TOWARDS OPTIMAL SCALING OF METROPOLIS-COUPLED MARKOV CHAIN MONTE CARLO

YVES F. ATCHADÉ, GARETH O. ROBERTS, AND JEFFREY S. ROSENTHAL

Abstract. We consider optimal temperature spacings for Metropolis-coupled Markov chain Monte Carlo (MCMCMC) and Simulated Tempering algorithms. We prove that, under certain conditions, it is optimal (in terms of maximising the expected squared jumping distance) to space the temperatures so that the proportion of temperature swaps which are accepted is approximately 0.234. This generalises related work by physicists, and is consistent with previous work about optimal scaling of random-walk Metropolis algorithms.

1. Introduction

The Metropolis-coupled Markov chain Monte Carlo (MCMCMC) algorithm [9], also known as *parallel tempering* or the *replica exchange method*, is a version of the Metropolis-Hastings algorithm [20, 12] which is very effective at sampling from multi-modal densities. The algorithm works by simulating multiple copies of the target (stationary) distribution, each at a different temperature. Through a swap move, the algorithm allows copies at lower (slower-mixing) temperatures to borrow information from copies at higher (faster-mixing) temperatures, to help them mix faster (in particular, to escape from local modes). The performance of MCMCMC depends crucially on the temperatures used for the different copies. There is an ongoing discussion (especially in the Physics literature) about how to best select these temperatures. In this paper, we show that this question can be partially addressed using the *optimal scaling* framework initiated in [23].

MCMCMC can be described as follows. We are interested in sampling from a target probability distribution $\pi(\cdot)$ having (complicated, probably multimodal, and possibly un-normalised) target density $f_d(\mathbf{x})$ on some state space $\mathcal{X}$, which is an open subset of $\mathbf{R}^d$ for some (large) dimension d. We define a sequence of associated *tempered* distributions $f_d^{(\beta_j)}(\mathbf{x})$, where $0 \le \beta_n < \beta_{n-1} < \ldots < \beta_1 < \beta_0 = 1$ are the selected *inverse temperature* values, subject to the restriction that $f_d^{(\beta_0)}(\mathbf{x}) = f_d^{(1)}(\mathbf{x}) = f_d(\mathbf{x})$. (Usually the densities $f_d^{(\beta_j)}$ are simply *powers* of the original density,

Date: July 30, 2009; last revised May 2010.

2000 *Mathematics Subject Classification.* 60J10, 65C05.

Key words and phrases. Metropolis-Coupled MCMC, Simulated tempering, Optimal scaling.

Y. Atchadé: Department of Statistics, University of Michigan, 1085 South University, Ann Arbor, 48109, MI, United States. *E-mail address:* yvesa@umich.edu.

G. O. Roberts: Department of Statistics, University of Warwick, Coventry, CV4 7AL UK. *E-mail address:* gareth.o.roberts@warwick.ac.uk.

J. S. Rosenthal: Department of Statistics, University of Toronto, Toronto, Ontario, Canada, M5S 3G3. *E-mail address:* jeff@math.toronto.edu. Partially supported by NSERC of Canada.

i.e. $f_d^{(\beta_j)}(\mathbf{x}) = (f_d(\mathbf{x}))^{\beta_j}$; in this case, if $\mathcal{X}$ has infinite volume, then we require $\beta_n > 0$.)

MCMCMC proceeds by running one chain at each of the $n+1$ values of β. It has state space $\mathcal{X}^{n+1}$, with unnormalised stationary density $\prod_{j=0}^{n} f_d^{(\beta_j)}(\mathbf{x}_j)$, where each $\mathbf{x}_j$ corresponds to a chain at the fixed inverse temperature β_j having stationary density $f_d^{(\beta_j)}(\mathbf{x}_j)$. In particular, the β_0 chain has stationary density $f_d^{(\beta_0)}(\mathbf{x}_0) = f_d(\mathbf{x}_0)$, so that after a long run the samples $\mathbf{x}_0$ should approximately correspond to the density $f_d(\mathbf{x})$ which is the density of actual interest. The hope is that the chains corresponding to smaller β (i.e., to higher temperatures) can mix more easily, and then can "lend" this mixing information to the β_0 chain, thus speeding up convergence of the $\mathbf{x}_0$ values to the density $f_d(\mathbf{x})$.

MCMCMC alternates between two different types of dynamics. On some iterations, it attempts a *within temperature move*, by updating each $\mathbf{x}_j$ according to some type of MCMC update (e.g., a usual random-walk Metropolis update) for which $f_d^{(\beta_j)}$ is a stationary density. On other iterations, it attempts a *temperature swap*, which consists of choosing two different inverse temperatures, say β_j and β_k, and then proposing to *swap* their respective chain values, i.e. to interchange the current values of $\mathbf{x}_j$ and $\mathbf{x}_k$. This proposed swap is then accepted according to the usual (symmetric) Metropolis algorithm probabilities, i.e. it is accepted with probability

$$\min\left(1,\ \frac{f_d^{(\beta_j)}(\mathbf{x}_k)\, f_d^{(\beta_k)}(\mathbf{x}_j)}{f_d^{(\beta_j)}(\mathbf{x}_j)\, f_d^{(\beta_k)}(\mathbf{x}_k)}\right), \tag{1}$$

otherwise it is rejected and the values of $\mathbf{x}$ are left unchanged. (The rejected values are normally discarded, though it is sometimes possible to make additional use of them [11, 8, 6].)

Such algorithms lead to many interesting questions and have been widely studied, see e.g. [9, 14, 15, 21, 7, 16, 18, 5, 29, 30]. This paper will concentrate on the specific question of optimising the choice of the β values to achieve maximal efficiency in the temperature swaps. Specifically, suppose we wish to design the algorithm such that the chain at inverse temperature β will propose to swap with another chain at inverse temperature $\beta + \epsilon$. What choice of ϵ is best?

Obviously, if ϵ is very large, then such swaps will usually be rejected. Similarly, if ϵ is very small, then such swaps will usually be accepted, but will not greatly improve mixing. Hence, the optimal ϵ is somewhere between the two extremes (this is sometimes called the "Goldilocks Principle"), and our task is to identify it.

Our main result (Theorem 1) will show that, under certain assumptions, it is optimal (in a sense to be defined later on) to choose the spacing ϵ such that the probability of such a swap being accepted is equal to 0.234. This is the same optimal acceptance probability derived previously for certain random-walk Metropolis algorithms [23, 25], and generalises some results in the physics literature [14, 21]. It also has connections (Section 5.1) to the Dirichlet form of an associated temperature process.

