Perfect Forward Simulation via Simulated Tempering

Stephen P. Brooks†
University of Cambridge, U.K.
Yanan Fan
University of Kent at Canterbury, U.K.
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. Our algorithm appears to be efficient and widely applicable.

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

1. Introduction

MCMC methods have enjoyed wide-ranging interest in an enormous variety of applications over the past few years (Brooks 1998). Various adaptations of the basic algorithms have also appeared, such as the Langevin algorithm (Roberts and Tweedie 1996; Roberts and Rosenthal 1996; Neal 1993), the slice sampler (Neal 2000; Roberts and Rosenthal 2002a); and, of course, reversible jump MCMC methods (Green 1995; Brooks et al 2001).

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. In an attempt to overcome problems associated with slow-mixing Markov chains, which become “stuck” in local modes, 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 is based upon the idea of using a series of transition kernels $K_{	au}$ for $\tau \in \tau$, with corresponding (unnormalised) stationary densities $\pi_{\tau}(x)$ each defined upon the same state space $X \subseteq \mathbb{R}^{n}$. In many cases, we take the set of so-called “temperatures” $\tau$ to be a finite set of integers, $\tau = \{1, \ldots, T\}$. We then define a Markov chain upon the augmented state space $X \times \tau$ with corresponding distribution

$$\pi(x, \tau) = w_{\tau} \pi_{\tau}(x), \quad x \in X, \ \tau \in \tau,$$

where $w_{\tau}$ denotes some (essentially arbitrary) weight (often termed the “pseudo-prior”; Geyer 1990, Geyer and Thompson 1995) for the temperature $\tau$. We typically take $\pi_{1}$ to be the distribution of principal interest (often referred to as the “cold” distribution) and then the other kernels are chosen so as to progress in steps between two extremes; the cold distribution $\pi_{1}$, and a “hot” distribution

†Address for correspondence: Steve Brooks, Statistical Laboratory, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WB, U.K. E-mail: steve@statslab.cam.ac.uk
\[ \pi_T(x) = \pi(x)^{1/r}, \quad \tau \in T. \]

Here \( \pi_{\infty} \) 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, alternative choices of temperatures and stationary distributions are possible and in many cases, natural choices may be apparent from the context of any particular application.

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 general 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 we discuss how separate tempering simulations can be coupled, through a particular reformulation of the simulated tempering algorithm. In Section 5 we present 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 greatly reduces the computational overheads associated with the CFTP scheme in that we no longer need to repeatedly simulate the Markov chain forwards from starting points taken increasingly further back in time.

In Section 6 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 improves computational efficiency.

In fact our geometric result holds beyond the tempering context and can be applied to any CFTP scheme. Essentially, we construct a dominating process, and then derive the distribution of the first hitting time (in the CFTP sense). This removes the need to actually simulate the dominating process and essentially reduces the CFTP algorithm to a simple forward simulation scheme whose convergence time is known a priori.

In Section 7 we discuss how these same ideas can also be used to provide theoretical bounds on the convergence rates of the simulated tempering scheme, which are closely related to the running time of our perfect simulation scheme. In Section 8 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 9, a trans-dimensional auto-regressive example in Section 8, and a real data analysis in Section 10. We close (Section 11) 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

Once an appropriate class of densities has been defined, the simulated tempering scheme works as follows. At time \( t \) and in state \((x_t, \tau_t)\), we first update the temperature by proposing a new
temperature \( \tau' \sim q(\tau_i, \tau') \). This proposal is then accepted (so that \( \tau_{t+1} = \tau' \)) with probability \( \alpha(\tau, \tau'; x_t) = \min[1, A(\tau, \tau'; x_t)] \) where

\[
A(\tau, \tau'; x_t) = \frac{\pi_{\tau'}(x_t)w_{\tau'}q(\tau', \tau)}{\pi_{\tau}(x_t)w_{\tau}q(\tau, \tau')} \tag{1}
\]

Geyer and Thompson (1995) suggest that with \( T = \{1, \ldots, T\} \), we might set \( q(\tau, \tau + 1) = q(\tau, \tau - 1) = \frac{1}{T} \), for \( \tau = 2, \ldots, T - 1 \) and \( q(1, 2) = q(T, T - 1) = 1 \), but more general schemes are also common.

Next, we update the parameters \( x_t \), conditioning upon this new temperature, using the Metropolis Hastings transition scheme (Chib and Greenberg 1995) to obtain \( x_{t+1} \) as follows. We begin by generating \( x' \sim q(x_t, x') \) and set \( x_{t+1} = x' \) with probability \( \alpha(x, x'; \tau_{t+1}) = \min[1, A(x, x', \tau_{t+1})] \) where

\[
A(x, x'; \tau_{t+1}) = \frac{\pi_{\tau'}(x')q(x'_t, x)}{\pi_{\tau}(x)q(x_t, x')} \tag{2}
\]

Otherwise, we set \( x_{t+1} = x_t \).

Of course, more general schemes are possible. Suppose that at temperature \( \tau \), \( \pi_{\tau}(x) \) is defined upon some space \( \mathcal{X}_\tau \subseteq \mathbb{R}^n \), so that the range of \( x \) may differ between temperatures. In that case, we may wish to alter the temperature update so that the parameter vector does not remain constant, but rather is updated together with the temperature. In this case, we might propose \( \tau' \sim q(\tau_i, \tau') \) and \( x' \sim q(x_i, x'; \tau') \) which would be accepted with probability \( \alpha(\tau, x), (x', \tau') = \min(1, A(\tau, x, (x', \tau'))) \) where

\[
A[(\tau, x), (x', \tau')] = \frac{\pi_{\tau'}(x_t)w_{\tau'}q(\tau', \tau)q(x'_t, x; \tau)}{\pi_{\tau}(x_t)w_{\tau}q(\tau, \tau')q(x_t, x'; \tau')} \tag{3}
\]

Alternatively if \( \mathcal{X}_\tau \subseteq \mathbb{R}^n \), i.e., at each temperature \( \tau \) the corresponding density \( \pi_{\tau}(x) \) may be defined upon a different dimensional space, then we require a reversible jump MCMC (Green 1995) update for this trans-dimensional move. We shall return to this idea in Section 8.

Geyer and Thompson (1995) (and more recently Moller and Nicholls 1999) also show how the simulated tempering schemes described above provide a very simple mechanism for introducing regenerations within the chain. If we introduce some distribution \( \pi_{\tau*}(\cdot) \) so that independent sampling is possible, then whenever \( \tau_i = \tau^* \), we can update \( x_t \) with an independent draw from \( \pi_{\tau^*} \) so that the new value, \( x_{t+1} \), is independent of \( x_t \), and the future path of the chain is independent of the past. Times when \( \tau_i = \tau^* \) 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.

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 period. Many methods have been proposed, 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 1995b; Douc et al 2002). Perfect (or exact) simulation was developed as a method for overcoming these problems by starting Markov chains infinitely far in the past so that at time \( t = 0 \), they will have already reached their equilibrium distribution. The problem, then, is how to generate chains starting infinitely far in the past or, more importantly, to determine where those chains would be at time zero.

Propp and Wilson (1996) and Fill (1998) describe two different coupling schemes for adapting MCMC simulation so that a draw from the target distribution can be guaranteed at time \( t = 0 \). Propp and Wilson’s Coupling From The Past (CFTP) algorithm is perhaps the easiest to implement, and it is upon this that we shall base our algorithm.

Essentially, the idea of CFTP is to conduct our Markov chain simulations from all possible starting points beginning at some point infinitely back in the past so that we know that by time \( t = 0 \) all of these chains will have reached their equilibrium distributions. Let us suppose that these chains are coupled
and that they are constructed in such a way that at any time there is a non-zero probability that two or more chains will coalesce. In fact this is a more stringent condition than is strictly necessary but the exposition is slightly simplified with this assumption which will be satisfied by our own construction.

