $\Pi(\boldsymbol{x},\tau)$. (Here, $\mathcal{R}[(\boldsymbol{x},\tau),(E,\tau')]$ is a simple extension of the definition in (4), with $\nu(E,\tau') = \Pi_{\tau^*}(E)$ if $\tau' = \tau^*$ and zero otherwise. We shall explain how Step 2C may be performed in the next section.)

ALGORITHM 3.1

STEP 0. ASSUME (6) HOLDS FOR SOME $\epsilon > 0$, AND THAT S^* IS I.I.D.-LIKE AS IN DEFINITION 3.3.
STEP 1. BEGIN AT TIME $t \leftarrow 0$ IN SOME STATE $(\boldsymbol{x}_0,\tau_0) \in \mathcal{X}\times\mathcal{T}$.
STEP 2A. FLIP AN ϵ-COIN.
STEP 2B. IF THE COIN IS HEADS, THEN GENERATE $\boldsymbol{x} \sim \Pi_{\tau^*}(\cdot)$ AND LET $(\boldsymbol{x}_{t+1},\tau_{t+1}) \leftarrow (\boldsymbol{x},\tau^*)$.
STEP 2C. IF THE COIN IS TAILS, THEN GENERATE $(\boldsymbol{x}_{t+1},\tau_{t+1}) \sim \mathcal{R}[(\boldsymbol{x}_t,\tau_t),\cdot]$ USING APPROPRIATE SIMULATED TEMPERING AND METROPOLIS HASTINGS UPDATING SCHEMES.
STEP 3. LET $t \leftarrow t+1$, AND RETURN TO STEP 2A.

3.3. Calculating ϵ

If S^* is i.i.d.-like, then the value of ϵ in (6) is crucial. Taking $k_1 = 1$, we have

$$\begin{aligned}\mathcal{P}[(\boldsymbol{x},\tau),\mathcal{S}^*] &= q_1(\tau,\tau^*)\,\alpha_1(\tau,\tau^*;\boldsymbol{x})\\ &\geq \inf_{\boldsymbol{x}\in\mathcal{X},\,\tau\in\mathcal{T}} q_1(\tau,\tau^*)\,\alpha_1(\tau,\tau^*;\boldsymbol{x})\end{aligned}$$

with $\alpha_1(\tau,\tau^*;\boldsymbol{x}) = \min[1, A_1(\tau,\tau^*;\boldsymbol{x})]$ and A_1 as defined in (1). Hence, by Corollary 3.2, $\mathcal{X}\times\mathcal{T}$ is $(1,\epsilon,\pi_{\tau^*})$-small, where

$$\epsilon = \inf_{\boldsymbol{x}\in\mathcal{X},\,\tau\in\mathcal{T}} q_1(\tau,\tau^*)\,\alpha_1(\tau,\tau^*;\boldsymbol{x}). \tag{7}$$

This value of ϵ corresponds to the smallest probability of moving to a point in $\mathcal{S}^*$ from any point in $\mathcal{X}\times\mathcal{T}$ and can be used in Algorithm 3.1 to sample from $\pi(\boldsymbol{x},\tau)$.

Remark 3. As noted in Remark 1, it is acceptable (though suboptimal) to use any *underestimate* of ϵ and, as we shall see, for practical purposes the value of ϵ in (7) provides an effective upper bound on the values used in the MCMC algorithms described in the next section. This observation also allows for the possibility of using various numerical techniques to estimate ϵ, provided that one is "conservative" to ensure an underestimate. For more on this see e.g. Cowles and Rosenthal (1998). We note also that it is sometimes possible to compute ϵ analytically even for complicated Markov chains, see for example Rosenthal (1995a) and Rosenthal (1996); but this can be difficult in general.

Example

If we use proposal probabilities for τ which are independent of the current temperature, so that $q_1(\tau,\tau') = \overline{q}_1(\tau')$ for all $\tau \in \mathcal{T}$, we have from (7) that

$$\begin{aligned} \epsilon &= \overline{q}_1(\tau^*) \inf_{\boldsymbol{x}\in\mathcal{X},\ \tau\in\mathcal{T}} \alpha_1(\tau,\tau^*;\boldsymbol{x}) \\ &= \overline{q}_1(\tau^*)\,\alpha_1^*, \text{ say.} \end{aligned} \tag{8}$$

As we shall see in the next section, in this special case optimal performance (i.e., an algorithm corresponding exactly to this value of epsilon) can be achieved.

In the special case where we have only two temperatures i.e., just the hot and cold distributions, the following result may be useful when the target distribution π_0 is a Bayesian posterior distribution.

PROPOSITION 3.3. *If the cold distribution $\tilde{\pi}_0(\boldsymbol{x}) = L(data|\boldsymbol{x})p(\boldsymbol{x})$ is a Bayesian posterior distribution corresponding to a likelihood $L(data|\boldsymbol{x})$ and prior $p(\boldsymbol{x})$, and we take only one other distribution $\tilde{\pi}_1(\boldsymbol{x}) = p(\boldsymbol{x})$ (and $\tau^* = 1$), then*

$$\alpha_1^* = \min\left[1,\ \frac{w_1 q_1(1,0)}{w_0 q_1(0,1) L^*}\right],$$

where L^* denotes the value of the likelihood evaluated at the maximum likelihood estimate.

Proof. Using the definition of α_1^* above, and of α_1 from Equation (1), we have

$$\alpha_1^* = \inf_{\boldsymbol{x}\in\mathcal{X},\ \tau\in\mathcal{T}} \alpha_1(\tau,\tau^*;\boldsymbol{x}) = \inf_{\boldsymbol{x}\in\mathcal{X},\ \tau\in\mathcal{T}} \min\left[1,\ \frac{\tilde{\pi}_{\tau^*}(\boldsymbol{x}) w_{\tau^*} q_1(\tau^*,\tau)}{\tilde{\pi}_\tau(\boldsymbol{x}) w_\tau q_1(\tau,\tau^*)}\right]$$

$$= \min\left[1,\ \inf_{\boldsymbol{x}\in\mathcal{X}} \frac{\tilde{\pi}_1(\boldsymbol{x}) w_1 q_1(1,0)}{\tilde{\pi}_0(\boldsymbol{x}) w_0 q_1(0,1)}\right] = \min\left[1,\ \inf_{\boldsymbol{x}\in\mathcal{X}} \frac{p(\boldsymbol{x}) w_1 q_1(1,0)}{L(\text{data}|\boldsymbol{x}) p(\boldsymbol{x}) w_0 q_1(0,1)}\right]$$

$$= \min\left[1,\ \inf_{\boldsymbol{x}\in\mathcal{X}} \frac{w_1 q_1(1,0)}{L(\text{data}|\boldsymbol{x}) w_0 q_1(0,1)}\right] = \min\left[1,\ \frac{w_1 q_1(1,0)}{L^* w_0 q_1(0,1)}\right]$$

as claimed. □

This result ensures that whenever the MLE's are available analytically, then so is ϵ. This makes the implementation of the perfect simulation schemes, described later in the paper, particularly easy to implement so long as we can sample directly from the full joint prior $p(\boldsymbol{x})$. Since *a priori* independence is a common assumption and since standard distributions are commonly used as priors, this condition will usually be satisfied. Of course, this result can be generalised beyond the Bayesian context by considering any decomposition of the target distribution into two components one of which can be maximised analytically and the other sampled from directly.

The result may be further extended to the case where we require intermediate temperatures between the hot and cold distributions. For example, we might take $\tilde{\pi}_\tau = L(\text{data}|\boldsymbol{x})^{1/\tau}p(\boldsymbol{x})$ as discussed in Section 2.1, retaining $p(\boldsymbol{x})$ as the hot distribution (obtained in the limit as $\tau \to \infty$). Then

$$\alpha_1^* = \min_{\tau\in\mathcal{T}} \frac{w_{\tau^*}q_1(\tau^*,\tau)}{w_\tau q_1(\tau,\tau^*)(L^*)^{1/\tau}}.$$

These results can be used to simplify the Bayesian implementation when the MLE's are available analytically.

4. New MCMC Algorithms

In this section, we propose and describe new MCMC algorithms which use ideas from simulated tempering to achieve perfect simulation. We first explain how the results above can be used to couple simulated tempering chains, before introducing our perfect simulation schemes.

4.1. Coupling of Simulated Tempering Chains

In order to more easily couple multiple copies of Algorithm 3.1, we reformulate the temperature and state transitions in terms of random variables which are generated independently of the current state of the chain as follows.

We begin by constructing a distribution $\overline{Q}_1(\cdot)$ on a subset $\mathcal{V} \subseteq \mathbb{R}$. Here $\mathcal{V}$ may be either discrete (i.e. a finite or countable subset of $\mathbb{R}$), in which case we write $\overline{q}_1(v)$ for its probability function, or continuous (i.e. an open subset of $\mathbb{R}$), in which case we write $\overline{q}_1(v)$ for its density function with respect to Lebesgue measure. We then define a function $g_1 : \mathcal{T}\times\mathcal{V} \to \mathcal{T}$ which is invertible in the first argument, in the sense that for all $\tau \in \mathcal{T}$ and $v \in \mathcal{V}$, there exists $v' \in \mathcal{V}$ such that $g_1[g_1(\tau, v), v'] = \tau$. If $\mathcal{V}$ is continuous, then we also assume that g_1 is differentiable in the second argument.

A simple special case is obtained when we take $g_1(\tau, v) = v$, which clearly satisfies the above conditions and is often used in practice.