We shall also consider (Section 4) the related *Simulated Tempering* algorithm, and shall prove that under certain conditions the 0.234 optimal acceptance rate applies there as well. We shall also compare (Corollary 1) Simulated Tempering to MCMCMC, and see that in a certain sense the former is exactly *twice* as efficient as the latter, but there are various mitigating factors so the comparison is far from clear-cut.

1.1. **Toy Example.** To illustrate the importance of temperature spacings, and of measuring the "influence" of hotter chains on colder ones, we consider a very simple toy example. Suppose the state space consists of just 101 discrete points, $\mathcal{X} = \{0, 1, 2, 3, \ldots, 100\}$, with (un-normalised) target density (with respect to counting measure on $\mathcal{X}$) given by $\pi\{x\} = 2^{-x} + 2^{-(100-x)}$ for $x \in \mathcal{X}$. That is, π has two modes, at 0 and 100, with a virtually insurmountable barrier of low probability states between them (Figure 1, top-left). Let the tempered densities be given (as usual) by powers of π, i.e. $f_d^{(\beta)}(\mathbf{x}) = \Big(\pi(\mathbf{x})\Big)^{\beta}$. Thus, for very small β, the insurmountable barrier is nicely removed (Figure 1, top-middle). Can we make use of such tempered densities to improve convergence of the original chain to π?

To be specific, suppose each fixed-β chain is a random-walk Metropolis algorithm with proposal kernel $q(x, x+1) = q(x, x-1) = 1/2$. Thus, the cold (large β) chains are virtually incapable of moving from state 0 to state 100 or back (Figure 1, top-left), but the hot (small β) chains have no such obstacle (Figure 1, top-middle). The question is, what values of the inverse temperatures β will best allow an MCMCMC algorithm to benefit from the rapid mixing of the hot chains, to provide good mixing for the cold chain?

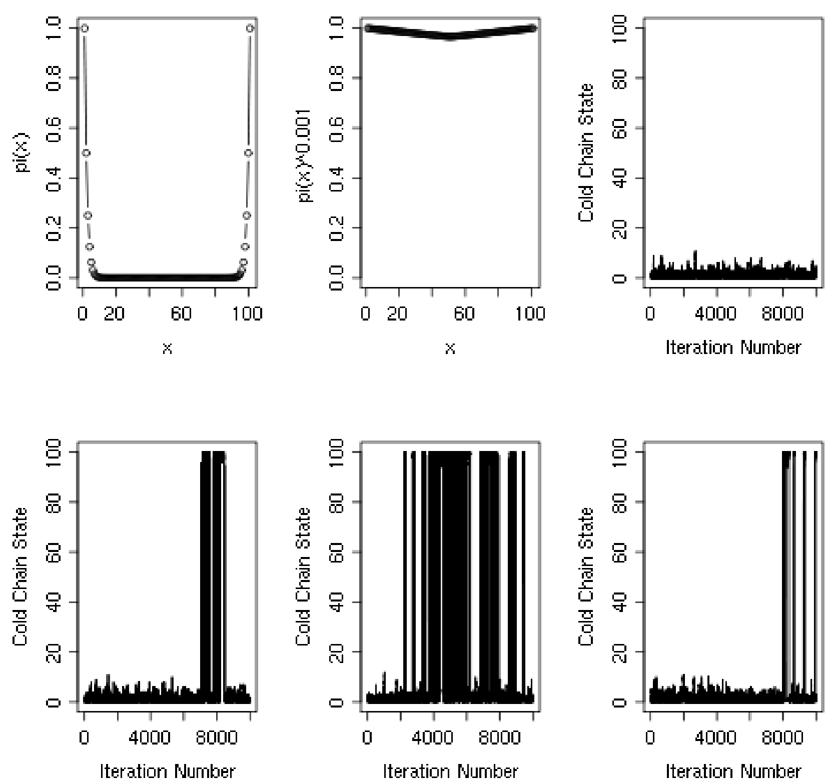

Figure 1: The toy example's target density (top-left), tempered density at inverse-temperature $\beta = 0.001$ (top-middle), random-walk Metropolis run (top-right), and trace plots of the cold chain over 10,000 full iterations of

MCMCMC runs with two temperatures (bottom-left), ten temperatures (bottom-middle), and fifty temperatures (bottom-right).

To test this, we ran an MCMCMC algorithm for 10,000 full iterations (each consisting of one update at each inverse-temperature, plus one attempted swap of adjacent inverse-temperature values), in each of four settings: with just one inverse-temperature, i.e. an ordinary MCMC algorithm with no temperature swaps (Figure 1, top-right); with two inverse-temperatures, 1 and 0.001 (Figure 1, bottom-left); with ten inverse-temperatures, $0.001^{j/9}$ for $j = 0, 1, 2, \ldots, 9$ (Figure 1, bottom-middle); and with fifty inverse-temperatures, $0.001^{j/49}$ for $j = 0, 1, 2, \ldots, 49$ (Figure 1, bottom-middle). We started each run with all chains at the state 0, to investigate the extent to which they were able to mix well and effectively sample from the equally-large mode at state 100.

The results of our runs were that the ordinary MCMC algorithm with no temperature swaps did not traverse the barrier at all, and just stayed near the state 0 (Figure 1, top-right). Adding one additional temperature improved this somewhat (Figure 1, bottom-left), and the cold chain was now able to sometimes sample from states near 100, but only occasionally. Using ten temperatures improved this greatly and led to quite good mixing of the cold chain (Figure 1, bottom-middle). Perhaps most interestingly, using lots more temperatures (fifty) did not improve mixing further, but actually made it much worse (Figure 1, bottom-right), despite the greatly increased computation time (since each temperature requires its own chain values to be updated at each iteration).

So, we see that in this example, ten geometrically-spaced temperatures performed much better than two temperatures (too few) or fifty temperatures (too many). Intuitively, with only two temperatures, it is too difficult for the algorithm to effectively swap values between adjacent temperatures (indeed, only about 5% of proposed swaps were accepted). By contrast, with fifty temperatures, virtually all swaps are accepted, but too many swaps are required before the rapidly-mixing hot chains can have much influence over the values of the cold chain, so this again does not lead to an efficient algorithm. Best is the compromise choice of ten temperatures. That choice makes the temperature differences small enough so swaps are easily accepted, but still large enough that they lead to efficient interchange between hot and cold chains and thus to efficient convergence of the cold chain to π. This illustrates that we would like the hot and cold chains' values to be able to influence each other effectively, which requires lots of successful swaps each of significant distance in the inverse-temperature domain.