Suppose that we started an infinite number of coupled chains with identical stationary distributions from every conceivable starting state within the state space at some time $t = -M$ in the past. Suppose further that by time $t = 0$ all of the chains had coalesced, that is that at some time $-M \leq t^* \leq 0$ the sample paths of the of the chains have merged so that they all share the same sample path from time $t^*$ onwards. Then at time $t = 0$ each of these chains will be in the same state, which we shall denote by $X^*$. 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 $X^*$ at time $t = 0$. Therefore if a coupled chain were started infinitely far in the past it too would be in state $X^*$ at time $t = 0$ and, since this chain will have reached its equilibrium distribution, $X^*$ 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 random seeds $w_t$ which are the basis for the Markov chain transitions at the corresponding times. This sequence is the same for all replicate chains. The CFTP algorithm then proceeds as follows.

**Algorithm 2.1**

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

**Step 2.** Generate $w_t$. Then, starting from every possible state in $\mathcal{X} \times \mathcal{T}$ at time $t$ simulate each chain forward until time $t = 0$ using these new random seeds and all those from previously generated times $t + 1, \ldots, 0$.

**Step 3.** Calculate the set of end-points for all of the chains. If this set contains only a single state, stop. Otherwise return to Step 2. with $t = t - 1$.

The state in which the algorithm stops will be a draw from the common stationary distribution of the chains. Of course, alternative implementations are possible many of which provide more efficient sampling schemes. For example Green and Murdoch (1998) suggest incrementing time by setting $t = 2t$ in Step 3 above. This reduces the number of forward simulations required at the expense of potentially having to go further back in time than is strictly necessary.

One obvious implementational problem with the CFTP algorithm described in the introduction 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 dominance to limit the number of chains that need to be simulated. For example, Propp and Wilson (1996) use the idea of stochastic monotonicity to construct lower and upper chains which start at the two extremes of a finite unidimensional state space. The lower chain dominates the rest in that it always lies no higher than any other chain. Similarly, the upper chain dominates the rest in that it always lies no lower than any other chain. Clearly, by monitoring just these two chains we know that if these two coalesce then every other chain in between must also have coalesced. This is known as dominated CFTP. Both the CFTP and dominated CFTP algorithms have also been extended to continuous and unbounded state spaces, see (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 $X_0, X_1, X_2, \ldots$ be a time-homogeneous Markov chain on a state space $\mathcal{X}$. Let $P(x, A) = P[X_{t+1} \in A \mid X_t = x]$ be the transition probabilities, and $P^k(x, A) = P[X_{t+k} \in A \mid X_t = x]$ be the higher-order transition probabilities.

A subset $S \subseteq \mathcal{X}$ is small (or, $(K_0, \epsilon, \nu)$-small) if there exists a probability measure $\nu(\cdot)$ on $\mathcal{X}$, a positive integer $K_0$, and $\epsilon > 0$ such that we have the minorisation condition

$$P^{K_0}(x, A) \geq \epsilon \nu(A), \quad x \in S, \quad A \subseteq \mathcal{X}$$  \hspace{1cm} (4)
Simulated Tempering

5

i.e., if we always have
\[ P(X_{t+k_0} \in A \mid X_t = x) \geq \epsilon \nu(A) \]
for all \( x \in \mathcal{S} \), all time indices \( t \), and all measurable subsets \( A \) of the state space. (Of course, if we instead have \( P_k(x, A) \geq \delta \mu(A) \) for some unnormalised measure \( \mu \), then (4) still holds with \( \nu(A) = \mu(A) / \mu(X) \) and \( \epsilon = \delta \mu(X) \).)

Small sets are useful in many ways. Roughly speaking, they say that once the chain is in the set \( \mathcal{S} \), then with probability \( \epsilon \) 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 \( \epsilon \), 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 Sections 6 and 7.

More formally, for \( x \in \mathcal{S} \) where \( \mathcal{S} \) satisfies (4), define the “residual kernel”
\[ R_k(x, A) = (1 - \epsilon)^{-1} [P_k(x, A) - \epsilon \nu(A)]. \]
Then \( R_k(x, \cdot) \) is a probability measure on \( \mathcal{X} \). Furthermore,
\[ P_k(x, A) = \epsilon \nu(A) + (1 - \epsilon) R_k(x, A), \quad A \subseteq \mathcal{X}, \quad x \in \mathcal{S}. \]

Hence, given \( X_t = x \in \mathcal{S} \), to determine the value of \( X_{t+k_0} \), we may proceed by flipping an \( \epsilon \)-coin (i.e., a coin whose probability of heads equals \( \epsilon \)), and then choosing \( X_{t+k_0} \sim \nu(\cdot) \) or \( X_{t+k_0} \sim R_k(x, \cdot) \) 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 \( \mathcal{S} = \mathcal{X} \) i.e., when the entire state space is small; this corresponds to uniform ergodicity.

**Algorithm 2.2**

**Step 0. Assume (4) holds with \( \mathcal{S} = \mathcal{X} \).**

**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 \( \epsilon \)-coin.**

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

**Step 2C. If the coin is tails, choose \( X_{t+k_0} \sim R_k(x, \cdot) \).**

**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 \( P(x, A) \), 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. Because of (6), this algorithm ensures that
\[ P(X_{t+k_0} \in A \mid X_t = x) = P^{k_0}(x, A), \]
as it must. Furthermore, by construction, we have \( P[X_{T_i} \in A] = \nu(A) \) 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 2.2 is also useful for coupling. Formally, chains \( \{X^{(i)}\}_{i \in I} \), indexed by an index set \( I \), are coupled if the corresponding random variables \( \{X^{(i)}_k\}_{k \in \mathbb{N}, i \in I} \) are all defined on a common probability space; see Lindvall (1992) for further discussion. Now, suppose the chains all have different
initial values $X^{(i)}_0$, but are all constructed simultaneously using Algorithm 2.2. This ensures that $P(X^{(i)}_t | X^{(i)}_0 = x) = \mathcal{P}^{k_0}(x, A)$ 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 $X^{(i)}_t$ 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 $\{T_i \leq t\}$, we have $X^{(i)}_t = X^{(j)}_t$ 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 $\{X^{(i)}\}_{i \in I}$ of coupled chains have coalesced at time $t$ if $X^{(i)}_t = X^{(j)}_t$ 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 1995b), which follows by inspection.

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

$$\mathcal{P}^{k_1}(x, S_1) \geq \epsilon_1 \quad \text{for all } x \in S_1$$

and

$$\mathcal{P}^{k_2}(x, \cdot) \geq \epsilon_2 \nu(\cdot) \quad \text{for all } x \in S_2,$$

for some probability measure $\nu(\cdot)$ on $X$. Then the subset $S_1$ is small with parameters $k_0 = k_1 + k_2$, and $\epsilon = \epsilon_1 \epsilon_2$.

Finally, we note that for coupling just two copies of a chain, the weaker notion of a *pseudo-small set* suffices (Roberts and Rosenthal 2002b). However, since we are interested in coupling more than two chains here, we shall use the full small set notion here.

**Remark.** Obviously, if a subset is $(k_0, \epsilon, \nu)$-small, then it is also $(k_0, \epsilon', \nu)$-small for any $\epsilon' < \epsilon$. This means that if we have an *underestimate* of $\epsilon$, 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 Section 6.