We then propose a new temperature given the current state $(\tau_t, \boldsymbol{x}_t)$ by first generating v_t from the distribution $\overline{Q}_1(\cdot)$ (thus, v_t is generated independently of the current state). and then setting $\tau' = g_1(\tau_t, v_t)$. The value τ' is then accepted (so $\tau_{t+1} = \tau'$) with probability

$$\alpha_1(\tau_t, v_t; \boldsymbol{x}_t) \;=\; \min[1, A_1(\tau_t, v_t; \boldsymbol{x}_t)]\,, \tag{9}$$

where

$$A_1(\tau_t, v_t; \boldsymbol{x}_t) = \frac{\tilde{\pi}(\boldsymbol{x}_t, \tau')\overline{q}_1(v')}{\tilde{\pi}(\boldsymbol{x}_t, \tau_t)\overline{q}_1(v_t)}\,|J_{g_1}(\tau_t, v_t)|\,,$$

v' satisfies $g_1[g_1(\tau_t, v_t), v'] = \tau_t$, and $J_{g_1}(\tau_t, v_t) = \frac{\partial g_1(\tau_t, v_t)}{\partial v_t}$ if $\mathcal{V}$ is continuous, while $J_{g_1}(\tau_t, v_t) = 1$ if $\mathcal{V}$ is discrete. This accept-reject step is performed by first generating $U_t^1 \sim U[0,1]$, and then if $U_t^1 \leq \alpha_1(\tau_t, v_t; \boldsymbol{x}_t)$ we accept the proposal (and set $\tau_{t+1} = \tau'$), otherwise we reject it (and set $\tau_{t+1} = \tau_t$).

Once we have updated the temperature, we next update the state vector $\boldsymbol{x}_t$. For this we require a distribution $\overline{Q}_2(\cdot)$ on a discrete or open subset $\mathcal{U} \subseteq \mathcal{X}$, having probability or density $\overline{q}_2(\boldsymbol{u})$, and a function $g_2 : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ which is invertible in the first argument, i.e. for all $\boldsymbol{x} \in \mathcal{X}$ and $u \in \mathcal{U}$, there exists $u' \in \mathcal{U}$ such that $g_2[g_2(\boldsymbol{x}, u), u'] = \boldsymbol{x}$. If $\mathcal{U}$ is continuous then we require g_2 to be differentiable in the second argument. Note that g_2 could depend upon the current temperature but, for convenience, we omit any notational dependence here.

We then update $\boldsymbol{x}_t$ by first generating $\boldsymbol{u}_t \sim \overline{Q}_2(\cdot)$, and setting $\boldsymbol{x}' = g_2(\boldsymbol{x}_t, \boldsymbol{u}_t)$. The proposed value $\boldsymbol{x}'$ is then accepted (so $\boldsymbol{x}_{t+1} = \boldsymbol{x}'$) with probability $\alpha_2(\boldsymbol{x}_t, \boldsymbol{u}_t; \tau_{t+1}) = \min[1, A_2(\boldsymbol{x}_t, \boldsymbol{u}_t; \tau_{t+1})]$, where

$$A_2(\boldsymbol{x}_t, \boldsymbol{u}_t; \tau_{t+1}) = \frac{\tilde{\pi}(\boldsymbol{x}', \tau_t)\overline{q}_2(\boldsymbol{u}')}{\tilde{\pi}(\boldsymbol{x}_t, \tau_{t+1})\overline{q}_2(\boldsymbol{u}_t)} \left| J_{g_2}(\boldsymbol{x}_t, \boldsymbol{u}_t) \right| ,$$

$\boldsymbol{u}'$ satisfies $g_2[g_2(\boldsymbol{x}_t, \boldsymbol{u}_t), \boldsymbol{u}'] = \boldsymbol{x}_t$, and $J_{g_2}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \frac{\partial g_2(\boldsymbol{x}_t, \boldsymbol{u}_t)}{\partial \boldsymbol{u}_t}$ if $\mathcal{U}$ is continuous, or $J_{g_2}(\boldsymbol{x}_t, \boldsymbol{u}_t) = 1$ if $\mathcal{U}$ is discrete. The accept-reject step is performed by generating $U_t^2 \sim U[0,1]$; then if $U_t^2 \leq \alpha_2(\boldsymbol{x}_t, \boldsymbol{u}_t)$ we accept the proposal and set $\boldsymbol{x}_{t+1} = \boldsymbol{x}'$, otherwise we reject the proposal and set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$.

The resulting algorithm is as follows.

ALGORITHM 4.1

STEP 0. LET $\mathcal{V}$, $\overline{Q}_1(\cdot)$, $\overline{Q}_2(\cdot)$, $g_1(\tau, v)$ AND $g_2(\boldsymbol{x}, \boldsymbol{u})$ BE AS ABOVE.

STEP 1. BEGIN AT TIME $t \leftarrow t_0$ IN SOME STATE $(\boldsymbol{x}_{t_0}, \tau_{t_0}) \in \mathcal{X} \times \mathcal{T}$.

STEP 2. GENERATE $U_t^1, U_t^2 \sim Unif[0,1]$, $v_t \sim \overline{Q}_1(\cdot)$, AND $\boldsymbol{u}_t \sim \overline{Q}_2(\cdot)$.

STEP 3. IF $U_t^1 \leq \alpha_1[\tau_t, g_1(\tau_t, v_t); \boldsymbol{x}_t]$, LET $\tau_{t+1} \leftarrow g_1(\tau_t, v_t)$, ELSE LET $\tau_{t+1} \leftarrow \tau_t$.

STEP 4. IF $U_t^2 \leq \alpha_2[\boldsymbol{x}_t, g_2(\boldsymbol{x}_t, \boldsymbol{u}_t); \tau_{t+1}]$, LET $\boldsymbol{x}_{t+1} \leftarrow g_2(\boldsymbol{x}_t, \boldsymbol{u}_t)$, ELSE LET $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$.

STEP 5. LET $t \leftarrow t + 1$, AND RETURN TO STEP 2.

Example

Suppose that we use a random walk metropolis update incrementing the current state $\boldsymbol{x} \in \mathbb{R}^n$ by generating $\boldsymbol{u} \in \mathbb{R}^n$ from some density $\overline{q}_2(\boldsymbol{u})$ and setting $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{u}$. Then $|J_{g_2}(\boldsymbol{x}, \boldsymbol{u})| = 1$, $\boldsymbol{u}' = -\boldsymbol{u}$ and we obtain the usual acceptance ratio given in (2). Similarly, if we take $v = 1$ with probability 0.5 and $v = -1$ otherwise and set $g_1(\tau, v) = \min[T, \max(1, \tau + v)]$, then we obtain the tempering update of Geyer and Thompson (1995) with corresponding acceptance ratio given in (1). (Of course, this example requires taking $k_1 > 1$ in (6) to be able to reach τ^* after k_1 steps.)

Example

An independence sampler-type proposal for the temperature update is obtained by taking $\mathcal{V} = \mathcal{T}$ so that $\mathcal{V}$ is discrete, and taking $\overline{Q}_1(\cdot)$ to be the distribution placing equal mass on all temperatures $\tau \in \mathcal{T}$. We then take $g_2(\tau, v_t) = v_t$, so that $J_{g_2}(\tau, v) \equiv 1$. Since τ^* can be reached directly from any state, this implementation allows us to apply (6) for any $k_1 \geq 1$.

To facilitate coupling, we make one further observation. If $g_1(\tau_t, v_t) = \tau^*$, then we are proposing a move to the "hot" temperature τ^*. This move will be accepted provided $U_t^1 \leq \alpha_1(\tau_t, \tau^*; \boldsymbol{x}_t)$. In that case, if the hot distribution $\Pi_{\tau^*}(\cdot)$ is available for direct sampling, then we may choose to update $\boldsymbol{x}_{t+1} \sim \Pi_{\tau^*}(\cdot)$, independently of $\boldsymbol{x}_t$, rather than proceeding with proposals from $\overline{Q}_2(\cdot)$ as usual. The advantage of doing this, is that it explicitly creates an "i.i.d.-like" event, in which $\boldsymbol{x}_{t+1}$ is updated independently of $\boldsymbol{x}_t$, thus allowing us to apply the results of the previous section.

To aid with coupling, we do indeed perform this direct sampling from $\Pi_{\tau^*}(\cdot)$, at least when $U_t^1 \leq \alpha_1^*$, where

$$\alpha_1^* \;=\; \inf_{\boldsymbol{x} \in \mathcal{X},\, v \in V_\tau^*,\, \tau \in \mathcal{T}} \alpha_1(\tau, v; \boldsymbol{x})\,, \tag{10}$$

and $V_\tau^* = \{v : g_1(\tau, v) = \tau^*\}$. Explicitly introducing this modification into Algorithm 4.1, we obtain the following algorithm.

Algorithm 4.2

Step 0. Let $\mathcal{V}$, $\overline{Q}_1(\cdot)$, $\overline{Q}_2(\cdot)$, $g_1(\tau, v)$, $g_2(\boldsymbol{x}, \boldsymbol{u})$ and α_1^* be as above. Assume $S^* = \mathcal{X} \times \{\tau^*\}$ is i.i.d.-like as in Definition 3.3.

Step 1. Begin at time $t \leftarrow t_0$ in some state $(\boldsymbol{x}_{t_0}, \tau_{t_0}) \in \mathcal{X} \times \mathcal{T}$.

Step 2. Generate $U_t^1, U_t^2 \sim Unif[0, 1]$, $v_t \sim \overline{Q}_1(\cdot)$, and $u_t \sim \overline{Q}_2(\cdot)$.

Step 3. If $U_t^1 \leq \alpha_1^*$ and $g_1(\tau_t, v_t) = \tau^*$, then generate $\boldsymbol{w}_t \sim \Pi_{\tau^*}(\cdot)$, and let $\tau_{t+1} \leftarrow \tau^*$ and $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{w}_t$, and skip to step 6. Otherwise proceed to step 4.

Step 4. If $U_t^1 \leq \alpha_1[\tau_t, g_1(\tau_t, v_t); \boldsymbol{x}_t]$, let $\tau_{t+1} \leftarrow g_1(\tau_t, v_t)$, else let $\tau_{t+1} \leftarrow \tau_t$.

Step 5. If $U_t^2 \leq \alpha_2[\boldsymbol{x}_t, g_2(\boldsymbol{x}_t, \boldsymbol{u}_t); \tau_{t+1}]$, let $\boldsymbol{x}_{t+1} \leftarrow g_2(\boldsymbol{x}_t, \boldsymbol{u}_t)$, else let $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$.

Step 6. Let $t \leftarrow t + 1$, and return to step 2.