This leads to the question of how to select the number and spacing of the inverse-temperature values for a given problem, which is the topic of this paper. To maximise the influence of the hot chain on the cold chain, we want to maximise the effective speed with which the chain values move along in the inverse-temperature domain. We shall do this by means of the expected squared jumping distance (ESJD) of this influence, defined below. Our main result (Theorem 1 below) says that to maximise ESJD, it is optimal (under certain assumptions) to select the inverse-temperatures so that the probability of accepting a proposed swap between adjacent inverse-temperature values is approximately 23%.

2. Optimal Temperature Spacings for MCMCMC

In this section, we consider the question of optimal temperature choice for MCMCMC algorithms. To fix ideas, we consider the specific situation in which the algorithm is proposing to swap the chain values at two specific inverse temperatures, namely β and $\beta+\epsilon$, where $\beta, \epsilon > 0$ and $\beta+\epsilon \leq 1$. We shall consider the question of what value of ϵ is optimal in terms of maximising the influence of the hot chain on the cold chain. We shall focus on the asymptotic situation in which the chain has already converged to its target stationary distribution, i.e. that $\mathbf{x} \sim \prod_{j=0}^{n} f_d^{(\beta_j)}$.

To measure this precisely, let us define $\gamma = \beta + \epsilon$ if the swap is accepted, or $\gamma = \beta$ if the swap is rejected. Then γ is an indication of where the β chain's value has traveled to, in the inverse temperature space. That is, if the proposed swap is accepted, then the value moves from β to $\gamma = \beta + \epsilon$ for a total inverse-temperature distance of $\gamma - \beta = \epsilon$, while if the swap is rejected then it does not move at all, i.e. the distance it moves is $\gamma - \beta = 0$. Hence, $\gamma - \beta$ indicates the extent to which the swap proposal succeeded in moving different x values around the β space, which is essential for getting benefit from the MCMCMC algorithm. So, the larger the magnitude of $\gamma - \beta$, the more efficient are the swap moves at providing mixing in the temperature domain. We therefore define the "optimal" choice of ϵ to be the one which maximises the stationary (i.e. asymptotic) expected squared jumping distance (ESJD) of this movement, i.e. which maximises

$$ESJD \;=\; \mathbb{E}_\pi\big[(\gamma-\beta)^2\big] \;=\; \epsilon^2 \times \mathbb{E}_\pi[\mathbf{P}(\text{swap accepted})] \;\equiv\; \epsilon^2 \times ACC\,, \tag{2}$$

where from (1),

$$ACC \;=\; \mathbb{E}_\pi[\mathbf{P}(\text{swap accepted})] \;=\; \mathbb{E}_\pi\left[\min\left(1,\; \frac{f_d^{(\beta_j)}(\mathbf{x}_k)\, f_d^{(\beta_k)}(\mathbf{x}_j)}{f_d^{(\beta_j)}(\mathbf{x}_j)\, f_d^{(\beta_k)}(\mathbf{x}_k)}\right)\right] \tag{3}$$

and where the expectations are with respect to the stationary (asymptotic) distribution $\mathbf{x} \sim \prod_{j=0}^{n} f_d^{(\beta_j)}$.

We shall also consider the generalisation of (2) to

$$ESJD(h) \;=\; \mathbb{E}_\pi\big[(h(\gamma) - h(\beta))^2\big] \tag{4}$$

for some $h : [0,1] \to \mathbb{R}$; then (2) corresponds to the case where h is the identity function. Of course, (2) and (4) represent just some possible measures of optimality, and other measures might not necessarily lead to equivalent optimisations; for some discussion related to this issue see Section 5.

To make progress on computing the optimal ϵ, we restrict (following [23, 25]) to the special case where

$$f_d(\mathbf{x}) = \prod_{i=1}^{d} f(x_i)\,, \tag{5}$$

i.e. the target density takes on a special product form. (Although (5) is a very restrictive assumption, it is known [23, 25] that conclusions drawn from this special case are often approximately applicable in much broader contexts.) We also assume that the tempered distributions are simply *powers* of the original density (which is the usual case), i.e. that

$$f_d^{(\beta)}(\mathbf{x}) \;=\; \prod_{i=1}^{d} f^{(\beta)}(x_i) \;\equiv\; \prod_{i=1}^{d} \big(f(x_i)\big)^{\beta}. \tag{6}$$

Intuitively, as the dimension d gets large, so that small changes in β lead to larger and larger changes in $f_d^{(\beta)}$, the inverse-temperature spread ϵ must decrease to preserve a non-vanishing probability of accepting a proposed swap. Hence, similar to [23, 25], we shall consider the limit as $d \nearrow \infty$ and correspondingly $\epsilon \searrow 0$. To get a non-trivial limit, we take

$$\epsilon = d^{-1/2}\ell \tag{7}$$

for some positive constant ℓ to be determined. (Choosing a smaller scaling would correspond to taking $\ell \searrow 0$, while choosing a larger scaling would correspond to letting $\ell \nearrow \infty$; either choice is sub-optimal, since the optimal ℓ will be strictly between 0 and ∞ as we shall see.)

2.1. **Main Result.** Under the above assumptions, we shall prove the following (where $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-s^2/2}\, ds$ is the cdf of a standard normal):

Theorem 1. *Consider the MCMCMC algorithm as described above, assuming (5), (6), and (7). Then as $d \to \infty$, the ESJD of (2) is maximized when ℓ is chosen to maximize $\ell^2 \times 2\,\Phi\left(-\ell\sqrt{I(\beta)/2}\right)$, where $I(\beta) > 0$ is defined in (10) below. Furthermore, for this optimal choice of ℓ, the corresponding probability of accepting a proposed swap is given (to three decimal points) by 0.234. In fact, the relationship between ESJD of (2) and ACC of (3) is given by*

$$ESJD \;=\; (2/dI(\beta)) \times ACC \times \big(\Phi^{-1}(ACC/2)\big)^2 \tag{8}$$

(see Figure 2). Finally, the optimal choice of ℓ also maximises ESJD(h) of (4), for any differentiable function $h : [0,1] \to \mathbb{R}$.