### 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 $(X_t, \tau_t) \in X \times \mathcal{T}$, rather than just $X_t \in X$.) 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 $\tau^*$ such that $S^* = \{(x, \tau^*) : x \in X\}$ is $(k_2, \epsilon_2, \nu)$-small. Suppose further that, from any state in any temperature, there is probability at least $\epsilon_1$ of reaching $S$ (i.e., jumping to $\tau = \tau^*$ after $k_1$ steps), so that $\mathcal{P}^{k_1}[(x, \tau), S^*] \geq \epsilon_1$ for all $(x, \tau) \in X \times \mathcal{T}$. Then the entire state space $X \times \mathcal{T}$ is $(k_0, \epsilon, \nu)$-small, where $k_0 = k_1 + k_2$, and $\epsilon = \epsilon_1 \epsilon_2$. That is, $\mathcal{P}^{k_0}[(x, \tau), A] \geq \epsilon \nu(A)$ for all $(x, \tau) \in X \times \mathcal{T}$.

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). Of course, if $S^*$ is a *singleton*, i.e. $|S^*| = 1$, then $S^*$ is both an atom and i.i.d.-like.
3.1. Case I: $S^\ast$ is an atom

**Definition 3.1.** The set $S^\ast = \{(x, \tau^\ast) : x \in X\}$ is an atom if $P[(x, \tau^\ast), A] = \mu(A)$ for all $x \in X$, i.e. the transition probabilities from $S^\ast$ do not depend upon $x$.

If $S^\ast$ is an atom, then we may take $k_1 = 1$ and $\epsilon_1 = 1$ and $\nu = \mu$, to obtain the following result.

**Proposition 3.2.** Suppose our simulated tempering chain has one constituent temperature $\tau^\ast$ which is an atom, with $\mu(\cdot)$ as above, and that $P^{k_1}[(x, \tau), S^\ast] \geq \epsilon_1$ for all $(x, \tau) \in X \times T$. Then $P^{k_1+1}[(x, \tau), A] \geq \epsilon_1 \mu(A)$ for all $(x, \tau) \in X \times T$ i.e., $X \times T$ is $(k_1 + 1, \epsilon_1, \mu)$-small.

Møller and Nicholls (1999) essentially combine this result with Algorithm 2.2 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 $S^\ast$ is i.i.d.-like, before demonstrating how easily such distributions occur and then constructing a more efficient forward-time perfect simulation scheme which removes the need to iteratively restart the algorithm further back in time.

3.2. Case II: $S^\ast$ is i.i.d.-like

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

$$P(X_{k+1} \in B | X_k = x, \tau_k = \tau, \tau_{k+1} = \tau^\ast) \geq \delta \mu(B), \quad B \subseteq X, \ x \in X, \ \tau \in T,$$

uniformly over choice of $x$ and $\tau$.

Thus, $S^\ast$ is $\delta$-uniform if the entry distributions into $S^\ast$ all have a component of size $\delta$ in common. If $S^\ast$ is $\delta$-uniform, and if also

$$P^{k_1}[(x, \tau), S^\ast] \geq \epsilon, \quad (x, \tau) \in X \times T,$$

then $P^{k_1}[(x, \tau), A] \geq \epsilon \delta \mu(A)$ whenever $A \subseteq X \times \{\tau^\ast\}$. We thus obtain the following result.

**Corollary 3.1.** Suppose our simulated tempering chain has one constituent temperature $\tau^\ast$ so that $S^\ast$ is $\delta$-uniform, and that (7) holds. Then $P^{k_1}[(x, \tau), A] \geq \epsilon \delta \mu(A)$ for all $(x, \tau) \in X \times T$ and $A \subseteq X \times \{\tau^\ast\}$, and $X \times T$ is therefore $(k_1, \delta \epsilon, M)$-small, where $M(S \times \{\tau^\ast\}) = \mu(S)$, and $M(X \times T) = 0$ for $\tau \neq \tau^\ast$.

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

$$P(X_{k+1} \in B | X_k = x, \tau_k = \tau, \tau_{k+1} = \tau^\ast) = \pi_{\tau^\ast}(B), \quad B \subseteq X, \ x \in X, \ \tau \in T,$$

independently of the value of $x$ and $\tau$.

Thus, if $S^\ast$ is i.i.d.-like, then the value of the chain upon entering $S^\ast$ is drawn independently of the previous value. This is possible under the Metropolis Hastings updating scheme whenever we sample a new state for the chain directly from $\pi_{\tau^\ast}(\cdot)$. We also have:

**Corollary 3.2.** Suppose our simulated tempering chain has one constituent temperature $\tau^\ast$ so that $S^\ast$ is i.i.d.-like, and that (7) holds. Then $P^{k_1}[(x, \tau), A] \geq \epsilon \pi_{\tau^\ast}(A)$ for all $(x, \tau) \in X \times T$ and $A \subseteq X \times \{\tau^\ast\}$, and $X \times T$ is therefore $(k_1, \epsilon, M)$-small, where $M(S \times \{\tau^\ast\}) = \pi_{\tau^\ast}(S)$, and $M(X \times T) = 0$ for $\tau \neq \tau^\ast$.

Combining Corollary 3.2 (with $k_1 = 1$) with Algorithm 2.2, we see that the following simulated tempering algorithm simulates a Markov chain having stationary distribution $\pi(x, \tau)$. [Here, $R[(x, \tau), A]$ is as defined in (5), with $\nu(A) = \pi_{\tau^\ast}(A)$; we shall explain how Step 2C may be performed in the next section.]
Algorithm 3.1

Step 0. Assume (7) holds for some \( \epsilon > 0 \), and that \( S^* \) is i.i.d.-like as in Definition 3.3.

Step 1. Begin at time \( t = 0 \) in some arbitrary state \((x_0, \tau_0) \in X \times T\).

Step 2A. Flip an \( \epsilon \)-coin. If the coin is heads, perform Step 2B, else perform Step 2C.

Step 2B. Generate \( x \sim \pi_\tau \cdot (x) \) and set \((x_{t+1}, \tau_{t+1}) = (x, \tau^*)\).

Step 2C. Generate \((x_{t+1}, \tau_{t+1})\) from \( \Pi([x, \tau], A) \) using appropriate simulated tempering and Metropolis-Hastings updating schemes.

Step 3. Set \( t = t + 1 \), and return to Step 2A.

### 3.3. Calculating \( \epsilon \)

The value of \( \epsilon \) obtained depends upon the implementation. Taking \( k_1 = 1 \), we have

\[
P[(x, \tau), S^*] = q(\tau, \tau^*) \alpha(\tau, \tau^*; x) \geq \inf_{x \in X, \tau \in T} q(\tau, \tau^*) \alpha(\tau, \tau^*; x)
\]

with \( \alpha(\tau, \tau^*; x) = \min[1, A(\tau, \tau^*; x)] \) and \( A \) as defined in (1). Hence, by Proposition 3.2, \( X \times T \) is \((1, \epsilon, \pi_\tau)-small\), where

\[
\epsilon = \inf_{x \in X, \tau \in T} q(\tau, \tau^*) \alpha(\tau, \tau^*; x). \tag{8}
\]

We can thus use this value of \( \epsilon \) and Algorithm 3.1 to sample from \( \pi(x, \tau) \). Note that \( \epsilon \) corresponds to the smallest probability of moving to a point in \( S^* \).

More generally, if the state spaces \( X_\tau \) are distinct for each temperature \( \tau \), then we may wish to update \( x_t \) during the tempering transition, in which case we take

\[
\epsilon = \inf_{x_0 \in X_{\tau_0}, y \in X_{\star \tau}, \tau \in T} q(\tau, \tau^*) \alpha([x, \tau], (y, \tau^*)) \tag{9}
\]

with \( \alpha([x, \tau], (y, \tau^*)) = \min(1, A([x, \tau], (y, \tau^*))) \) and \( A \) as defined in (3).