In this algorithm, the event $\{g_1(\tau_t, v_t) = \tau^* \text{ and } U_t^1 \leq \alpha_1^*\}$ corresponds to the event that, regardless of the current state, the chain will jump to $(\boldsymbol{w}_t, \tau^*)$ at time $t+1$. Similarly, if $V^* = \cap_{\tau \in \mathcal{T}} V_\tau^* = \{v : g_1(\tau, v) = \tau^* \; \forall \tau \in \mathcal{T}\}$, then *all* chains will jump to $(\boldsymbol{w}_t, \tau^*)$ at time $t+1$ if $v_t \in V^*$ and $U_t^1 \leq \alpha_1^*$. This event occurs with probability

$$\epsilon' \;=\; \overline{Q}_1(V^*)\alpha_1^*. \tag{11}$$

If this event does not occur, then we first update the temperature using a standard tempering update based upon v_t and U_t^1, and then update the state using an MCMC update based upon $\boldsymbol{u}_t$ and U_t^2, just as in Algorithm 4.1. Note that ϵ' defined above is smaller than ϵ as defined in (7) and so the algorithm above is not an optimal one. Whilst it is always possible in theory to construct an optimal algorithm, the practical difficulties in implementing STEP 3 above for the optimal algorithm are often insurmountable. However, in the special case where $\mathcal{V} = \mathcal{T}$ and $g_1(\tau, v) = v$ for all $\tau, v \in \mathcal{T}$, $V^* = \{\tau^*\}$ and all chains simultaneously jump to $(\boldsymbol{w}_t, \tau^*)$ with the optimal probability given in (8) i.e., $\epsilon' = \overline{q}_1(\tau^*)\alpha_1^* = \epsilon$.

A natural consequence of Remark 1 is that Step 3 in the above algorithm can be replaced by Step 3' below and yet will continue to provide draws from the correct stationary distribution.

STEP 3'. IF $U_t^1 \leq \alpha_1^*$ AND $v_t \in V^*$, THEN GENERATE $\boldsymbol{w}_t \sim \Pi_{\tau^*}(\cdot)$, AND LET $\tau_{t+1} \leftarrow \tau^*$ AND $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{w}_t$, AND SKIP TO STEP 6. OTHERWISE PROCEED TO STEP 4.

The advantage of this alternative formulation for Step 3 in Algorithm 4.2, is that we can then use the event $(v_t, U_t^1) \in V^* \times [0, \alpha_1^*]$ to cause multiple chains to all coalesce at the same time. Indeed, we see the following.

PROPOSITION 4.1. *Suppose that we jointly define multiple chains $\{(\boldsymbol{X}^{(i)}, \tau^{(i)})\}_{i\in I}$, which each follow Algorithm 4.2, but with Step 3 replaced by Step 3' defined above. Suppose further that they are all defined jointly (i.e., coupled) by all using the same identical random variable sequence $\{v_t, U_t^1, \boldsymbol{u}_t, U_t^2, \boldsymbol{w}_t\}_{t\in\mathbb{N}}$. Then the chains will have coalesced at time t^* (i.e., $\boldsymbol{X}_{t^*}^{(i)} = \boldsymbol{X}_{t^*}^{(j)}$ for all $i, j \in I$), where $t^* = 1+\min\{t : \{v_t \in V^*$ and $U_t^1 \leq \alpha_1^*\}$. Furthermore, if the chains have coalesced at some time t, then they will also have coalesced at time $t + k$ for all $k \geq 0$.*

4.2. A Perfect Simulation Algorithm Using Simulated Tempering

For our new algorithm, we use CFTP to create a perfect sampling scheme. Furthermore, we use the forward-time modification from Theorem 2.1. That is, rather than sampling at times $t = -1, -2, ...$, we draw $T \sim \textit{Geometric}(\epsilon')$ (with ϵ' as defined in (11) above) and sample forward from time 0 to time T, being sure to begin in the regeneration distribution at time 0.

The one subtle point is how to generate from the residual kernel $\mathcal{R}(\cdot, \cdot)$. Fortunately, this turns out to be easy. All we must do is, if $v_t \in V^*$ then we rescale U_t^1 so that it takes values only in $[\alpha_1^*, 1]$ i.e., we sample $U_t^1 \sim U(\alpha_1^*, 1)$ rather than $U_t^1 \sim U(0, 1)$. This has the effect of conditioning on not having both $v_t = \tau^*$ and $U_t^1 \leq \alpha_1^*$, while leaving all the other relative probabilities unchanged and is most easily achieved by replacing U_t^1 by $\alpha_1^* + (1 - \alpha_1^*)U_t^1$ whenever $v_t \in V^*$. An alternative is to toss away those samples where $v_t \in V^*$ and $U_t^1 \leq \alpha_1^*$ and re-sample v_t and U_t^1. Though potentially wasteful in this context (since we essentially sample $U_t^1 \sim U(\alpha_1^*, 1)$ via rejection rather than directly via transformation) the latter option provides the only viable scheme for more complex algorithms, as we shall see in the next section.

Putting this all together, we therefore obtain the following perfect simulation scheme.

ALGORITHM 4.3

STEP 0. LET $\mathcal{V}$, $\overline{Q}_1(\cdot)$, $\overline{Q}_2(\cdot)$, $g_1(\tau, v)$, $g_2(\boldsymbol{x}, \boldsymbol{u})$, V^*, α_1^* AND ϵ' BE AS ABOVE.

STEP 1. DRAW A RANDOM VARIABLE $T \sim \textit{Geometric}(\epsilon')$, AND LET $t \leftarrow 0$.

STEP 2. DRAW $\boldsymbol{w}_0 \sim \Pi_\tau^*(\cdot)$, AND LET $\boldsymbol{x}_1 \leftarrow \boldsymbol{w}_t$ AND $\tau_{t+1} \leftarrow \tau^*$. THEN LET $t \leftarrow 1$.

STEP 3. IF $t = T$, STOP AND RETURN $(\boldsymbol{x}_T, \tau_T)$. OTHERWISE CONTINUE.

STEP 4. DRAW $U_t^1, U_t^2 \sim U(0,1)$, $v_t \sim \overline{Q}_1(\cdot)$, AND $\boldsymbol{u}_t \sim \overline{Q}_2(\cdot)$.

STEP 5. IF $v_t \in V^*$ LET $U_t^1 \leftarrow \alpha_1^* + (1-\alpha_1^*)U_t^1$ AND CONTINUE. OTHERWISE, RETURN TO STEP 4.

STEP 6. IF $U_t^1 \leq \alpha_1[\tau_t, g_1(\tau_t, v_t); \boldsymbol{x}_t]$, LET $\tau_{t+1} \leftarrow g_1(\tau_t, v_t)$, ELSE LET $\tau_{t+1} \leftarrow \tau_t$.

STEP 7. IF $U_t^2 \leq \alpha_2[\boldsymbol{x}_t, g_2(\boldsymbol{x}_t, \boldsymbol{u}_t); \tau_{t+1}]$, LET $\boldsymbol{x}_{t+1} \leftarrow g_2(\boldsymbol{x}_t, \boldsymbol{u}_t)$, ELSE LET $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$.

STEP 8. LET $t \leftarrow t+1$, AND RETURN TO STEP 3.

Note that the variables $w_1, ..., w_T$ do not need to be generated, since the chain does not return to S^* after leaving the initial state. Intuitively, Algorithm 4.3 is simply a forward-time version (as in Theorem 2.1) of CFTP with a simulated tempering chain. The final state $(\boldsymbol{x}_T, \tau_T)$ will be the desired draw from $\pi(\boldsymbol{x}, \tau)$. In the next section, we generalise this scheme to problems where the range of $\boldsymbol{x}$ differs between temperatures.

5. Differing Supports, and Trans-dimensional MCMC

Suppose that for each temperature $\tau \in \mathcal{T}$, there is a distribution $\Pi_\tau(\cdot)$ defined on some state space $\mathcal{X}_\tau$. We assume that $\mathcal{X}_\tau$ is an open subset of $\mathbb{R}^{n_\tau}$ for some $n_\tau \in \mathbb{N}$, with corresponding Borel σ-algebra $\mathcal{F}_\tau$, and corresponding density $\pi_\tau(\boldsymbol{x})$ with respect to Lebesgue measure on $\mathbb{R}^{n_\tau}$.

If the dimensions n_τ differ, this corresponds to "trans-dimensional" (or "reversible jump") MCMC algorithms (Norman and Filinov 1969, Preston 1977, Green 1995) for constructing Markov chain transitions between states of differing dimensions. Schemes of this sort are most widely applied in the context of Bayesian model determination (Richardson and Green (1997); Dellaportas and Forster 1999; Fan and Brooks 2000) for which the stationary distribution denotes a joint distribution over both models $\tau \in \mathcal{T}$ and their associated parameters ($\boldsymbol{x} \in \mathcal{X}_\tau$). If the n_τ are equal, this may correspond to a problem in which the supports under different temperatures do not completely intersect. In some cases this can be deliberately imposed so as to provide a more efficient algorithm. See Section 6.1, for example.

In this context, the appropriate state space is given by $\mathcal{X} = \bigcup_\tau (\mathcal{X}_\tau \times \{\tau\})$, with corresponding σ-algebra $\mathcal{F} = \bigcup_{\tau\in\mathcal{T}} \bigcup_{E\in\mathcal{F}_\tau} (E \times \{\tau\})$, and corresponding stationary distribution given by $\Pi(E\times\{\tau\}) = w_\tau\,\Pi_\tau(E)$ for $E \subseteq \mathcal{X}_\tau$.

Since the $\mathcal{X}_\tau$ differ, it is generally impractical to first update τ while $\boldsymbol{x}$ remains fixed since such moves might rarely be accepted. For example, if $n_\tau = n\ \forall\tau$ and $\cap_{\tau\in\mathcal{T}}\mathcal{X}_\tau = \emptyset$ then no temperature transitions could be accepted whilst $\boldsymbol{x}$ remained fixed. In practice, most tempering schemes would update τ and $\boldsymbol{x}$ together in this context and so we shall focus on the joint updating scheme here. However, we note that similar ideas would apply equally well to the schemes where τ and $\boldsymbol{x}$ were updated separately if that were appropriate.