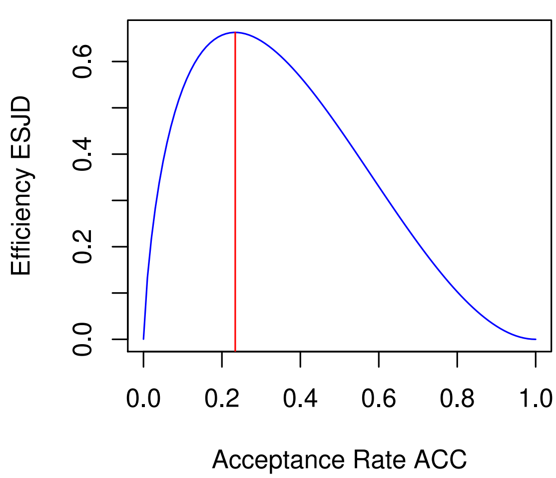

Figure 2: A graph of the relationship between the expected squared jumping distance (ESJD) and asymptotic acceptance rate (ACC),

as described in equation (8), in units where $dI(\beta) = 1$.

Proof. For this MCMCMC algorithm, the acceptance probability of a temperature swap is given by

$$\alpha \;\equiv\; 1 \wedge e^B$$

where

$$e^B \;=\; \frac{f_d^{(\beta)}(\mathbf{y}) f_d^{(\beta+\epsilon)}(\mathbf{x})}{f_d^{(\beta)}(\mathbf{x}) f_d^{(\beta+\epsilon)}(\mathbf{y})}$$

and where $\mathbf{x} \sim f_d^{(\beta)}$ and $\mathbf{y} \sim f_d^{(\beta+\epsilon)}$ are independent and in stationarity. We write $g = \log f$ so that $g_d^{(\beta)}(\mathbf{x}) = \log f_d^{(\beta)}(\mathbf{x})$, etc. We then have that

$$B \;=\; \Big[\big(g_d^{(\beta)}(\mathbf{y}) - g_d^{(\beta+\epsilon)}(\mathbf{y})\big) - \big(g_d^{(\beta)}(\mathbf{x}) - g_d^{(\beta+\epsilon)}(\mathbf{x})\big)\Big] \;\equiv\; T_d(\mathbf{y}) - T_d(\mathbf{x})\,,$$

where

$$T_d(\mathbf{x}) \;=\; g_d^{(\beta)}(\mathbf{x}) - g_d^{(\beta+\epsilon)}(\mathbf{x}) \;=\; \beta g_d(\mathbf{x}) - (\beta+\epsilon)\, g_d(\mathbf{x}) \;=\; -\epsilon\, g_d(\mathbf{x}) \;=\; -\sum_{i=1}^{d} \epsilon\, g(x_i)\,.$$

Now, write $\mathbb{E}^\beta$ for expectation with respect to the distribution having density proportional to f^β, and similarly for Var^β. Then in this distribution, g has mean

$$\mathbb{E}^\beta(g) \;=\; \frac{\int \log f(x) f^\beta(x) dx}{\int f^\beta(x) dx} \;\equiv\; M(\beta)\,, \tag{9}$$

and variance

$$\mathrm{Var}^\beta(g) \;=\; \frac{\int (\log f(x))^2 f^\beta(x) dx}{\int f^\beta(x) dx} - M(\beta)^2 \;\equiv\; I(\beta)\,. \tag{10}$$

Hence, in the distribution proportional to f_d^β, $T_d(\mathbf{x})$ has mean

$$\mathbb{E}_d^\beta\big(T_d(\mathbf{x})\big) \;=\; -d\epsilon\, M(\beta) \;\equiv\; \mu(\beta)\,,$$

and variance

$$\mathrm{Var}_d^\beta\big(T_d(\mathbf{x})\big) \;=\; d\epsilon^2 I(\beta) \;\equiv\; \sigma(\beta)^2\,.$$

Now, taking derivative with respect to β (using the "quotient rule"),

$$M'(\beta) \;=\; \frac{\int (\log f(x))^2 f^\beta(x) dx}{\int f^\beta(x) dx} - \left(\frac{\int f^\beta \log f(x) dx}{\int f^\beta(x) dx}\right)^2 \;=\; I(\beta)\,, \tag{11}$$

from which it follows that

$$\mu'(\beta) \;=\; -d\,\epsilon\, M'(\beta) \;=\; -d\,\epsilon\, I(\beta) \;=\; -\sigma(\beta)^2/\epsilon\,.$$

Now, recall that $\epsilon = d^{-1/2}\ell$. Hence,

$$T_d(\mathbf{y}) - T_d(\mathbf{x}) = -\frac{\ell}{\sqrt{d}} \sum_{i=1}^{d} \big(g(y_i) - g(x_i)\big)\,.$$

We claim that, as $d \to \infty$, $T_d(\mathbf{y}) - T_d(\mathbf{x})$ converges weakly to a random variable $A \sim N\left(-\sigma(\beta)^2,\, 2\,\sigma(\beta)^2\right)$. To see this, let $\phi_d(t)$ be the characteristic function of $T_d(\mathbf{y}) - T_d(\mathbf{x})$. Then we have:

$$\phi_d(t) = \mathbb{E}\left[e^{-it\epsilon(T_d(\mathbf{y}) - T_d(\mathbf{x}))}\right] = e^{-it\epsilon d(M(\beta+\epsilon) - M(\beta))} \left\{\mathbb{E}\left(e^{-it\epsilon \bar{g}(y_1)}\right) \mathbb{E}\left(e^{-it\epsilon \bar{g}(x_1)}\right)\right\}^d\,,$$

where $\bar{g}(x) = g(x) - \mathbb{E}^{\beta}(g(x))$. Since $M(\beta+\epsilon) - M(\beta) = \epsilon M(\beta) + o(\epsilon)$ and $\epsilon = d^{-1/2}\ell$, it follows that $\lim_{d\to\infty} it\epsilon d(M(\beta+\epsilon) - M(\beta)) = it\ell^2 I(\beta)$. Hence, using a Taylor series expansion,

$$\mathbb{E}\left(e^{-it\epsilon\bar{g}(y_1)}\right)\mathbb{E}\left(e^{-it\epsilon\bar{g}(x_1)}\right) = \left(1 - \frac{\ell^2 t^2}{2d} I(\beta+\epsilon) + o(d^{-1})\right)\left(1 - \frac{\ell^2 t^2}{2d} I(\beta) + o(d^{-1})\right)$$
$$= \left(1 - \frac{\ell^2 t^2}{2d} I(\beta) + o(d^{-1})\right)^2 .$$

Hence, $\phi_d(t)$ converges to $e^{-it\ell I(\beta)} e^{-\ell^2 t^2 I(\beta)}$ as $d \to \infty$. This limiting function is the characteristic function of the distribution $N(-\ell^2 I(\beta), 2\ell^2 I(\beta))$, thus proving the claim.