Finally, suppose that we take a proposal distribution for \( \tau \) which is independent of the current temperature, so that \( \tau' \sim q(\tau') \). Then from (8), we have that

\[
\epsilon = q(\tau^*) \inf_{x \in X, \tau \in T} \alpha(\tau, \tau^*; x) = q(\tau^*) \alpha^*, \text{ say.} \tag{10}
\]

It is this last formulation that we generally shall use in the following sections, though the same ideas may be directly extended to the two more general cases above.

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 \( \pi_0 \) is a Bayesian posterior.

**Proposition 3.3.** If the cold distribution \( \pi_0(x) \propto L(\text{data}|x)p(x) \) is a Bayesian posterior corresponding to a likelihood \( L(\text{data}|x) \) and prior \( p(x) \), and we take only one other distribution \( \pi_1(x) \propto p(x) \), then

\[
\alpha^* = \frac{w_1 q(1, 0)}{w_0 q(0, 1) L^*},
\]

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

This result follows directly from the definitions of \( \alpha^* \) above and of \( \alpha \) in Equation (1).

This result ensures that whenever the MLE is available analytically, then so is \( \epsilon \) and 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 \( p(x) \). Since a priori independence
is a common assumption and since standard distributions are commonly used as priors, this condition will nearly always 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. In this case, we may take \( \pi_\tau \propto L(\text{data}|\mathbf{x})^{1/\tau} p(\mathbf{x}) \), retaining \( p(\mathbf{x}) \) as the hot distribution (obtained in the limit as \( \tau \to \infty \)). Then

\[
\alpha^* = \min_{\tau \in T} \frac{w_\tau \cdot q(\tau^*, \tau)}{w_\tau \cdot q(\tau, \tau^*) (L^*)^{1/\tau}}.
\]

These results can be used to simplify the Bayesian implementation when the MLE’s are available analytically and essentially means that using the perfect simulation methods above is equivalent (computationally) to the maximisation of the likelihood. Of course, the advantage of our methods is that we obtain draws from the full posterior distribution under any, essentially arbitrary, prior rather than simply the posterior mode under a flat prior corresponding to the MLE.

We now explain how the results above can be used to couple simulated tempering chains, before introducing our perfect simulation scheme.

4. Coupling of Simulated Tempering Chains

In the context of our simulated tempering algorithm, we can couple two or more chains by reformulating the simulated tempering transitions in Section 1 as follows. We first update the temperature \( \tau \), but now suppose that this is done by generating \( v_t \in \mathcal{V} \subseteq \mathbb{R} \) from some distribution \( q_1(v) \) i.e., \( v_t \) is generated independently of the current state. We then set set \( \tau' = g(\tau, v_t) \) for some function \( g: \mathcal{T} \times \mathcal{V} \to \mathcal{T} \) which is differentiable in the second argument and invertible in the first, in the sense that for all \( \tau \in \mathcal{T} \) and \( v \in \mathcal{V} \), there exists \( v' \in \mathcal{V} \) such that \( g(g(\tau, v), v') = \tau' \). For our purposes, we shall normally take \( q_1(v) \) to be a distribution placing equal mass on all temperatures \( \tau \in \mathcal{T} \) and then take \( g(\tau, v_t) = v_t \). This ensures that \( g(\tau, \tau^*) > 0 \forall \tau \in \mathcal{T} \).

Whatever the choice of \( q_1 \) and \( g \), the move is then accepted with probability

\[
\alpha_1(\tau_t, v_t; \mathbf{x}_{t+1}) = \min(1, A_1[\tau_t, v_t; \mathbf{x}_{t+1}]),
\]

where

\[
A_1(\tau_t, v; \mathbf{x}_{t+1}) = \frac{\pi(\mathbf{x}_{t+1}, \tau') q_1(v')}{\pi(\mathbf{x}_{t+1}, \tau) q_1(v)} |J_g(\tau, v)|,
\]

\( v' \) satisfies \( g(\tau_t, v_t), v' = \tau_t \), and

\[
J_g(\tau, v) = \frac{\partial g(\tau, v)}{\partial v}.
\]

This accept-reject step is performed by generating \( v_t' \sim U[0, 1] \) and if \( v_t' \leq \alpha_1(\tau_t, v_t; \mathbf{x}_{t+1}) \) we accept the transition, else it is rejected.

Once we have updated the temperature, we next update the state vector \( \mathbf{x}_t \). This is done by generating \( u_t \in \mathcal{U} \subseteq \mathbb{R}^n \) from some density \( q_2(u) \) and then setting \( \mathbf{x}' = f(\mathbf{x}_t, u) \) for some function \( f: \mathcal{X} \times \mathcal{U} \to \mathcal{X} \) which is invertible in the same sense as the function \( g \) above. The move from \( \mathbf{x}_t \) to \( \mathbf{x}' \) is then accepted with probability \( \alpha_2(\mathbf{x}_t, u_t; \tau_t) = \min(1, A_2[\mathbf{x}_t, u_t; \tau_t]) \), where

\[
A_2(\mathbf{x}, u; \tau_t) = \frac{\pi(\mathbf{x}', \tau_t) q_2(u')}{\pi(\mathbf{x}, \tau_t) q_2(u)} |J_f(\mathbf{x}_t, u)|
\]

and

\[
J_f(\mathbf{x}_t, u) = \frac{\partial f(\mathbf{x}_t, u)}{\partial u}.
\]

This accept-reject step is performed by generating \( u_t' \sim U[0, 1] \) and if \( u_t' \leq \alpha_2(\mathbf{x}_t, u_t) \) we accept the transition, else it is rejected.
Example

Suppose that we use a random walk Metropolis update incrementing the current state \( x \in \mathbb{R}^n \) by generating \( u \in \mathbb{R}^n \) from some density \( q(u) \) and setting \( x' = x + u \). Then \( |J_p(x, u)| = 1 \), \( u' = -u \) 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 \( g(\tau, v) = \min[T, \max(1, \tau + v)] \) then we obtain the tempering update of Geyer and Thompson (1995) with corresponding acceptance ratio given in (1). As explained above, we will take \( q_1(v) = 1 / |T| \) for \( v \in T \) and \( g(\tau, v) = v \). In this case, \( J_g(\tau, v) = 1 \).

As well as the tempering and Metropolis Hastings updates, we will need a “special” update to be used whenever we find ourselves in the “hot” distribution. For this transition we require only one random variable \( w_t \in \mathcal{X} \) which we sample directly from \( \pi(w) = \pi(\tau)(w) \). Taking \( \epsilon = q(\tau^*) \alpha^* \), where now \( \alpha^* = \inf_{x, \tau \in \mathcal{T}} \alpha_1(\tau, \tau^*; x) \), and explicitly introducing all of these random variables into the algorithm, we obtain a slightly different formulation, as follows.

Algorithm 4.1

1. **Begin at time** \( t = t_0 \) in some arbitrary state \((x_{t_0}, \tau_{t_0}) \in \mathcal{X} \times \mathcal{T}\).
2. **If** \( v_t \leq \alpha^* \) and \( v_t = \tau^* \), **perform step** 2B, **else** **perform step** 2C.
3. **Set** \( \tau_{t+1} = \tau^* \) and \( x_{t+1} = w_t \).
4. **If** \( v_t \leq \alpha(\tau_t, g(\tau_t, v_t); x_t) \), **set** \( \tau_{t+1} = g(\tau_t, v_t) \), **else** **set** \( \tau_{t+1} = \tau_t \). **And**, **if** \( u_t \leq \alpha_2(x_t, f(x_t, u_t); \tau_{t+1}) \), **set** \( x_{t+1} = f(x_t, u_t) \), **else** **set** \( x_{t+1} = x_t \).
5. **Set** \( t = t + 1 \) and **return to** **step** 2A.