5.1. The Tempering Scheme

First, we require a proposal distribution $\overline{Q}_1(\cdot)$ on $\mathcal{V} \subseteq \mathbb{R}$ with corresponding probability function (if $\mathcal{V}$ is discrete) or density function (with respect to Lebesgue measure if $\mathcal{V}$ is continuous) $\overline{q}_1(v)$ and a function $g_1 : \mathcal{T}\times\mathcal{V} \to \mathcal{T}$ which is invertible in the first argument as before. Given that we are currently in state $(\boldsymbol{x}_t, \tau_t)$ we then propose a new value for τ by generating v_t from the distribution $\overline{Q}_1(\cdot)$ and setting $\tau' = g_1(\tau_t, v_t)$. We also require a proposal distribution $\overline{Q}_2(\cdot;\tau')$ defined on $\mathcal{U}_{\tau'} \subseteq \mathcal{X}_{\tau'}$ with corresponding probability or density $\overline{q}_2(\boldsymbol{u};\tau')$ and a function $g_2(\cdot;\tau,\tau') : \mathcal{X}_\tau \times \mathcal{X}_{\tau'} \to \mathcal{X}_{\tau'}$ which is invertible in that for all $\boldsymbol{x} \in \mathcal{X}_\tau$, $\boldsymbol{u} \in \mathcal{U}_{\tau'}$ and $\tau,\ \tau' \in \mathcal{T}$, there exists $\boldsymbol{u}' \in \mathcal{X}_\tau$ such that $g_2(g_2(\boldsymbol{x},\boldsymbol{u};\tau',\tau),\boldsymbol{u}';\tau,\tau') = \boldsymbol{x}$. We also require that g_2 be invertible in the second argument as before. We then generate $\boldsymbol{u}_t \sim \overline{Q}_2(\cdot;\tau')$ and set $\boldsymbol{x}' = g_2(\boldsymbol{x}_t,\boldsymbol{u}_t;\tau,\tau')$. The proposed move to $(\boldsymbol{x}',\tau')$ is then accepted with probability $\alpha[\boldsymbol{u}_t, v_t; \boldsymbol{x}_t, \tau_t] = \min(1, A[\boldsymbol{u}_t, v_t; \boldsymbol{x}_t, \tau_t])$ where

$$A[\boldsymbol{u}_t, v_t; \boldsymbol{x}_t, \tau_t] = \frac{\tilde{\pi}_{\tau'}(\boldsymbol{x}_t) w_{\tau'} \overline{q}_1(v') \overline{q}_2(\boldsymbol{u}';\tau_t)}{\tilde{\pi}_{\tau_t}(\boldsymbol{x}) w_{\tau_t} \overline{q}_1(v_t) \overline{q}_2(\boldsymbol{u}_t;\tau')} |J_g[(\boldsymbol{x}_t,\tau_t),(\boldsymbol{u}_t,v_t)]| \tag{12}$$

where v' satisfies $g_1(g_1(\tau_t, v_t), v') = \tau_t$ and $\boldsymbol{u}'$ satisfies $g_2(g_2(\boldsymbol{x}_t,\boldsymbol{u}_t;\tau',\tau_t),\boldsymbol{u}';\tau_t,\tau') = \boldsymbol{x}_t$, and

$$J_g[(\boldsymbol{x}_t,\tau_t),(\boldsymbol{u}_t,v_t)] = \frac{\partial(g_1(\tau_t,v_t) g_2(\boldsymbol{x}_t,\boldsymbol{u}_t;\tau_t,g_1(\tau_t,v_t))}{\partial(\boldsymbol{u}_t,v_t)}.$$

Our initial algorithm, generalising Algorithm 4.1 is thus as follows.

ALGORITHM 5.1

STEP 0. LET $\mathcal{T}$, $\{\mathcal{X}_\tau\}$, $\overline{Q}_1(\cdot)$ AND $\overline{Q}_2(\cdot;\tau)$, BE AS ABOVE, WITH THE $\{\mathcal{X}_\tau\}$ ALL OF EQUAL DIMENSION.

STEP 1. BEGIN AT TIME $t \leftarrow t_0$ IN SOME STATE $(\boldsymbol{x}_{t_0}, \tau_{t_0}) \in \mathcal{X}\times\mathcal{T}$.

STEP 2. GENERATE $v_t \sim \overline{Q}_1(\cdot)$ AND SET $\tau' = g_1(\tau_t, v_t)$, $\boldsymbol{u}_t \sim \overline{Q}_2(\cdot;\tau')$ AND SET $\boldsymbol{x}' = g_2(\boldsymbol{x}_t,\boldsymbol{u}_t;\tau_t,\tau')$, AND $U_t \sim Unif[0,1]$.

STEP 3. IF $U_t \leq \alpha(\boldsymbol{u}_t, v_t; \boldsymbol{x}_t, \tau_t)$, LET $\tau_{t+1} \leftarrow \tau'$ AND $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}'$, ELSE LET $\tau_{t+1} \leftarrow \tau_t$ AND $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$.

STEP 4. LET $t \leftarrow t+1$, AND RETURN TO STEP 2.

5.2. *The Perfect Simulation Scheme*

All of the results presented in the previous sections extend without change to the trans-dimensional context. Even though now τ and $\boldsymbol{x}$ are being updated simultaneously (rather than sequentially), the definition of i.i.d.-like, and the minorisation values of Corollary 3.2, go through without change.

To convert Algorithm 5.1 into a perfect simulation algorithm, we again require a value of ϵ. We begin by defining V_τ^* and V^* as in the previous section and setting

$$\alpha^* = \inf_{\boldsymbol{x}\in\mathcal{X}_\tau,\ v\in V_\tau^*,\ \boldsymbol{y}\in\mathcal{X}_{\tau^*},\ \tau\in\mathcal{T}} \alpha(\boldsymbol{u}, v; \boldsymbol{x}, \tau). \tag{13}$$

Then we take

$$\epsilon = \overline{Q}_1(V^*)\,\alpha^*. \tag{14}$$

Note that, in (13), we take the infimum just over those $\boldsymbol{x}$ and $\boldsymbol{y}$ in the corresponding state spaces, thus potentially allowing for larger ϵ by defining the state spaces to avoid "bad" points for the case where $n_\tau = n\ \forall\tau$. We return to this point in Section 7.

Applying the forward-time modification to Algorithm 5.1, we obtain the following.

ALGORITHM 5.2

STEP 0. LET $\mathcal{T}$, $\{\mathcal{X}_\tau\}$, $\overline{Q}_1(\cdot)$, $\overline{Q}_2(\cdot;\tau)$, ϵ, V^* AND α^* BE AS ABOVE, WITH THE $\{\mathcal{X}_\tau\}$ ALL OF EQUAL DIMENSION.

STEP 1. DRAW A RANDOM VARIABLE $T \sim Geometric(\epsilon)$, AND LET $t \leftarrow 0$.

STEP 2. DRAW $\boldsymbol{w}_0 \sim \Pi_\tau^*(\cdot)$, AND LET $\boldsymbol{x}_1 \leftarrow \boldsymbol{w}_t$ AND $\tau_{t+1} \leftarrow \tau^*$. THEN LET $t \leftarrow 1$.

STEP 3. IF $t = T$, STOP AND RETURN $(\boldsymbol{x}_T, \tau_T)$. OTHERWISE CONTINUE.

STEP 4. DRAW $U_t \sim U(0,1)$, $v_t \sim \overline{Q}_1(\cdot)$ AND SET $\tau' = g_1(\tau_t, v_t)$, $\boldsymbol{u}_t' \sim \overline{Q}_2(\cdot;\tau')$ AND SET $\boldsymbol{x}' = g_2(\boldsymbol{x}_t, \boldsymbol{u}_t, \tau_t, \tau')$.

STEP 5. IF $\tau' \in V^*$ AND $U_t \leq \alpha^*$, THEN RETURN TO STEP 4 AND DRAW FRESH VALUES. OTHERWISE CONTINUE.

STEP 6. IF $U_t \leq \alpha[\boldsymbol{u}_t, v_t; \boldsymbol{x}_t, \tau_t)$, LET $\tau_{t+1} \leftarrow \tau'$ AND $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}'$, ELSE LET $\tau_{t+1} \leftarrow \tau_t$ AND $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$.

STEP 7. LET $t \leftarrow t+1$, AND RETURN TO STEP 3.

5.3. *Specialisation to Bayesian Models*

As before, the implementation of the trans-dimensional perfect simulation scheme for Bayesian models can be simplified via the following generalisation of Proposition 3.3.

PROPOSITION 5.1. *If the target distribution $\tilde{\pi}(\boldsymbol{x}, \tau) = w_\tau L_\tau(data|\boldsymbol{x}) p(x|\tau)$, where $L_\tau(data|\boldsymbol{x})$ denotes the likelihood under model τ associated with parameters $\boldsymbol{x}$, and $p(\boldsymbol{x}|\tau)$ denotes the prior for $\boldsymbol{x}$ under model τ, and we take $g_2(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{x}$, $\overline{q}_2(\boldsymbol{u}; \tau, \tau^*) = \pi_{\tau^*}(\boldsymbol{u})$, and $\overline{q}_2(\boldsymbol{u}; \tau) = p(\boldsymbol{u}|\tau)$ for $\tau \neq \tau^*$, then*

$$\epsilon = \min \left[1, \min_{v \in V_t^*} \min_{\tau \in \mathcal{T}} \frac{w_{\tau^*} \overline{q}_1(v)}{w_\tau L_\tau^*} \right], \tag{15}$$

where L_τ^* denotes the value of the likelihood associated with model $\tau \in \mathcal{T}$, evaluated at the corresponding maximum likelihood estimate.

This result is particularly easy to apply when we have a nested model structure, flat priors and an independence proposal for τ. In this case, if we let τ_s denotes the maximal (or saturated) model, with $\mathcal{X}_{\tau_s} = \cup_{\tau \in \mathcal{T}} \mathcal{X}_\tau$, then the minimum in (15) is achieved when $\tau = \tau_s$, since $L_{\tau_s}^* \geq L_\tau^* \ \forall \tau \in \mathcal{T}$, by definition of τ_s as the superset of all other models in $\mathcal{T}$. We illustrate the application of this trans-dimensional perfect simulation scheme in Section 6.3

6. Examples

In this section, we present several examples to which we apply our new algorithms. Example I is a "toy" example for illustrative purposes; Example II is taken from an applied statistics problem; and Example III illustrates the extension to the transdimensional context.