To continue, recall (e.g. [23], Proposition 2.4) that if $A \sim N(m, s^2)$, then

$$\mathbb{E}(1 \wedge e^A) = \Phi\left(\frac{m}{s}\right) + e^{m+\frac{s2}{2}}\,\Phi\left(-s - \frac{m}{s}\right) .$$

In particular, if $A \sim N(-c,\, 2c)$ (so that $\mathbb{E}(e^A) = 1$), then

$$\mathbb{E}(1 \wedge e^A) = 2\,\Phi\left(-\sqrt{c/2}\right) .$$

Since the function $1 \wedge e^B$ is a bounded function, it follows that as $d \to \infty$, the acceptance rate of MCMCMC converges to

$$\mathbb{E}(\alpha) = \mathbb{E}(1 \wedge e^B) \sim \mathbb{E}\left(1 \wedge \exp\left(N(-\sigma(\beta)^2, 2\sigma(\beta)^2)\right)\right) = 2\,\Phi(-\sigma(\beta)/\sqrt{2}) .$$

Now, with $\epsilon = d^{-1/2}\ell$, $\epsilon^2 d = \ell^2$ and $\sigma(\beta)^2/2 = d\epsilon^2 I(\beta)/2 = \ell^2 I(\beta)/2$, whence $\sigma(\beta)/\sqrt{2} = \ell\,[I(\beta)/2]^{1/2}$. Then as $d \to \infty$, $\mathbf{P}(\text{accept swap}) = \mathbb{E}(1 \wedge e^B) \to 2\,\Phi(-\ell\,[I(\beta)/2]^{1/2})$, and so

$$ESJD = \epsilon^2\,\mathbf{P}(\text{accept swap}) \sim (\ell^2/d) \times 2\,\Phi(-\ell\,[I(\beta)/2]^{1/2}) \equiv e_{MC}(\ell) . \quad (12)$$

Hence, maximising ESJD is equivalent to choosing $\ell = \ell_{opt}$ to maximise $\ell^2 \times 2\,\Phi(-\ell\,[I(\beta)/2]^{1/2})$, with the second factor being the acceptance probability. It then follows as in [23, 25] that when $\ell = \ell_{opt}$, the acceptance probability becomes 0.234 (to three decimal places). Indeed, making the substitution $u = \ell[I(\beta)/2]^{1/2}$ shows that finding ℓ_{opt} is equivalent to finding the value $\hat{u}$ of u which maximizes $8I(\beta)^{-2}u^2\Phi(-u)$, and then evaluating $\hat{a} = 2\Phi(-\hat{u})$. It follows that the value of $\hat{u}$, and hence also the value of $\hat{a}$, does not depend on the value of $I(\beta)$ (provided $I(\beta) > 0$). So, it suffices to assume $I(\beta) = 1$, in which case we compute numerically that $\hat{u} \doteq 1.190609$, so that $\hat{a} = 2\,\Phi(-1.190609) \doteq 0.2338071 \doteq 0.234$.

Finally, if instead of ESJD we consider $ESJD(h) \equiv \mathbb{E}[\left(h(\gamma) - h(\beta)\right)^2]$ for some differentiable function h, then as $\epsilon \searrow 0$ we have $\left(h(\gamma) - h(\beta)\right)^2 \sim [h'(\beta)]^2\,(\gamma - \beta)^2$, so $ESJD(h) \sim [h'(\beta)]^2\,(ESJD)$. Hence, as a function of ϵ, maximising $ESJD(h)$ is equivalent to maximising ESJD, and is thus maximised at the same value ϵ_{opt}. ∎

2.2. **Iteratively Selecting the Inverse Temperatures.** The above results show that, in high dimensions under certain assumptions, it is most efficient to choose the inverse temperatures for MCMCMC such that the average acceptance probability between any two adjacent temperatures is about 23%.

This leads to the question of how to select such temperatures. Although this may not be a practical approach in general, we shall adopt an intensive iterative approach to this in order to assess the effects of our theory in applications. We assume we know (perhaps by running some preliminary simulations) some sufficiently small inverse temperature value $\bar{\beta}$ such that the mixing of the chain with corresponding density $f_d^{\bar{\beta}}$ is sufficiently fast. This value $\bar{\beta}$ shall be our minimal inverse temperature. (In the examples below, we use $\bar{\beta} = 0.01$.)

In terms of $\bar{\beta}$, we construct the remaining β_j values iteratively, as follows. First, starting with $\beta_0 = 1$, we find β_1 such that the acceptance probability of a swap between $\beta_0 = 1$ and β_1 is approximately 0.23. Then, given β_1, we find β_2 such that the acceptance probability of the swap between β_1 and β_2 is again approximately 0.23. We continue in this way to construct $\beta_3, \beta_4, \ldots$, continuing until we have $\beta_j \leq \bar{\beta}$ for some j. At that point, we replace β_j by $\bar{\beta}$ and stop.

To implement this iterative algorithm, we need to find for each inverse-temperature β a corresponding inverse-temperature $\beta' < \beta$ such that the average acceptance probability of the swaps between β and β' is approximately 0.23. We use a simulation-based approach for this, via random variables $\{(X_n, X'_n, \rho_n)\}_{n\geq 0}$. After initialisations for $n = 0$, then at each time $n \geq 1$, we let $\beta'_n = \beta(1 + e^{\rho_n})^{-1}$, and draw $X_{n+1} \sim f_d^{\beta}$ and $X'_{n+1} \sim f_d^{\beta'_n}$ (or update them from some Markov chain dynamics preserving those distributions). The probability of accepting a swap between β and β'_n is then given by $\alpha_{n+1} \equiv 1 \wedge e^{B_{n+1}}$ where

$$B_{n+1} = -(\beta'_n - \beta)\left(g_d(X'_{n+1}) - g_d(X_{n+1})\right).$$

We then attempt to converge towards 0.23 by replacing ρ_n by

$$\rho_{n+1} \;=\; \rho_n + n^{-1}\left(\alpha_{n+1} - 0.23\right),$$

i.e. using a stochastic approximation algorithm [22, 2]. This ensures that if $\alpha_{n+1} > 0.23$ then β' will decrease, while if $\alpha_{n+1} > 0.23$ then β' will increase, as it should.

We shall use this iterative simulation approach to determine the inverse temperatures $\{\beta_j\}$ in all of our simulation examples in Section 3 below.