This algorithm clearly needs some explanation. Our \( \epsilon \)-coin tells us whether or not all of an infinite number of chains would coalesce at this time and this occurs whenever we propose a move to \( \tau^* \) that would be accepted by any chain no matter what the current state is. This occurs if \( v_t = \tau^* \) and if \( v_t \leq \alpha^* \) and thus occurs with the probability \( \epsilon \) given in (10). If we do couple, then we set \( \tau_{t+1} = \tau^* \) and generate \( x_t \) from \( \pi(\tau^*)(\cdot) \). This latter is achieved by setting \( x_{t+1} = w_t \) in step 2B. If we do not couple, then we first update the temperature using a standard tempering update based upon \( v_t \) and \( v_t \), and then update the state using an MCMC update based upon \( u_t \) and \( u_t \), as in step 2C.

In terms of Algorithm 4.1, we obtain by inspection the following result.

**Proposition 4.1.** Suppose that two chains \((X, \tau_X)\) and \((Y, \tau_Y)\) each follow Algorithm 4.1. Suppose further that they each use the same random variable sequence \( \{u_t, u_t, u_t, u_t, w_t, w_t, \ldots\} \). Then the two chains \((X, \tau_X)\) and \((Y, \tau_Y)\) are coupled essentially by definition. Furthermore, if \( X_t = Y_t \), then \( X_{t+k} = Y_{t+k} \) for all \( k \geq 0 \).

The final concept we require in the context of our perfect simulation algorithm is the idea of dominance. Stochastic dominance (see e.g. Meyn and Tweedie 1993) is defined in terms of some partial ordering. We shall adopt the following specialised definition; it corresponds to the partial ordering on \( T \) where \( \tau^* \leq \tau \) for all \( \tau \in \mathcal{T} \), but other elements of \( \mathcal{T} \) are not comparable.

**Definition 4.1.** We shall say that the chain \((X, \tau_X)\) dominates \((Y, \tau_Y)\) if whenever \( \tau_X = \tau^* \), then also \( \tau_Y = \tau^* \).

Later in this paper, we shall show how to construct a particular dominating process which provides a very natural scheme for the construction of a perfect simulation algorithm in which the point of convergence of a forward simulation scheme is essentially pre-determined. However, before we introduce the tempering-based scheme, we introduce a very general results that allows us to rewrite any general CFTP scheme as a forward-time simulation algorithm, avoiding the need to repeatedly run the CFTP scheme from starting points further and further back into the past.
5. 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 obtain 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 is also the essential underlying idea of the “read-once CFTP” of Wilson (2000). Here, we present the following theorem which both formalises and generalises the result, so that it may be generically applied to any uniformly ergodic simulation scheme.

**Theorem 5.1.** Consider a Markov chain \{X_n\} on a state space \mathcal{X}, with stationary distribution \pi(\cdot). Suppose the chain’s transition probabilities \mathcal{P}(x, \cdot) satisfy the uniform minorisation condition \mathcal{P}^{k_0}(x, \cdot) \geq \nu(\cdot) for all x \in \mathcal{X}, some positive integer \( k_0 \), some \( \epsilon > 0 \), and some probability measure \nu(\cdot) on \mathcal{X}. Let \mathcal{R}^{k_0}(x, \cdot) = (1 - \epsilon)^{-1}[\mathcal{P}^{k_0}(x, \cdot) - \nu(\cdot)] be the residual kernel. Let \{Y_n\} be a Markov chain with initial distribution \nu i.e., \( Y_0 \sim \nu(\cdot) \), and transition probabilities \mathcal{R}^{k_0}(x, \cdot). If we generate \( T \sim \text{geometric}(\epsilon) \) independently of the chain \{Y_n\}, then

\[ Y_T \sim \pi(\cdot). \]

That is, the chain \{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 \{X_n\}, with coupling and regenerations defined as in Algorithm 4.1. Let

\[ T = \min\{t \geq 0 : \{X_n\} \text{regenerates at time } -k_0 t\}. \]

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

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 \( \pi(\cdot) \), and the result follows. \( \square \)

Theorem 5.1 thus says that, whenever we have a uniformly ergodic Markov chain, we can generate a perfect draw from stationary simply by running the residual chain for a \text{Geometric}(\epsilon) time, where \( \epsilon \) is the uniform minorisation parameter. We emphasise that Theorem 5.1 applies to any uniformly ergodic chain, whether or not it is related to simulated tempering.

Recall now that \text{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 \( \epsilon \) 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. Finally, condition (3) depends upon the convergence time of the underlying Markov chain, and will generally hold for sufficiently rapidly mixing chains.

We shall see in the next section that, for the simulated tempering chains that we shall consider, conditions (1), (2), and (3) will indeed hold. Hence, Theorem 5.1 will indeed be very useful in this case.

6. A Perfect Simulation Algorithm Using Simulated Tempering

Suppose now that we run two coupled simulated tempering chains, \( \{X, \tau_X\} \) and \( \{Y, \tau_Y\} \). Clearly, if \( \nu_t = r^* \) and \( \nu_t^0 \leq \alpha^* \), then these two chains will coalesce at time \( t + 1 \), since both will be in state

Simulated Tempering

11
Brooks, Fan and Rosenthal

$(w_t, \tau^*)$ at this time; see Algorithm 4.1 above. Further, these chains will then remain coalesced for all times beyond $t + 1$, by Proposition 4.1.

The CFTP scheme described in Algorithm 2.1 can therefore be used to produce a perfect simulation scheme. However, a far more efficient scheme may be obtained by considering the following dominating process. Suppose that we have a process $\{Z_t\}$ which can exist in one of two states $\tau^*$ and some other state $\tau^0$, say. Suppose that the transition mechanism for this process is such that at time $t$ if $v_t = \tau^*$ and $v_0^0 \leq \alpha^*$, then $Z_{t+1} = \tau^*$ else $Z_{t+1} = \tau^0$. Clearly, if we use the same $v_0^0$ and $v_t$ sequences, then this and our simulated tempering scheme are coupled and the $Z$ process dominates the $(X, \tau)$ process in the sense described in Definition 4.1 i.e., $\tau_t = \tau^*$ whenever $Z_t = \tau^*$. To see this simply note that $Z_t = \tau^*$ implies that $v_{t-1} = \tau^*$ and $v_0^0 \leq \alpha^*$ which in turn implies that $\tau_t = \tau^*$ and $x_t = w_t$, see Algorithm 4.1.

Clearly, the $Z_t$ process dominates all $(X_t, \tau_t)$ chains based upon the same $v_t$ and $v_0^0$ sequences, irrespective of their starting point. Thus, if $Z_t = \tau^*$ then all coupled chains simultaneously move to $(w'_t, \tau^0)$ and therefore coalesce at this time. Proposition 4.1 ensures that they will all follow a single path from time $t + 1$ onward and this means that in the CFTP scheme described in Algorithm 2.1, rather than following an infinite number of chains started from every possible starting point, we need only monitor the $Z_t$ sequence to determine the starting time $-T$ from which to run a single tempering chain.