6.1. *Example I: Sampling from a Beta*(α, β)

Following Green and Murdoch (1998), suppose we wish to obtain a perfect sample from a beta(α, β) distribution, having density $x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha, \beta)$ for $x \in [0, 1]$, where $\alpha, \beta > 1$. We can do this via our simulated tempering algorithm by constructing a chain with two temperatures: $\tau = 0$ corresponding to the beta distribution, and $\tau = 1$ corresponding to a standard uniform random variable (from which we can sample directly).

Here $\mathcal{X}_0 = \mathcal{X}_1 = [0, 1]$, so from (7) we have

$$\epsilon = \inf_{x \in [0,1],\ \tau = 0,1} q(\tau, 1)\alpha_1(\tau, 1; \boldsymbol{x})$$

Clearly, this infimum is obtained when $\tau = 0$ and, if we take $\overline{q}_1(0) = 1/2$ (i.e., $q(1, 0) = q(0, 1) = 1/2$) then $\epsilon = \alpha_1^*/2$, where

$$\alpha_1^* = \inf_{x \in [0,1]} \min \left(1, \frac{w_1 B(\alpha, \beta)}{w_0 x^{\alpha-1}(1-x)^{\beta-1}} \right).$$

This is clearly minimised (for $x \in [0, 1]$) at $x = (\alpha - 1)/(\alpha + \beta - 2)$, the mode of the beta distribution, so that

$$\alpha_1^* = \min \left(1, \frac{c w_1}{1 - w_1} \right),$$

since $w_0 = 1 - w_1$ and where

$$c = \frac{B(\alpha, \beta)(\alpha + \beta - 2)^{\alpha+\beta-2}}{(\alpha - 1)^{\alpha-1}(\beta - 1)^{\beta-1}}.$$

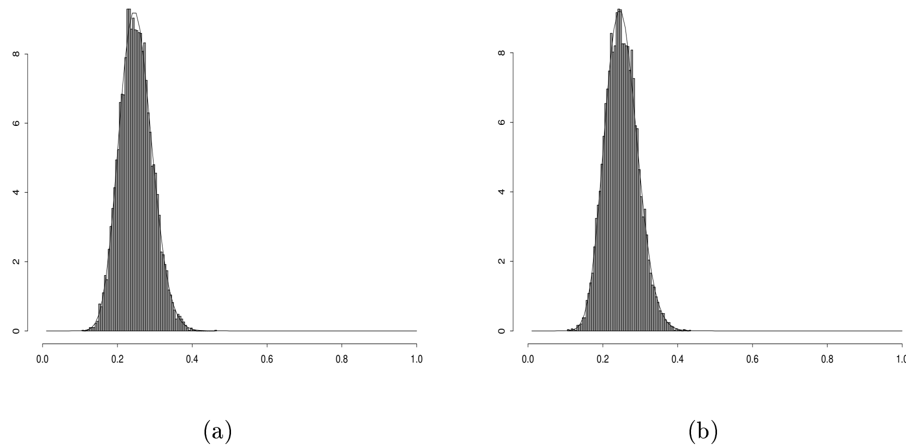


Fig. 1. True density function for a $Beta(25, 75)$ random variable imposed upon an empirical estimate (histogram) using (a) 10000 draws using our perfect simulation scheme, and (b) with 10000 draws from Splus' built-in beta generator, *rbeta*.

We might then set $w_1 = w_0 = 1/2$ for example and use Algorithm 4.3 to obtain independent draws from the Beta distribution of interest. Now, suppose that we run our perfect simulation to obtain a draw from the stationary distribution $\pi(\boldsymbol{x}, \tau)$. If $\tau_0 \neq 0$, we might run the perfect simulation algorithm again to obtain another draw from $\pi(\boldsymbol{x}, \tau)$. We can continue this until we obtain a draw for which $\tau_0 = 0$, then the corresponding value of x_0 will be a draw from the target Beta distribution. Now, since the marginal probability $\pi(\tau = 0) = 1 - w_1$ at stationarity, the number of simulations we would need to perform in order to obtain a draw from the cold (Beta) distribution follows a geometric distribution with parameter $1 - w_1$.

Thus, we would expect to require $1/(1 - w_1)$ simulations in order to obtain a draw from the Beta distribution, each of which would be of expected length $1/\epsilon$. Thus, if we wished to decide the value of w_1 which minimised the total expected number of iterations in order to obtain a draw from the Beta distribution, we need to minimise

$$\begin{aligned} N &= \frac{1}{1 - w_1} \frac{1}{\epsilon} \\ &= \frac{2}{(1 - w_1) \min(1, cw_1/(1 - w_1))} \\ &= \frac{2}{\min(1 - w_1, cw_1)} \end{aligned}$$

Clearly, the minimum of N is obtained when $\min(1 - w_1, cw_1)$ is maximised corresponding to the value $w_1 = 1/(c + 1)$. Therefore, the minimum expected number of iterations is $2(c + 1)/c$ with an expected number of simulations of $(c + 1)/c$, each of expected length 2 (since $\alpha^* = 1$ and therefore

$\epsilon = 1/2$). Now, since $\alpha^* = 1$, that means that we automatically accept any proposed jump to the hot distribution. Therefore, when we simulate forwards from $t = -T$ $v_t \neq 1$ $\forall t = -T, ..., 1$ (i.e., $v_t = 0$). This means that the only way that τ_0 could equal 1 would be if all proposals to move to 0 were rejected.

If, following Green and Murdoch (1998) we take $\alpha = 25$ and $\beta = 75$, then $c = 0.1082$, suggesting that we take $w_1 = 0.9024$. We would then expect to run 10.260 simulations, each of expected length 2 with a total number of expected iterations equal to 20.520, i.e. about 21 iterations. We used our perfect tempering scheme to generate from this beta distribution and these quantities were also verified empirically. The simulation output* is displayed in Figure 1, together with the corresponding output from a standard Beta random variable generator.

We compare the performance of our algorithm with the perfect simulation algorithm of Green and Murdoch (1998), where their distribution for T has a mean of around 52 iterations and, because they obtain this value by repeated simulation, rather than directly from the geometric as we do, they actually conduct an average of approximately 102 iterations. Of course, within each iteration we require two updates, so the figures of 21 and 102 aren't directly comparable. However, we can see that our simulation scheme seems to be somewhat quicker mainly due to the fact that we sample the starting point directly, rather than having to obtain it via repeated simulation. In more complex (and realistic) settings, this saving might be increasingly significant.

This simple illustrative example can be extended to consider the case where we have hot and cold distributions with differing state spaces. Suppose, for example, that our hot distribution is uniform on $[0, 1/2]$ rather than on $[0, 1]$. In this case $\mathcal{X}_1 = [0, 1/2]$, even though we still have $\mathcal{X}_0 = [0, 1]$. Suppose our implementation involves updating the temperature and state together as in (13), with proposals $q_2(x, x')$ for $x' \in [0, 1]$ and $q_2(x, x') = 2$ for $x' \in [0, 1/2]$. Then by (13),

$$\epsilon = \inf_{x \in [0,1/2],\ y \in [0,1]} q_1(0,1)\, \alpha_1[(x,0),\ (y,1)] = \frac{1}{2} \inf_{x \in [0,1/2],\ y \in [0,1]} \min\left(1,\ \frac{2\, w_1\, B(\alpha,\beta)}{w_0\, x^{\alpha-1}(1-x)^{\beta-1}\, 2}\right). \quad (16)$$

Assume for definiteness that $\alpha \leq \beta$. Then the infimum again occurs at $x = (\alpha - 1)/(\alpha + \beta - 2)$ and it is again optimal to take $w_1 = 1/(c + 1)$. Thus, with this alternative perfect simulation scheme based upon distinct state spaces, we still require about 21 iterations to obtain a draw from the Beta distribution. (Note that setting $\mathcal{X}_1 = [0, 1/2]$ was crucial, since for $x > 1/2$, the "2" in the numerator of (16) would be replaced by 0, thus leading to $\epsilon = 0$ if instead $\mathcal{X}_1 = [0, 1]$.) The main point here is that, even with an "incorrect" hot distribution like uniform on $[0, 1/2]$, with the "wrong" state space $\mathcal{X}_1 = [0, 1/2]$, the more general formulation corresponding to (13) still allows us to use our perfect simulation algorithm to efficiently sample from the (cold) distribution of interest. This point is discussed further in Section 7.

*The Splus code used is available at `http://probability.ca/jeff/research.html`.

6.2. Example II: Analysis of Band-return Data

Many wildlife studies involve the analysis of band-return or ring-recovery data. These involve marking individuals with tags or bands and then recording the time at which each band is returned upon the death of the corresponding individual. Here, we follow the Bayesian analysis outlined in Brooks *et al* (2002) for band-return data and examine the data of Brownie *et al* (1985) concerning a study of male mallards ringed as nestlings.

Here, we have data of the form $m_{ij}, i = 1, \ldots, I, j = 1, \ldots, J, J \geq I$, where m_{ij} denotes the number of animals released at the beginning of year i and whose tag was returned in the year up to the end of year j. We also have data R_i recording the number of animals marked and released at the beginning of year i. We then assume a model with the following parameters. Let λ denote the probability of a particular animal being recovered given that it has died. We also let ϕ denote the probability that an adult survives from one year to the next, but let ϕ_1 denote the corresponding survival rate for animals in their first year. This segregation of the population is common in such studies as it is commonly observed that the mortality rate of very young birds is much higher than that for adults.