2.3. **Comparison with Geometric Temperature Spacing.** In some cases, it is believed that the optimal choice of temperatures is “geometric”, i.e. that we want $\beta_{j+1} = c\beta_j$ for appropriate constant c. Now, the notation of Theorem 1 corresponds to setting $\beta_j = \beta + \epsilon$ and $\beta_{j+1} = \beta$. So, it is optimal to have $\beta_{j+1} = c\beta_j$ precisely when $\epsilon_{opt} \propto \beta$ in Theorem 1.

Recall now that, from the end of the proof of Theorem 1, $\hat{u} \equiv \ell_{opt}[I(\beta)/2]^{1/2} \doteq 1.190609$, and in particular $\hat{u}$ does not depend on β. Then $\ell_{opt} = \hat{u}[2/I(\beta)]^{1/2}$. So, $\epsilon_{opt} = \ell_{opt} d^{-1/2} = \hat{u} d^{-1/2}[2/I(\beta)]^{1/2} \propto I(\beta)^{-1/2}$. It follows that the condition $\epsilon_{opt} \propto \beta$ is equivalent to the condition $I(\beta)^{-1/2} \propto \beta$, i.e. $I(\beta) \propto \beta^{-2}$.

While this does hold for some examples, e.g. the case when $f(x) = e^{-|x|^r}$ (see Section 2.4 below), it does *not* hold for most other examples (e.g. for the Gamma distribution, various $\log x$ terms appear which do not lead to such a simple relationship).

Thus, the optimal spacing of inverse temperatures is not necessarily geometric. In our simulations below, we consider both geometric spacings, and spacings found using our 23% rule. We shall see that the 23% rule leads to superior performance.

2.4. **A Simple Analytical Example.** Consider the specific example where the target density is given by (5) with $f(x) = e^{-|x|^r}$ for some fixed $r > 0$. (This includes the Gaussian ($r = 2$) and Double-Exponential ($r = 1$) cases.)

For this example, $f^\beta(x) \propto e^{-\beta|x|^r}$, and $g(x) = -|x|^r$. We then compute from (9) that

$$M(\beta) \;=\; \mathbb{E}^\beta(g) \;=\; \frac{\int g(x)\, f^\beta(x)\, dx}{\int f^\beta(x)\, dx} \;=\; \frac{-\int |x|^r\, e^{-|x|^r}\, dx}{\int e^{-|x|^r}\, dx}$$

$$= \frac{-2\,\Gamma(1/r)\,\beta^{-(1+r)/r}\,r^{-2}}{2\,\Gamma(1+\frac{1}{r})\,\beta^{-1/r}} \;=\; -1/r\beta$$

where $\Gamma(z) = \int_0^\infty t^{z-1}\, e^{-t}\, dt$ is the gamma function. Then by (11),

$$I(\beta) \equiv \mathrm{Var}^\beta(g) \;=\; M'(\beta) \;=\; 1/r\beta^2 \;\propto\; \beta^{-2}\,.$$

Hence, since $I(\beta) \propto \beta^{-2}$, it follows as above that the spread of inverse-temperatures should be geometric, with $\beta_{j+1} = c\beta_j$ for some constant c.

2.5. **Relation to Previous Literature.** The issue of temperature spacings and acceptance rates for MCMCMC has been widely studied in the physics literature [14, 15, 21, 7, 16, 5, 29, 30]. For example, the conclusion of Theorem 1, that the inverse temperatures should be chosen to make the swap acceptance probabilities all equal to the same value (here 0.234), is related to the "uniform exchange rates" criterion described in [13], p. 629. In addition, simulation studies [7] have suggested tuning the temperatures so the swap acceptance rate is about 20%, and analytic studies have suggested [16] an optimal rate of 0.23 in some specific circumstances, as we now discuss.

In physics, the *canonical ensemble* corresponds to writing $f_d^{(\beta)}(\mathbf{x}) = e^{-\beta V(\mathbf{x})}$, where V is the *energy* function and $1/\beta$ is the *temperature*. Thus, in our notation, $V(\mathbf{x}) = -g_d(\mathbf{x})$. Furthermore the average energy at inverse-temperature β is given by

$$U(\beta) \;=\; \frac{\int V(\mathbf{x})\, e^{-\beta V(\mathbf{x})}\, d\mathbf{x}}{\int e^{-\beta V(\mathbf{x})}\, d\mathbf{x}}\,.$$

Physicists have approached the problem using the concepts of *density of states* and *heat capacity*. If the state space $\mathcal{X}$ is discrete, the density of states is defined as the function $G(u) := \#\{x \in \mathcal{X} :\; V(x) = u\}$. If $\mathcal{X}$ is continuous, then $G(u)$ is defined to be the $(d-1)$-dimensional Lebesgue measure of the level set $\{x \in \mathcal{X} :\; V(x) = u\}$. The *heat capacity* of the system at inverse-temperature β is defined as

$$C(\beta) = -\beta^2 \frac{dU(\beta)}{d\beta} \;=\; -\beta^2 \mathrm{Var}^\beta\left(V(X)\right)\,.$$

In our notation, $\mathrm{Var}^\beta\left(V(X)\right) = \mathrm{Var}^\beta\left(-g_d(X)\right) = I(\beta)$, so $C(\beta) = -\beta^2 I(\beta)$. Hence, $C(\beta)$ is constant if and only if $I(\beta) \propto \beta^{-2}$ as discussed in Section 2.3.

The previous literature has considered, as have we, the idealised case in which exact sampling is done from each distribution f^β. In this case, a swap in MCMCMC between β and β' ($\beta < \beta'$) has acceptance probability

$$A(\beta, \beta') \;=\; \mathbb{E}\left[\min\left(1, e^{(\beta-\beta')(V(X)-V(X'))}\right)\right]\,,$$

with the expectation taken with respect to $X \sim f^{\beta}$ and $X' \sim f^{\beta'}$. Assuming that the heat capacity is constant (i.e., $C(\beta) \equiv C$), and that the density of states has the form

$$G(u) = \left(1 + \frac{\beta_0}{C}(u - u_0)\right)^C G(u_0),$$

[14] shows that the average acceptance probability of a swap between β and $\beta' > \beta$ is given by:

$$\begin{aligned} A(\beta, \beta') = \frac{2\Gamma(2C+2)}{\Gamma(C+1)^2} \int_0^{\beta/\beta'} \frac{u^C}{(1+u)^{2(C+1)}} du \\ = \frac{2}{\mathcal{B}(C+1, C+1)} \int_0^{1/(1+R)} \theta^C (1-\theta)^C d\theta, \quad (13) \end{aligned}$$

where $R = \beta'/\beta > 1$, $\Gamma(x)$ is the Gamma function and $\mathcal{B}(a, b)$, the Beta function.