The implementation is further simplified by the fact that the $Z_t$ sequence is time-reversible (in fact the current state is independent of any other) and also by the observation that at any time $t$ the distribution of the next time that $Z_t$ jumps to state $\tau^*$ follows a geometric distribution with parameter $\epsilon$, as in Theorem 5.1. Indeed, this follows directly from the fact that the $v_t$ and $v_0^0$ are both independent of one another and across time. Therefore, for all $t,$

$$
P(Z_{t+1} = \tau^*) = P(v_t = \tau^*)P(v_0^0 \leq \alpha^*) = q_1(\tau^*)\alpha^* = \epsilon
$$

Under the CFTP scheme of Algorithm 2.1, we would start by generating $v_{-1}$ and $v_0^0$ to obtain $Z_0$ and would stop if $Z_0 = \tau^*$ since this means that all chains would be in state $(w_0, \tau^*)$ at this time. This would only occur if $v_{-1} = \tau^*$ and $v_0^0 \leq \alpha^*$. If this is not the case, we would generate $v_{-2}$ and $v_0^0$ and see whether or not the chains have coalesced by time $t = 0$. For this to occur, the chains would have to coalesce either at time $t = 0$ (which we know doesn’t happen since either $v_{-1} \neq \tau^*$ or if $v_{-1} = \tau^*$, then $v_{0}^0 > \alpha^*$), or at time $t = -1$, which can only occur if $v_{-2} = \tau^*$ and $v_{0}^0 \leq \alpha^*$. We repeat this process going further and further back in time until we find a $t$ such that $v_t = \tau^*$ and $v_0^0 \leq \alpha^*$.

Alternatively, we could use the forward-time modification of Theorem 5.1. That is, rather than sampling the $v_t$ and $v_0^0$ at times $t = -1, -2, \ldots$, we draw $T \sim \text{Geometric}(\epsilon)$, set $v_0 = \tau^*$ and sample $v_0^0 \sim U[0, \alpha^*]$. Then, for $0 < t < T$, we draw $u_t \sim q_1(v)$ and, if $v_t = \tau^*$ we generate $v_0^0 \sim U(\alpha^*, 1]$ else we generate $v_0^0 \sim U[0, 1]$. We therefore obtain the following more efficient perfect simulation scheme.

**Algorithm 6.1**

**Step 1.** Draw a random variable $T \sim \text{Geometric}(\epsilon)$, and set $t = 0$.

**Step 2.** Draw $w_0 \sim \pi_0^0(w)$, set $x_{t+1} = w_t$ and $\tau_{t+1} = \tau^*$. Set $t = t + 1$.

**Step 3.** If $t = T$, stop (and return $x_T$). Otherwise, draw $v_0^0, u_0^0 \sim U(0, 1)$, $u_t \sim q_2(u)$ and $v_t \sim q_1(v)$. If $v_t = \tau^*$, then set $v_0^0 = \alpha^* + (1 - \alpha^*)v_0^0$.

**Step 4.** If $v_0^0 \leq \alpha_2[x_t, f(x_t, u_t), \tau_{t+1}]$, set $\tau_{t+1} = g(\tau_t, v_t)$, else set $\tau_{t+1} = \tau_t$. Then, if $u_0^0 \leq \alpha_2[x_t, f(x_t, u_t), \tau_{t+1}], x_{t+1} = f(x_t, u_t)$, else set $x_{t+1} = x_t$.

**Step 5.** Set $t = t + 1$, and return to Step 3.

Intuitively, Algorithm 6.1 is simply a forward-time version (as in Theorem 5.1) of CFTP with a simulated tempering chain, using the dominance idea as above. The final state $x_T$ will be the desired draw from $\pi(x, \tau)$. 


Remark. Algorithm 6.1 requires a value of the minorising parameter $\epsilon$. However, as noted at the end of Section 2, it is acceptable (though suboptimal) to use any underestimate of $\epsilon$. This allows for the possibility of using various numerical techniques to estimate $\epsilon$, 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 $\epsilon$ analytically even for complicated Markov chains, see for example Rosenthal (1995b) and Rosenthal (1996); but this can be difficult in general.

7. 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 6.1 is distributed as $\text{Geometric}(\epsilon)$. We shall see below that, since the Markov chains used by Algorithm 6.1 satisfy a uniform minorisation condition with parameter $\epsilon$, the distance to stationarity is also going down essentially like $\text{Geometric}(\epsilon)$. This implies that Algorithm 6.1 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 6.1 has in common with various convergence rate studies (e.g. Rosenthal 1995b, Rosenthal 1996, Cowles and Rosenthal 1998, Douc et al 2002) the need to compute a minorisation parameter $\epsilon$. Various techniques (both analytic and numerical) employed to compute $\epsilon$ for convergence rate studies, can also be employed to implement Algorithm 6.1. The results below further indicate the connection between the two approaches.

Third, the convergence rate studies provide context for Algorithm 6.1. 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 1995b; 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 $\epsilon$ must work from all states $x$. 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 6.1.

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 $\mathcal{X}$ by

$$||\mu(\cdot) - \nu(\cdot)||_{TV} = \sup_{A \subseteq \mathcal{X}} |\mu(A) - \nu(A)|.$$  

We shall be particularly interested in bounding the distance

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

where $\mathcal{P}^k[(x, \tau), \cdot]$ is the distribution of the chain after $k$ steps when started at $(x, \tau)$, 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 1995b; 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 (1995a).

**Proposition 7.1.** Let $\mathcal{P}[(x, \tau), \cdot]$ 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 $(k_0, \epsilon, \nu)$-small. Then for any starting point $(x, \tau) \in \mathcal{X} \times T$, and any positive integer $k$, we have

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

where $[r]$ denotes the greatest integer not exceeding $r$. 

Simulated Tempering 13
We now consider the question of convergence rates for simulated tempering. Combining Proposition 7.1 with Proposition 3.1, we immediately obtain the following.

**Proposition 7.2.** Suppose that our simulated tempering chain has one constituent temperature \( \tau^* \) so that \( S^* = \{(x, \tau^*) : x \in X\} \) is \((k_2, \epsilon_2, \rho^-)\)-small. Suppose further that \( P^k(\{x, \tau\}, S^*) \geq \epsilon_1 \) for all \( (x, \tau) \in X \times T \). Let \( \pi(\cdot) \) be a stationary distribution for the chain. Then for any starting point \( (x, \tau) \in X \times T \), and any positive integer \( k \), we have

\[
\|P^k((x, \tau), \cdot) - \pi(\cdot)\|_{TV} \leq (1 - \epsilon_1^2)^{k/(k_1 + k_2)}
\]

where \( [r] \) 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 7.3.** Suppose a simulated tempering chain has one constituent temperature \( \tau^* \) so that \( S^* \) is an atom. Suppose further that \( P^k(\{x, \tau\}, S^*) \geq \epsilon_1 \) for all \( (x, \tau) \in X \times T \). Then for any starting point \( (x, \tau) \in X \times T \), and any positive integer \( k \), we have

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

Similarly, we have the following specialisation if \( S^* \) is \( \delta \)-uniform.

**Proposition 7.4.** Suppose a simulated tempering chain has one constituent temperature \( \tau^* \) so that \( S^* \) is \( \delta \)-uniform. Suppose further that \( P^k(\{x, \tau\}, S^*) \geq \epsilon_1 \) for all \( (x, \tau) \in X \times T \). Then for any starting point \( (x, \tau) \in X \times T \), and any positive integer \( k \), we have

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

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

**Corollary 7.1.** Suppose a simulated tempering chain has one constituent temperature \( \tau^* \) so that \( S^* \) is i.i.d.-like. Suppose further that \( P^k(\{x, \tau\}, S^*) \geq \epsilon_1 \) for all \( (x, \tau) \in X \times T \). Then for any starting point \( (x, \tau) \in X \times T \), and any positive integer \( k \), we have

\[
\|P^k((x, \tau), \cdot) - \pi(\cdot)\|_{TV} \leq (1 - \epsilon_1)^{k/k_1}
\]

Corollary 7.1 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 7.1 to Algorithm 6.1. Indeed, in Algorithm 6.1, \( X \times T \) is \( \epsilon \)-small, and \( S^* \) is i.i.d.-like, so we obtain the following.