Given data $\{R_i, m_{ij} : i = 1, \ldots, I, j = 1, \ldots, J\}$, we obtain the product-multinomial likelihood

$$L(\boldsymbol{R}, \boldsymbol{m}|\phi_1, \phi, \lambda) \;=\; c\,\Delta \prod_{i=1}^{I} \prod_{j=i}^{J} p_{ij}^{m_{ij}} \tag{17}$$

where

$$p_{ij} \;=\; \begin{cases} \lambda\tilde{\phi}_1 & j = i, \\ \lambda\phi_1\tilde{\phi}\phi^{j-i-1} & j = i+1, \ldots, J \end{cases},$$

$\tilde{\phi} = 1 - \phi$, $\tilde{\phi}_1 = 1 - \phi_1$, Δ denotes the likelihood term associated with individuals that are tagged but whose tags are not returned during the study and c denotes the product of multinomial normalisation constants. If we let $q_i = 1 - \sum_{j=i}^{J} p_{ij}$ be the probability of non-recovery of an animal released at the beginning of year i, either because it was still alive at the end of the experiment or because it died and was not found, and $u_i = R_i - \sum_{j=i}^{J} m_{ij}$ denote the number of animals released at the beginning of year i and never recovered, then $\Delta = \prod_{i=1}^{I} q_i^{u_i}$. Note that Δ is a function of all of the model parameters. Finally,

$$k = \prod_i \left[R_i!/(u_i! \prod_j m_{ij}!) \right].$$

Following Brooks *et al* (2002) we adopt standard uniform priors for all three parameters (since they are all probabilities). Thus our target ("cold") distribution is simply

$$\tilde{\pi}_0(\phi_1, \phi, \lambda|\boldsymbol{R}, \boldsymbol{m}) = \Delta \prod_{i=1}^{I} \prod_{j=i}^{J} p_{ij}^{m_{ij}}.$$

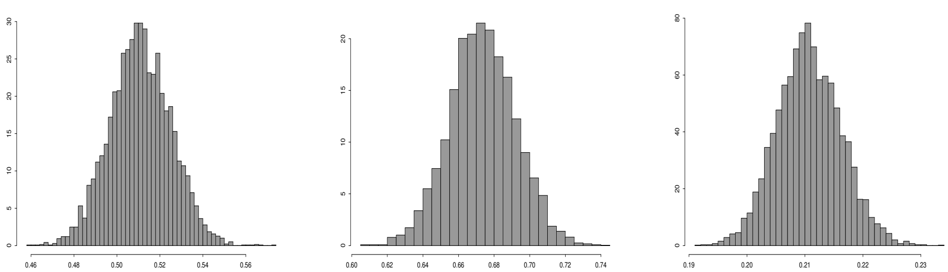
Then, following Proposition 3.3 we take our "hot" distribution for the model parameters to be the Uniform distribution over the unit cube. Though movement between these two temperatures may be made easier by the introduction of intermediate temperatures, we found that it was not necessary and so, in order to maximise the amount of time spent in the two key temperatures, we introduce no others and take $q_1(0,1) = q_1(1,0) = 1/2$.

Freeman and Morgan (1992) conduct a classical analysis of these data and show that $\log L^* = -157.17$. Thus, if we wish to minimise the the geometric waiting time of the perfect simulation algorithm, Proposition 3.3 suggests taking

$$\frac{w_1}{w_0} = \frac{L^* q_1(0,1)}{q_1(1,0)} = L^*$$

or setting $\log w_1 \approx -157.17$. In this case, $\alpha^* \approx 1$ and $\epsilon \approx 1/2$.

Running 10,000 replications of the perfect simulation scheme described in Algorithm 4.3 we find that with $w_1 = L^*$ all of the final states of the replications have $\tau = 1$. Thus, though this value minimises the length of each run, an enormous number of runs are required to gain even a small number of samples from the target distribution. By sequentially reducing the value of w_1, we observed that with $\log w_1 \approx -169$, the proportion of final states with $\tau = 1$ reduces to about 30%. However, the value of ϵ decreases to approximately 5×10^{-6}, corresponding to a geometric distribution with a mean of around 200,000 iterations.



(a) (b) (c)

Fig. 2. Histograms summarising the perfect simulation output for the Mallard example. Plots based upon 10,000 separate replications of the perfect simulation algorithm and correspond to (a) ϕ_1, (b) ϕ and (c) λ.

In this case, since w_1 is so small that $w_0 \approx 1$, multiplying w_1 by some small factor a has the effect of increasing the geometric waiting time by that factor, but decreasing the proportion of final states in the hot distribution by roughly the same amount. Thus, essentially any value of $\tilde{w}_1$ which enables the chains to jump between temperatures will lead to the same level of efficiency in terms of the number of expected iterations per sample from the target distribution. Using the value of $\tilde{w}_1$ above, we observed the simulation output summarised by the histograms in Figure 2. The corresponding posterior means (and standard deviations) for ϕ_1, ϕ and λ were 0.511 (0.014), 0.674 (0.019) and 0.211 (0.006) respectively and are comparable to previously published results for similar models, see

Brooks *et al* (2000).

6.3. Example III: Autoregressive Time Series

Suppose that we have a univariate time series $y_1, ..., y_t$ which we believe can be described by an autoregressive (AR) process of order τ, i.e.

$$y_t = \sum_{\ell=1}^{\tau} a_\ell y_{t-\ell} + z_t$$

where $z_t \sim N(0, \sigma^2)$. This can be rewritten in matrix-vector form as

$$\boldsymbol{z} = \boldsymbol{y}_1^k - \boldsymbol{Y}^k \boldsymbol{a}^k,$$

where $\boldsymbol{y}_0$ and $\boldsymbol{y}_1^k$ are formed by partitioning the data vector $\boldsymbol{y}$ into, respectively, the first k values and the remainder and $\boldsymbol{a}$ and $\boldsymbol{Y}^k$ take appropriate forms. Now suppose that the model order τ is unknown, then Ehlers and Brooks (2001) demonstrate how to construct a reversible jump MCMC scheme for exploring the corresponding posterior distribution $\pi(\boldsymbol{x}, \tau)$ where, under model $\tau = 1, ..., \tau_{\max}$, $\boldsymbol{x} \in \mathbb{R}^\tau$ (i.e., $\mathcal{X}_\tau = \mathbb{R}^\tau$, here). This scheme may be augmented by allowing $\tau = 0$ corresponding to a purely random process. This is clearly not of interest, but it is easy to show that with a conjugate prior, the posterior conditional distribution $\pi(\sigma^2|\tau = 0)$ is a member of the inverse gamma family and can therefore be sampled directly from its stationary distribution. Thus, $\tau = 0$ corresponds to our "hot" model τ^* and $\mathcal{S}^* = (\sigma^2, \tau^*)$ is i.i.d.-like.

We take vague independent $N(0, \sigma_a^2)$ priors for each of the autoregressive parameters under any model and propose updating the current model parameters using Gibbs steps in which we update the error variance from its standard conditional distribution and then the autoregressive parameters as a block update from their joint conditional distribution (see Ehlers and Brooks 2001). Similarly, we can update the model by randomly proposing to move to any model $\tau \in \{0, ..., \tau_{\max}\}$ with equal probability. If we take $\tau \neq 0$, then we propose new parameter values by simulating from the prior. However, if $\tau = 0$, then we draw a new value of σ^2 from the corresponding full conditional, $\pi_0(\sigma^2)$. Then, by Proposition 5.1 and the fact that all of the models $\tau \in \mathcal{T}$ are nested, we have that

$$\epsilon = \min\left(1, \frac{p(\tau^*)}{p(\tau_{\max}) L^*_{\tau_{\max}}}\right),$$

where $p(\tau)$ denotes the prior probability associated with model τ and $L_{\tau_{\max}}$ is the likelihood evaluated at the MLE:

$$\widehat{\sigma}^2 = \widehat{\boldsymbol{a}}^T \widehat{\boldsymbol{a}}; \quad \widehat{\boldsymbol{a}} = \boldsymbol{C}^T (\boldsymbol{Y}^{\tau_{\max}})^T \boldsymbol{y} / \widehat{\sigma}^2; \text{ and } \boldsymbol{C}^{-1} = (\boldsymbol{Y}^{\tau_{\max}})^T \boldsymbol{Y}^{\tau_{\max}} / \widehat{\sigma}^2.$$

7. Discussion

In this paper, we introduce a new algorithm (Algorithm 4.3) for producing perfect samples from a target distribution $\pi(\cdot)$. The algorithm is a version of CFTP using simulated tempering MCMC, but converted to a forward-time algorithm using Theorem 2.1. We extend these ideas to the trans-dimensional MCMC context, making the connection between the simulated tempering and reversible jump MCMC schemes (Algorithm 5.2), and relate our perfect simulation scheme to the theory of rigorous bounds on Markov chain convergence rates. We make several observations that simplify the implementation of the methods, including the connection to the maximum likelihood value, and the possibility of under-estimating ϵ. We illustrate our techniques on several different examples, including one involving a real data set. We conclude our paper by making some final observations relating to the efficient implementation of our methods and by comparing our proposed simulation schemes with the rejection sampling method.

The set of temperatures, combined with their associated weights, determine the probability that a single run of the perfect simulation scheme described in Algorithm 4.3 provides a realisation from the cold distribution. Obviously, we would like to make this probability as high as possible and this can be achieved by having as small a set of temperatures as possible. It is clear from equations (7) and (14) that when $k_1 = 1$, the value of epsilon is determined entirely by the transition from a single temperature τ to τ^*. Suppose that the minimising value in (14) is $(\boldsymbol{x}, \tau) = (\boldsymbol{x}_{opt}, \tau_{opt})$, so that $\epsilon = q_1(\tau_{opt}, \tau^*)\alpha_1(\tau_{opt}, \tau; \boldsymbol{x}_{opt})$. If $\tau_{opt} \neq \tau_1$ (i.e., the cold distribution), then ϵ can be increased by removing τ_{opt} from $\mathcal{T}$. Similarly, if $\tau_{opt} = \tau_1$, then the existence of the other temperatures within $\mathcal{T}$ has no affect upon the geometric distribution for T. Thus, these additional temperatures may also be removed from $\mathcal{T}$ since they necessarily decrease the probability that the algorithm returns a realisation from the cold distribution. Thus, whenever $k_1 = 1$, the most efficient forward simulation schemes comprises only two temperatures: the hot and the cold distribution.

For the $k_1 = 1$ case with only two temperatures as above, we can directly compare the efficiency of our scheme with that of the usual rejection sampling scheme using draws from the hot distribution to obtain realisations from the cold one. The general rejection sampling scheme proceeds as follows. Given a density f from which we wish to obtain samples, we can instead draw realisations $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ from some other density g and accept $\boldsymbol{x}_i$ as a realisation from f with probability $f(\boldsymbol{x}_i)/[Mg(\boldsymbol{x}_i)]$ where $M = \sup_{\boldsymbol{x}} f(\boldsymbol{x})/g(\boldsymbol{x})$. Here the supremum must be taken over the union of the supports of f and g. In the context of the beta example of Section 6.1 where the hot distribution is uniform on $[0, 1]$, the rejection sampling scheme using uniform draws to sample from the beta(a, b) density has an acceptance rate of c. Therefore, the expected number of draws to obtain a single realisation from f has a geometric distribution with mean $1/c$. This can be compared with expected number of iterations $N = 2(c+1)/c = 2+2/c$ using the perfect simulation scheme. Thus, the rejection sampling scheme is more efficient than the perfect simulation scheme in this case.