Assuming a constant heat capacity and a more specific form for the density of state,

$$G(u) = \frac{(2\pi)^{C+1}}{\Gamma(C+1)} u^C,$$

[21] derived the same formula as in (13) for $A(\beta, \beta')$, which they call *the incomplete beta function law for parallel tempering*. Letting C (and thus the dimension d of the system) go to infinity, [15, 21] use (13) to derive a limiting expression for the acceptance probability of MCMCMC similar to that obtained in (12) above. [16] uses these limits to argue that 0.23 is approximately the optimal asymptotic swap acceptance rate using arguments somewhat similar to ours (indeed, Figure 1 of [16] contains the same acceptance-versus-efficiency curve implied by (12)).

Now, it seems that the heat capacity $C(\beta)$ being constant is quite a restrictive assumption; it is satisfied for the example $f(\mathbf{x}) = e^{-|x|^r}$ considered in Section 2.4, but is usually not satisfied for more complicated functions. By contrast, our assumption (6) does not impose any special conditions on the nature of the density function $f(\mathbf{x})$.

3. Simulation examples

3.1. **A Simple Gaussian Example.** We illustrate Theorem 1 with the following simulation example. We implement the MCMCMC described above for Gaussian target and tempered distributions. Specifically, we consider just two inverse-temperatures $\beta_0 \equiv 1$ and β_1 with $0 < \beta_1 < 1$. We define $f_d(\mathbf{x})$ and $f_d^{(\beta)}(\mathbf{x})$ as in (5) and (6), with f the density of a standard normal distribution $N(0, 1)$, so that $f^{(\beta)} = N(0, \beta^{-1})$. We use a version of the algorithm where the within-temperature moves are given by a Random Walk Metropolis (RWM) algorithm with Gaussian proposal distribution $N(\mathbf{x},\ (2.38^2/\beta d)\, I_d)$ which is asymptotically optimal [25]. Our algorithm attemps 20 within-temperature moves for every one time it attempts a temperature swap.

We ran this algorithm for each of 50 different possible choices of β_1, each with $0 < \beta_1 < 1$. For each such choice of β_1, we ran the algorithm for a total of 500,000 iterations to ensure good averaging, and then estimated the acceptance probability as the fraction of proposed swaps accepted, and the expected square jump distance

(ESJD) as the average of the squared temperature jump distances $(\gamma - \beta_0)^2$ (or, equivalently, as $(\beta_0 - \beta_1)^2$ times the average squared swap distance). Figure 3 plots the estimated acceptance probabilities versus squared jump distances (SJD), for each of four different dimensions (10, 20, 50, and 100). We can see from these results that the swap acceptance probability 0.234 is indeed close to optimal, and it gets closer to optimal as the dimension increases, and furthermore the relationship between the two quantities is given approximately by (8) and Figure 2, just as Theorem 1 predicts.

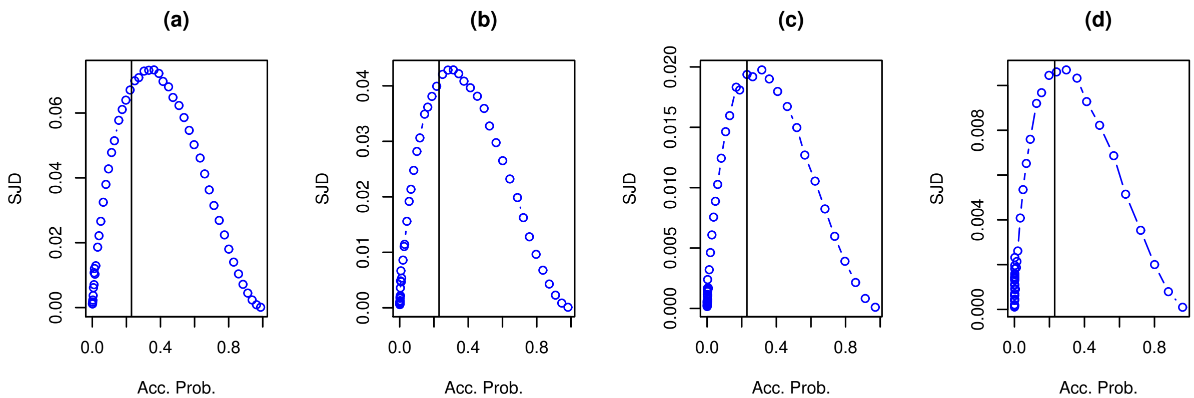

Figure 3: squared jump distance (SJD) versus estimated Acceptance Probability for the Gaussian case $f = N(0,1)$ and $f^\beta = N(0, \beta^{-1})$, in dimensions (a) $d = 10$, (b) $d = 20$, (c) $d = 50$, and (d) $d = 100$.

3.2. **Inhomogeneous Gaussian Example.** We now modify the previous example to a non-i.i.d. case where $f_d(\mathbf{x}) = \prod_{i=1}^d f_i(x_i)$ with $f_i = N(0, i^2)$, so that $f_i^\beta = N(0, i^2\beta^{-1})$. (The inhomogeneity implies, in particular, that assumption (5) is no longer satisfied.) The rest of the details remain exactly the same as for the previous example, except that the proposal distribution for the RWM within-temperature moves is now taken to be $N(x,\ (2.38^2 i^2/\beta d) I_d)$ for the i^{th} coordinate, which is optimal in this case. The resulting plots (for dimensions $d = 10$ and $d = 100$) are given in Figure 4. Here again, an optimum value of β for the ESJD function emerges at a swap acceptance probability of approximately 23%. Again, the agreement with Theorem 1 and the relationship (8) and Figure 2 is striking.

This is unsurprising given that this target distribution can be written as a collection of scalar linear transformations of Example 3.1, and Theorem 1 remains invariant through such transformations.

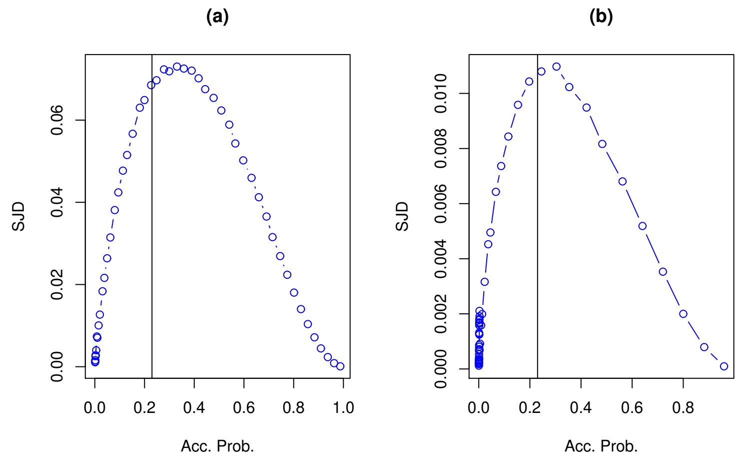

Figure 4: SJD versus Acceptance Probability for the inhomogeneous Gaussian case $f_i = N(0, i^2)$ and $f_i^\beta = N(0, i^2\beta^{-1})$, in dimensions (a) $d = 10$ and (b) $d = 100$.