**Theorem 7.1.** Consider the underlying simulated tempering Markov chain of Algorithm 6.1. Let \( \epsilon = q(\tau^*) \alpha^* \) and \( \alpha^* = \inf_{x \in X, \tau^*} \tau^* \). Then for any starting point \( (x, \tau) \in X \times T \), and any positive integer \( k \), the total variation distance of the Markov chain to stationarity after \( k \) iterations, satisfies

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

Theorem 7.1 says that, if we consider the simulated tempering chain underlying Algorithm 6.1, then the convergence rate of this chain is essentially Geometric(\( \epsilon \)). This is to be compared with the implementation of Algorithm 6.1 itself, which is run for a total of \( T \sim \text{Geometric}(\epsilon) \) iterations. That is, Algorithm 6.1 manages to obtain a perfect sample from \( \pi(\cdot) \), in a time comparable to the relaxation time of the underlying Markov chain.

**Remark.** 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 7.2 and 7.3 that, the more likely the chain is to jump to $S^*$, the larger we can make $e_1$ 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 $S^*$. On the other hand, the more likely the chain is to jump to $S$, the less time the chain will spend in $\tau = \tau_1$ in stationary. 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 $S^*$, between the speed of convergence to stationarity (and the corresponding running time) of Algorithm 6.1, and the usefulness of the resulting samples from stationarity, which we shall explore further in the context of the simple example in Section 9.

8. Trans-dimensional MCMC

Closely related to the simulated tempering idea are trans-dimensional (often referred to as “reversible jump”) MCMC algorithms (Norman and Filimonov 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 $\pi(x, m)$ denotes a joint distribution over both models $m \in \mathcal{M}$ and their associated parameters $(x \in \mathcal{X}_m)$. This essentially generalises the tempering scheme to the case where the stationary distributions are each defined upon separate spaces $\mathcal{X}_m$ which may be of different dimension, as first described in Section 2.1. Hence, the state space for the trans-dimensional chain is give by $\bigcup_{m} \mathcal{X}_m \times \{m\}$. We denote the dimension of $\mathcal{X}_m$ by $\ell_m$.

In the trans-dimensional case, the temperature indicator $\tau$ is replaced by what is often termed the “model indicator” $m$ and, typically, all models may be of equal interest there commonly being no real concept of a “cold” model. Jumps between models proceed similarly to those between temperatures in the tempering scheme. We shall assume a fairly general form of implementation for the reversible jump MCMC scheme, though our results apply equally well to any implementation.

8.1. The Reversible Jump MCMC Scheme

As is noted from the tempering sections, we suppose that at iteration $k$ we are in model $m_k$ and state $x_k$ and that we propose to move to model $m'$ with probability $q(m, m')$. This move is performed by drawing a random vector $v_k \in \mathcal{X}_{m'}$ from a proposal distribution $q_{m'}(v_k)$. Then, we propose to move to state $x' = g(x_k, v_k)$ in the new model and accept this proposal with probability $A([x, m], [x', m']) = \min[1, A(m, m' : x_k, v_k)]$, where

$$A(m, m' : x_k, v_k) = \frac{\pi_{m'}(x') q_{m'}(m', m) q(v')}{{\pi}_m(x) p(m) q(m, m') q(v)} - |J_g(x_k, v_k)|, \tag{12}$$

$$J_g(x, v) = \frac{\partial g(x_k, v)}{\partial (x_k, v_k)},$$

$p(m)$ denotes the prior probability associated with model $m$, $v'$ satisfies $g(g(x, v), v') = x$ and the final term in (12) denotes the Jacobian associated with the transformation from $(x_k, v_k)$ to $[g(x_k, v_k), v']$. Note the similarity here with the acceptance probability given in (11).

In practice, reversible jump and traditional MCMC moves are usually combined within a single algorithm. So that both within-model (MCMC) and between-model (reversible jump MCMC) moves are proposed at each iteration.

8.2. Extensions to the Trans-dimensional Case

All of the results presented in the previous sections can be extended to the trans-dimensional context. We begin with a redefinition of the term i.i.d-like.
Definition 8.1. For a trans-dimensional Markov chain, the set \( S^* = \{(x, m^*) : x \in X_m\} \) is i.i.d.-like if there is a probability distribution \( \pi_{m^*} \) on \( X_m\) such that
\[
P(X_{k+1} \in B | X_k = x, m_k = m, m_{k+1} = m^*) = \pi_{m^*}(B)
\]
for all \( B \subseteq X_m\), \( m \in \mathcal{M}\), and \( x \in X_m\), independently of \( x \) and \( m \).

Clearly, if there exists a model \( m^* \) with corresponding distribution \( \pi_{m^*}(x) \), from which we can sample directly, then we may arrange for \( S^* = \{(x, m^*) : x \in X_m\} \) to be i.i.d.-like. We extend Corollary 3.2 as follows.

Corollary 8.1. Suppose our simulated tempering chain has one constituent model \( m^* \) which is i.i.d.-like, and that
\[
P^k[(x, m), S^*] \geq \epsilon, \quad m \in \mathcal{M}, \ x \in X_m.
\]
Then \( P^k[(x, m), A] \geq \epsilon \pi_{m*}(A) \) for all \( m \in \mathcal{M} \) and \( x \in X_m \) and \( A \subseteq X_m \times \{m\} \). Therefore \( \{(x, m) : m \in \mathcal{M}, x \in X_m\} \) is \((k, \epsilon, \mathcal{M})\)-small, where \( \mathcal{M}(S \times \{m^*\}) = \pi_{m^*}(S) \), and \( \mathcal{M}(X_m \times \{m\}) = 0 \) for \( m \neq m^* \).

As in the fixed-dimensional case, we have
\[
\epsilon = \inf_{x \in X_m, m \in \mathcal{M}, \forall \in X_m} q(m, m^*) \alpha[(x, m), (g(x, \psi), m^*)] \\
= q(m, m^*) \alpha^*
\]

Thus, Algorithm 6.1 may be extended to the trans-dimensional context as follows.

Algorithm 8.1

Step 1. Draw a random variable \( T \sim Geometric(\epsilon) \), and set \( t = 0 \).
Step 2. Draw \( u_t \sim \pi_{m^*}(w) \), set \( x_{t+1} = u_t \) and \( m_{t+1} = \tau^* \). Set \( t = t + 1 \).
Step 3. If \( t = T \), stop (and return \( x_T \)). Otherwise, draw \( v_t^0, u_t^0 \sim U(0,1) \), \( m \sim q(m_t, m) \), \( v_t \sim q_m(u) \) and \( u_t \sim q(g(u)) \). If \( m_t = m^* \), then set \( u_t^0 = \alpha^* + (1 - \alpha^*) u_t^0 \).
Step 4. If \( u_t^0 \leq \alpha^*[(x_t, m_t), (g(x_t, v_t), m)] \) set \( m_{t+1} = m \) and \( z = g(x_t, v_t) \), else set \( m_{t+1} = m_t \) and \( z = x_t \). Then, if \( u_t^0 \leq \alpha^*[(x_t, f(z, u_t), \tau^*)] \), set \( x_{t+1} = f(x_t, u_t) \), else set \( x_{t+1} = x_t \).
Step 5. Set \( t = t + 1 \), and return to step 3.