When we take temperatures with disjoint stationary distributions in the beta example the hot distribution can no longer be used as a rejection sampling density, since $M = \infty$. However, in this case the proposal density used to update x when moving from the cold to the hot distribution can be used instead. In this case the standard uniform would again be used as the rejection sampling density with the same efficiency as before.

Thus with $k_1 = 1$, the rejection sampling dominates the proposed perfect simulation scheme whether or not the supports of the hot and cold distributions differ. This is essentially because of the restriction that the chain must be able to move to the hot distribution within a single step and that whatever proposal distribution we use to make this transition can always be more efficiently used within a rejection sampling scheme. However, this need not be the case if we base our perfect simulation scheme on $k_1 > 1$ step transitions.

Returning to our beta example, suppose we take $\alpha = \beta = 1$ so that the cold distribution is simply the standard uniform, and take, as our hot distribution, the uniform on $[0, 1/2]$. Suppose also that for our between-temperature transitions we update the state variable x by sampling $u \sim U[-e/2, e/2]$ and setting $g_2(x, u; \tau, \tau') = x + u$. If $e < 1$, the chain cannot move directly to the hot distribution from every state and so $\epsilon = 0$ for the 1-step transition scheme. However, if we look at the corresponding 2-step scheme so that ϵ is the smallest probability of jumping from anywhere to the hot distribution in exactly two moves, then we find that $\epsilon = 1/4$. Thus, the 2-step scheme will work where the 1-step scheme does not. In addition, neither the hot distribution nor the between-temperature proposal distribution can be used to construct a rejection sampling scheme. This will always be the case when we have distinct supports and use a random-walk type proposal for the between-temperature transitions as may be comon in practice.

An additional advantage of taking $k_1 > 1$ is that in the calculation of ϵ we can allow the chain to make use of "intermediate" temperatures on the way from the hot distribution to the cold distribution, and a good choice of these intermediate temperatures (and corresponding distributions) could be extremely helpful. The addition of the extra steps through $k_1 > 1$ allows far greater flexibility in the perfect simulation scheme and rapidly increases the value of ϵ. On the other hand, the calculation of ϵ is often considerably harder to compute analytically (or even bound tightly) in this more general case, which may limit the perfect simulation algorithm's usefulness in practice until more efficient results can be developed in this area.

Acknowledgements. We thank the organiser Petros Dellaportas and all the participants in the TMR Workshop on MCMC Model Choice, in Spetses, Greece, in August 2001, for inspiration related to this paper. We are grateful also to Geoff Nicholls, Jesper Møller and Wilfrid Kendall all of whom were kind enough to discuss and comment on the basic ideas underpinning our work. The first author acknowledges the support of the UK Engineering and Physical Sciences Research Council under grant number AF/00537. The third author acknowledges support from NSERC of Canada.

References

Breyer, L. A. and G. O. Roberts (2000), Catalytic Perfect Simulation. Technical report, Lancaster University

Brockwell, A. and J. Kadane (2002), Practical Regeneration for MCMC Simulation. Technical report, Carnegie Mellon University

Brooks, S. P. (1998), Markov Chain Monte Carlo Method and its Application. *The Statistician* **47**, 69–100

Brooks, S. P., E. A. Catchpole, B. J. T. Morgan and S. C. Barry (2000), On the Bayesian Analysis of Ring-Recovery Data. *Biometrics* **56**, 951–956

Brooks, S. P., E. A. Catchpole, B. J. T. Morgan and M. Harris (2002), Bayesian Methods for Analysing Ringing Data. *Journal of Applied Statistics* **29**, 187–206

Brooks, S. P., P. Giudici and G. O. Roberts (2003), Efficient Construction of Reversible Jump Proposal Distributions (with discussion). *Journal of the Royal Statistical Society, Series B* **65**, 3–55

Brooks, S. P. and G. O. Roberts (1998), Diagnosing Convergence of Markov Chain Monte Carlo Algorithms. *Statistics and Computing* **8**, 319–335

Brownie, C., D. R. Anderson, K. P. Burnham and D. S. Robson (1985), *Statistical Inference from Band-Recovery Data – A Handbook*. United States Department of the Interior, Fish and Wildlife Service

Chib, S. and E. Greenberg (1995), Understanding the Metropolis-Hastings Algorithm. *The American Statistician* **49**, 327–335

Cowles, M. K. and J. S. Rosenthal (1998), A simulation approach to convergence rates for Markov chain Monte Carlo algorithms. *Statistics and Computing* **8**, 115–124

Dellaportas, P. and J. J. Forster (1999), Markov Chain Monte Carlo Model Determination for Hierarchical and Graphical Log-Linear Models. *Biometrika* **86**, 615–633

Doob, J. I. (1953), *Stochastic Processes*. Wiley: New York

Douc, R., E. Moulines and J. S. Rosenthal (2002), Quantitative Convergence Rates for Inhomogeneous Markov Chains. Technical report, University of Toronto

Ehlers, R. S. and S. P. Brooks (2001), Model Uncertainty in Integrated ARMA Processes. Technical report, University of Cambridge

Fan, Y. and S. P. Brooks (2000), Bayesian Modelling of Prehistoric Corbelled Domes. *Journal of the Royal Statistical Society, Series D* **49**, 339–354

Fill, J. A. (1998), An Interruptable Algorithm for Exact Sampling via Markov Chains. *Annals of Applied Probability* **8**, 131–162

Fill, J. A., M. Machida, D. J. Murdoch and J. S. Rosenthal (2000), Extension of Fill's perfect rejection sampling algorithm to general chains. *Random Structures and Algorithms* **17**, 290–316

Freeman, S. N. and B. J. T. Morgan (1992), A Modelling Strategy for Recovery Data from Birds Ringed as Nestlings. *Biometrics* **48**, 217–235

Gelman, A. and D. B. Rubin (1992), A Single Series from the Gibbs Sampler Provides a False Sense of Security. In J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (eds.), *Bayesian Statistics 4*, pp. 625–631, New York: Oxford University Press

Geyer, C. J. (1990), Reweighting Monte Carlo Mixtures. Technical report, University of Minnesota

Geyer, C. J. and E. A. Thompson (1995), Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. *Journal of the American Statistical Association* **90**, 909–920

Green, P. J. (1995), Reversible Jump MCMC Computation and Bayesian Model determination. *Biometrika* **82**, 711–732

Green, P. J. and D. Murdoch (1998), Exact Sampling for Bayesian Inference: Towards General Purpose Algorithms. In J. M. Bernardo, J. O. Berger, A. F. M. Smith and A. P. Dawid (eds.), *Bayesian Statistics 6*, Oxford University Press

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

Liu, J. S. and C. Sabatti (1999), Simulated Sintering: Markov chain Monte Carlo with Spaces of Varying Dimension. In J. M. Bernardo, A. F. M. Smith, A. P. Dawid and J. O. Berger (eds.), *Bayesian Statistics 6*, pp. 389–414, Oxford University Press

Marinari, E. and G. Parisi (1992), Simulated Tempering: A New Monte Carlo Scheme. *Europhysics letters* **19**, 451–458

Meyn, S. P. and R. L. Tweedie (1993), *Markov Chains and Stochastic Stability*. Springer-Verlag

Meyn, S. P. and R. L. Tweedie (1994), Computable Bounds for geometric Convergence Rates of Markov Chains. *Annals of Applied Probability* **4**, 981–1011

Møller, J. and G. K. Nicholls (1999), Perfect Simulation for Sample-Based Inference. Technical report, Aalborg University

Murdoch, D. J. and P. J. Green (1998), Exact Sampling from a Continuous State Space. *Scandinavian Journal of Statistics* **25**, 483–502

Mykland, P., L. Tierney and B. Yu (1995), Regeneration in Markov Chain Samplers. *Journal of the American Statistical Association* **90**, 233–241

Neal, R. (2000), Slice sampling. Technical report, University of Toronto

Neal, R. M. (1993), Probabilistic inference using Markov Chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, Technical Report No. CRG-TR-93-1

Norman, G. E. and V. S. Filinov (1969), Investigations of Phase Transitions by a Monte Carlo Method. *High Temperature* **7**, 216–222

Nummelin, E. (1984), *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge University Press

Preston, C. J. (1977), Spatial birth-death processes. *Bulletin of the International Statistical Institute* **46**, 371–391

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

Raftery, A. E. and S. M. Lewis (1992), How Many Iterations in the Gibbs Sampler? In J. M. Bernardo, A. F. M. Smith, A. P. Dawid and J. O. Berger (eds.), *Bayesian Statistics 4*, pp. 763–774, Oxford University Press

Richardson, S. and P. J. Green (1997), On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal of the Royal Statistical Society, Series B* **59**, 731–792

Roberts, G. O. and J. S. Rosenthal (1996), Quantitative Bounds for Convergence Rates of Continuous Time Markov Chains. *Electronic Journal or Probability* **1**, 1–21

Roberts, G. O. and J. S. Rosenthal (2002a), The Polar Slice Sampler. *Stochastic Models* **18**, 257–280

Roberts, G. O. and J. S. Rosenthal (2002b), Small and Pseudo-Small Sets for Markov Chains. *Stochastic Models* **17**, 121–145

Roberts, G. O. and R. L. Tweedie (1996), Exponential Convergence of Langevin Diffusions and their Discrete Approximations. *Bernoulli* **2**, 341–363

Rosenthal, J. S. (1993), Rates of Convergence for Data Augmentation on Finite Sample Spaces. *Annals of Applied Probability* **3**, 819–839

Rosenthal, J. S. (1995a), Minorization Conditions and Convergence Rates for Markov Chain Monte Carlo. *Journal of the American Statistical Association* **90**, 558–566

Rosenthal, J. S. (1995b), Rates of Convergence for Gibbs Sampling for Variance Component Models. *Annals of Statistics* **23**, 740–761

Rosenthal, J. S. (1996), Analysis of the Gibbs Sampler for a model related to James-Stein Estimators. *Statistics and Computing* **6**, 269–275

Wilson, D. B. (2000), How to couple from the past using a read-once source of randomness. *Random Structures and Algorithms* **16**, 85–113

A. Connection to Convergence Rate Estimation

Small sets can be used to provide *a priori* quantitative, theoretical bounds on the convergence time of Markov chains to stationarity, as we now discuss. This is relevant to our study in three ways.