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

Proposition 8.1. If the target distribution \( \pi(x, m) \propto L_m(data|x)p(x|m)p(m) \) is a Bayesian posterior, where \( L_m(data|x) \) denotes the likelihood under model \( m \) associated with parameters \( x \), \( p(x|m) \) denotes the prior for \( x \) under model \( m \) and \( p(m) \) denotes the corresponding prior model probability, and we take \( g(x, \psi) = (v, x) \), \( q_m(v) = \pi_{m^*}(v) \) and \( q(g(v), m) = p(v|m) \forall m \neq m^* \), then
\[
\alpha^* = \min_{m \in \mathcal{M}} \frac{p(m^*)q(m^*, m)}{p(m)q(m, m^*)L_m^*},
\]
where \( L_m^* \) denotes the value of the likelihood associated with model \( m \) evaluated at the corresponding maximum likelihood estimate.

This result is particularly easy to apply when we have a nested model structure. If we let \( m_s \) denotes the maximal (or saturated) model, with \( X_m = \bigcup_{m \in \mathcal{M}} X_m \), then the minimum in (13) is achieved when \( m = m_s \), since \( L_m \geq L_m \forall m \in \mathcal{M} \), by definition of \( m_s \) as the superset of all other models in \( \mathcal{M} \). We illustrate the application of this trans-dimensional perfect simulation scheme with the aid of a simple example.
8.3. Example: Autoregressive Time Series
Suppose that we have a univariate time series $y_1, \ldots, y_t$ which we believe can be described by an autoregressive (AR) process of order $m$, i.e.

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

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

$$z = y^k - Y^k a^k,$$

where $y_0$ and $y^k$ are formed by partitioning the data vector $y$ into, respectively, the first $k$ values and the remainder and $a$ and $Y^k$ take appropriate forms. Now suppose that the model order $m$ is unknown, then Ehlers and Brooks (2001) demonstrate how to construct a reversible jump MCMC scheme for exploring the corresponding posterior distribution $\pi(x, m)$ where, under model $m = 1, \ldots, m_{\text{max}}$, $x \in \mathbb{R}^m$ (i.e., $x_m = \mathbb{R}^m$, here). This scheme may be augmented by allowing $m = 0$ 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 $\pi(\sigma^2|m = 0)$ is a member of the inverse gamma family and can therefore be sampled directly from its stationary distribution. Thus, $m = 0$ corresponds to our “hot” model $m^*$ and $S^* = (\sigma^2, m^*)$ is i.i.d.-like.

We take vague independent $N(0, \sigma^2)$ 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 $m \in \{0, \ldots, m_{\text{max}}\}$ with equal probability. If we $m \neq 0$, then we propose new parameter values by simulating from the prior. However, if $m = 0$, then we draw a new value of $\sigma^2$ from the corresponding full conditional, $\pi_0(\sigma^2)$. Then, by Corollary 8.1 and the fact that all of the models $m \in M$ are nested, we have that

$$\alpha^* = \frac{p(m^*)}{p(m_{\text{max}})L_m},$$

where $L_m$ is the likelihood evaluated at the MLE:

$$\hat{\sigma}^2 = \tilde{a}^T \tilde{a}, \quad \tilde{a} = C(Y^{m_{\text{max}}})^T y/\hat{\sigma}^2; \quad \text{and} \quad C^{-1} = (Y^{m_{\text{max}}})^T Y^{m_{\text{max}}}/\hat{\sigma}^2.$$

9. Example I: Sampling from a Beta($\alpha, \beta$)
Following Green and Murdoch (1998), suppose we wish to obtain a perfect sample from a beta($\alpha, \beta$) distribution, with $\alpha, \beta > 1$. 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 $X_0 = \chi_1$ and so, from (8) we have

$$\epsilon = \inf_{\tau \in [0, 1]} \frac{q(\tau, 1)}{q(1, 0)}$$

Clearly, this infimum is obtained when $\tau = 0$ and, if we take $q_1(0) = 1/2$ (i.e., $q(1, 0) = q(0, 1) = 1/2$) then $\epsilon = \alpha^*/2$, where

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

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

$$\alpha^* = \min \left(1, \frac{cw_1}{1 - w_1} \right),$$
since \( w_0 = 1 - w_1 \) and where

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

We might then set \( w_1 = w_0 = 1/2 \) for example and use Algorithm 6.1 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 \( \pi(x, \tau) \). If at time zero, \( \tau_0 \neq 0 \), we might run the perfect simulation algorithm again to obtain another draw from \( \pi(x, \tau) \). 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 stationary 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

\[
N = \frac{1}{1 - w_1 \epsilon} = 2 \cdot \frac{1}{(1 - w_1) \min(1, cw_1/(1 - w_1))} = \frac{2}{\min(1 - w_1, cw_1)}
\]

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 \( \tau_0 \) 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

---

**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.
with a total number of expected iterations equal to 20.520. 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 is likely to be increasingly dramatic.

10. 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 \( \lambda \) denote the probability of a particular animal being recovered given that it has died. We also let \( \phi \) denote the probability that an adult survives from one year to the next, but let \( \phi_1 \) 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

\[
L(R, m|\phi_1, \phi, \lambda) = c \Delta \prod_{i=1}^{I} \prod_{j=i}^{J} p_{ij}^{m_{ij}}
\]

where

\[
p_{ij} = \left\{ \begin{array}{ll}
\lambda \phi_1 & j = i, \\
\lambda \phi_1 \phi_{j-i-1} & j = i + 1, \ldots, J
\end{array} \right.
\]

\( \phi = 1 - \phi_1, \quad \phi_0 = 1 - \phi_1, \quad \Delta \) 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 \( q_i = 1 - \sum_{j=1}^{J} p_{ij} \) 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 \( u_i = R_i - \sum_{j=1}^{J} m_{ij} \) 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 \( \Delta \) is a function of all of the model parameters. Finally,

\[
c = \prod_{i} \left[ R_i! / (u_i! \prod_{j} m_{ij}!) \right].
\]

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

\[
\pi_0(\phi_1, \phi, \lambda|R, m) \propto \Delta \prod_{i=1}^{I} \prod_{j=i}^{J} p_{ij}^{m_{ij}}.
\]
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 $q(0,1) = q(1,0) = 1/2$.

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

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

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 6.1 we find that with $w_1 = L^*$ we 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 $\epsilon$ decreases to approximately $5 \times 10^{-6}$, corresponding to a geometric distribution with a mean of around 200,000 iterations.

![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) $\phi_1$, (b) $\phi$ and (c) $\lambda$.](image)

In this case, since $w_1$ is so small that $w_0 \approx 1$, multiplying $w_1$ by some small factor $\alpha$ 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 $w_1$ 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 $w_1$ above, we observed the simulation output summarised by the histograms in Figure 2. The corresponding posterior means (and standard deviations) for $\phi_1$, $\phi$ and $\lambda$ 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).

11. Discussion

In this paper, we have introduced a new algorithm (Algorithm 6.1) for producing perfect samples from a target distribution $\pi(\cdot)$. The algorithm is a version of CFPT using simulated tempering MCMC, but converted to a forward-time algorithm using Theorem 5.1. We also extended these ideas to the trans-dimensional MCMC context, making the connection between the simulated tempering
and reversible jump MCMC schemes (Algorithm 8.1). We related our perfect simulation scheme to the theory of rigorous bounds on Markov chain convergence rates.

We have made several observations that simplify the implementation of the methods, including the connection to the maximum likelihood value, and the possibility of under-estimating \( \epsilon \). We have illustrated our techniques on several different examples, including one involving a real data set.

We feel that our results demonstrate the usefulness and applicability of our Algorithm 6.1, and that this algorithm can be applied more widely. We look forward to further applications in the future.

**Acknowledgments.** 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 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.

**References**


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


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