Firstly, we see that the number of Markov chain iterations required by Algorithm 4.3 is distributed as $Geometric(\epsilon)$. We shall see below that, since the Markov chains used by Algorithm 4.3 satisfy a uniform minorisation condition with parameter ϵ, the distance to stationarity is also going down essentially like $Geometric(\epsilon)$. This implies that Algorithm 4.3 is "efficient", in the sense of requiring a number of Markov chain iterations of the same order as the number required for usual Markov chain convergence to stationarity.

Second, Algorithm 4.3 has in common with various convergence rate studies (e.g. Rosenthal 1995a, Rosenthal 1996, Cowles and Rosenthal 1998, Douc *et al* 2002) the need to compute a minorisation parameter ϵ. Various techniques (both analytic and numerical) employed to compute ϵ for convergence rate studies, can also be employed to implement Algorithm 4.3. The results below further indicate the connection between the two approaches.

Third, the convergence rate studies provide context for Algorithm 4.3. Indeed, a number of authors have derived such convergence rate bounds when the small set is just a subset of the state space (e.g. Meyn and Tweedie 1994; Rosenthal 1995a; Douc *et al* 2002). However, the use of simulated tempering allows for a minorisation condition over the *entire* state space, and a one-iteration minorisation (i.e. with $k_0 = 1$) at that. This is a disadvantage in that our ϵ must work from all states $\boldsymbol{x}$. However, it is a huge advantage in that the resulting algorithm, like the resulting convergence-rate results below, is much cleaner, not requiring e.g. a "drift condition" to force the chain to move back to the minorising subset. This provides further intuition about why simulated tempering is so helpful in Algorithm 4.3.

To begin our study of convergence rate estimation, we define the *total variation distance* between two probability measures $\mu(\cdot)$ and $\nu(\cdot)$ on a state space $\mathcal{X}$, with σ-algebra $\mathcal{F}$, by

$$\|\mu(\cdot) - \nu(\cdot)\|_{TV} = \sup_{A \in \mathcal{F}} |\mu(A) - \nu(A)| \,.$$

We shall be particularly interested in bounding the distance

$$\|\mathcal{P}^k[(\boldsymbol{x}, \tau), \cdot] - \pi(\cdot)\|_{TV} \,,$$

where $\mathcal{P}^k[(\boldsymbol{x}, \tau], \cdot)$ is the distribution of the chain after k steps when started at $(\boldsymbol{x}, \tau)$, and $\pi(\cdot)$ is some stationary distribution.

Convergence rate results are possible if our small set is just a subset of the state space, but they require the use of drift conditions and more complicated theorems (see e.g. Meyn and Tweedie 1994;

Rosenthal 1995a; Douc *et al* 2002). However, when the entire state space is small, as for our simulated tempering algorithms, then the situation is far simpler, and we have the following straightforward result. It goes back to Doob (1953), and follows easily from the coupling inequality (see e.g. Lindvall 1992) and the splitting construction referred to earlier; see e.g. Nummelin (1984), Meyn and Tweedie (1993) and Rosenthal (1995b).

PROPOSITION A.1. *Let $\mathcal{P}[(\boldsymbol{x},\tau),\cdot]$ be the transition probabilities for a time-homogeneous Markov chain on a state space $\mathcal{X}$, with stationary distribution $\pi(\cdot)$. Suppose that the entire state space $\mathcal{X}$ is (k_0,ϵ,ν)-small. Then for any starting point $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$, and any positive integer k, we have*

$$\|\mathcal{P}^k((\boldsymbol{x},\tau),\cdot)-\pi(\cdot)\|_{TV} \;\leq\; (1-\epsilon)^{\lfloor k/k_0\rfloor}$$

where $\lfloor r\rfloor$ denotes the greatest integer not exceeding r.

We now consider the question of convergence rates for simulated tempering. Combining Proposition A.1 with Proposition 3.1, we immediately obtain the following.

PROPOSITION A.2. *Suppose that our simulated tempering chain has one constituent temperature τ^* so that $\mathcal{S}^*=\{(\boldsymbol{x},\tau^*):\boldsymbol{x}\in\mathcal{X}\}$ is (k_2,ϵ_2,ν)-small. Suppose further that $\mathcal{P}^{k_1}[(\boldsymbol{x},\tau),\mathcal{S}^*]\geq\epsilon_1$ for all $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$. Let $\pi(\cdot)$ be a stationary distribution for the chain. Then for any starting point $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$, and any positive integer k, we have*

$$\|\mathcal{P}^k[(\boldsymbol{x},\tau),\cdot]-\pi(\cdot)\|_{TV} \;\leq\; (1-\epsilon_1\epsilon_2)^{\lfloor k/(k_1+k_2)\rfloor}$$

where $\lfloor r\rfloor$ is the greatest integer not exceeding r.

This proposition thus gives a quantitative upper-bound on the distance to stationarity of the chain after k iterations. As in Proposition 3.2, we have the following specialisation if S^* is an atom.

PROPOSITION A.3. *Suppose a simulated tempering chain has one constituent temperature τ^* so that S^* is an atom. Suppose further that $\mathcal{P}^{k_1}[(\boldsymbol{x},\tau),\mathcal{S}^*]\geq\epsilon_1$ for all $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$. Then for any starting point $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$, and any positive integer k, we have*

$$\|\mathcal{P}^k[(\boldsymbol{x},\tau),\cdot]-\pi(\cdot)\|_{TV} \;\leq\; (1-\epsilon_1)^{\lfloor k/(k_1+1)\rfloor}$$

Similarly, we have the following specialisation if S^* is δ-uniform.

PROPOSITION A.4. *Suppose a simulated tempering chain has one constituent temperature τ^* so that S^* is δ-uniform. Suppose further that $\mathcal{P}^{k_1}[(\boldsymbol{x},\tau),\mathcal{S}^*]\geq\epsilon_1$ for all $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$. Then for any starting point $(\boldsymbol{x},\tau)\in\mathcal{X}\times\mathcal{T}$, and any positive integer k, we have*

$$\|\mathcal{P}^k[(\boldsymbol{x},\tau),\cdot]-\pi(\cdot)\|_{TV} \;\leq\; (1-\delta\epsilon_1)^{\lfloor k/k_1\rfloor}$$

In particular, with $\delta = 1$, we have the following.

COROLLARY A.1. *Suppose a simulated tempering chain has one constituent temperature τ^* so that S^* is i.i.d.-like. Suppose further that $\mathcal{P}^{k_1}[(\boldsymbol{x}, \tau), \mathcal{S}^*] \geq \epsilon_1$ for all $(\boldsymbol{x}, \tau) \in \mathcal{X} \times \mathcal{T}$. Then for any starting point $(\boldsymbol{x}, \tau) \in \mathcal{X} \times \mathcal{T}$, and any positive integer k, we have*

$$\|\mathcal{P}^k((\boldsymbol{x}, \tau), \cdot) - \pi(\cdot)\|_{TV} \;\leq\; (1 - \epsilon_1)^{\lfloor k/k_1 \rfloor}$$

Corollary A.1 thus says essentially that the chain's convergence time, divided by k_1, is bounded above by a $Geometric(\epsilon_1)$ random variable. In particular, with $k_1 = 1$, they say that the chain has a convergence time which is essentially $Geometric(\epsilon_1)$.

We can easily apply Corollary A.1 to Algorithm 4.3. Indeed, in Algorithm 4.3, $\mathcal{X} \times \mathcal{T}$ is ϵ-small, and S^* is i.i.d.-like, so we obtain the following.

THEOREM A.1. *Consider the underlying simulated tempering Markov chain of Algorithm 4.3. Let $\epsilon = \overline{q}_1(\tau^*)\, \alpha^*$ and $\alpha^* = \inf_{\boldsymbol{x}, \in \mathcal{X},\, \tau \in \mathcal{T}} \alpha_1(\tau, \tau^*; \boldsymbol{x})$. Then for any starting point $(\boldsymbol{x}, \tau) \in \mathcal{X} \times \mathcal{T}$, and any positive integer k, the total variation distance of the Markov chain to stationarity after k iterations, satisfies*

$$\|\mathcal{P}^k[(\boldsymbol{x}, \tau), \cdot] - \pi(\cdot)\|_{TV} \;\leq\; (1 - \epsilon)^k \,.$$

Theorem A.1 says that, if we consider the simulated tempering chain underlying Algorithm 4.3, then the convergence time of this chain is essentially $Geometric(\epsilon)$. This is to be compared with the implementation of Algorithm 4.3 itself, which is run for a total of $T \sim Geometric(\epsilon)$ iterations. That is, Algorithm 4.3 manages to obtain a perfect sample from $\pi(\cdot)$, in a time comparable to the relaxation time of the underlying Markov chain.

Remark 4. In designing the underlying simulated tempering chain, there may be some freedom to determine how much of the time the chain will spend in the various constituent temperatures. Now, we see from Propositions A.2 and A.3 that, the more likely the chain is to jump to $\mathcal{S}^*$, the larger we can make ϵ_1 and the smaller we can make k_1. Hence, our bounds on the convergence rate get better the more likely the chain is to jump to $\mathcal{S}^*$. On the other hand, the more likely the chain is to jump to $\mathcal{S}^*$, the less time the chain will spend in $\tau = \tau_1$ in stationarity. Hence, the less useful will be the samples obtained from the simulated tempering stationary distribution. There is thus a trade-off, when increasing the chance of jumping to $\mathcal{S}^*$, between the speed of convergence to stationarity (and the corresponding running time) of Algorithm 4.3, and the usefulness of the resulting samples from stationarity, which we explore further in the context of the simple example in Section 6.1.