3.3. **A Mixture Distribution Example.** In this example and the next, we compare the two temperature scheduling strategies discussed in Sections 2.2 and 2.3. One strategy consists of selecting the inverse-temperatures as a geometric sequence; we saw that this strategy is optimal when $I(\beta) \propto \beta^{-2}$, or equivalently the heat capacity is constant. The other strategy consists of choosing the temperatures such that the acceptance probability between any two adjacent temperatures is about 23%. We report here a simulation example comparing the two strategies.

Consider the density

$$f_d(x_1, \ldots, x_d) \;=\; \prod_{i=1}^{d} f(x_i; \omega_i, \mu_i, \sigma_i^2)$$

corresponding to a mixture of normal distributions, where $d = 20$, and

$$f(x; \omega, \mu, \sigma^2) = \sum_{j=1}^{3} \omega_j \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2\sigma_j^2}(x-\mu_j)^2},$$

with $\omega = (1/3, 1/3, 1/3)$, $\mu = (-5, 0, 5)$, and $\sigma = (0.2, 0.2, 0.2)$. For the 23% rule, we build the temperatures sequentially as described in Section 2.2, with $\bar{\beta} = 0.01$. The number of chains is thus itself unknown initially, and turns out to be 9 as shown in the top line of Table 1. The geometric schedule uses that same number (9) of chains, geometrically spaced between $\beta_0 \equiv 1$ and $\bar{\beta} \equiv 0.01$. (Thus, the geometric schedule is allowed to "borrow" the total number of chains from the 23% rule, which is in some sense overly generous to the geometric strategy.) Each chain was run for $200,000$ iterations.

We report the square jump distances both in the temperature space and in the $\mathcal{X}$-space. We observe that the 23% rule performs significantly better that the geometric scheduling, in terms of having larger average squared jumping distances in both the β space and the $\mathcal{X}$ space.

0.23 rule	1.00	0.675	0.395	0.206	0.106	0.048	0.022	0.0105	0.01
Geometric spacing	1.00	0.562	0.316	0.178	0.100	0.056	0.032	0.018	0.01

Table 1: Inverse-temperature schedules for the two different temperature scheduling strategies for the mixture distribution example.

	$\mathbb{E}\left[\lvert\beta_n - \beta_{n-1}\rvert^2\right]$	$\mathbb{E}\left[\lVert X_n - X_{n-1}\rVert^2\right]$
0.23 rule	0.209	0.428
Geometric spacing	0.084	0.315

Table 2: Squared Jumping distances in β space and $\mathcal{X}$ space for the two different temperature scheduling strategies for the mixture distribution example.

3.4. **Ising Distribution.** We now compare the two scheduling strategies using the Ising distribution on the $N \times N$ two-dimensional lattice, given by $\pi(x) = \exp(\mathcal{E}(x))/Z$, where Z is the normalizing constant and

$$\mathcal{E}(x) = J\left(\sum_{i=1}^{N}\sum_{j=1}^{N-1} x_{ij}x_{i,j+1} + \sum_{i=1}^{N-1}\sum_{j=1}^{N} x_{ij}x_{i+1,j}\right), \tag{14}$$

with $x_i \in \{1, -1\}$. Obviously, this distribution does not satisfy the assumption (5). For definiteness, we choose $N = 50$ and $J = 0.45$.

The Ising distribution admits a phase transition at the critical temperature $T_c = 2J/\log(1+\sqrt{2}) \doteq 1.021$, i.e. critical inverse-temperature $\beta_c \doteq 0.979$, around which the heat capacity undergoes stiff variation. Tempering techniques like MCMCMC and Simulated Tempering can perform poorly near critical temperatures because small changes in the temperature around T_c result in drastic variations of the properties of the distribution. We expect the 23% rule to outperform the geometric schedule in this case.

For our algorithm, for the within-temperature (i.e., $\mathcal{X}$-space) moves, we use a Metropolis-Hastings algorithm with a proposal which consists of randomly selecting a position (i, j) on the lattice and flipping its spin from x_{ij} to $-x_{ij}$.

We first determine iteratively the inverse-temperature points using the algorithm described in Section 2.2. Then, given the lowest and highest temperatures and the number of temperature points determined by this algorithm, we compute the geometric spacing. Figure 5 gives the selected temperatures (not inverse-temperatures, i.e. it shows $T_j \equiv 1/\beta_j$) from the two methods (the circles represent the 23% rule and the 'triangles' represent the geometric spacing).

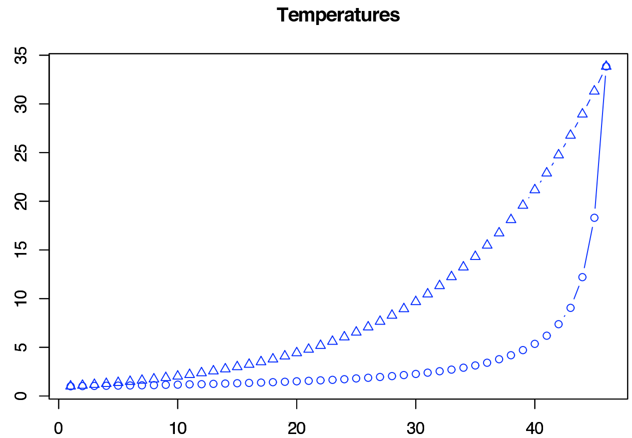

Figure 5: The temperatures $\{T_j\}$ selected for the Ising model (triangle: geometric spacing; circle: 23% rule).

We note here that the 23% rule and the geometric spacing produce very different temperature points. Interestingly, in order to maintain the right acceptance rate, the 23% rule puts more temperature points near the critical temperature (1.021); this is further illustrated in Table 3.

23% rule	1.00	1.02	1.06	1.10	1.14	1.18	1.24	1.30	1.37	1.44	1.53 $\cdots$
Geometric rule	1.00	1.20	1.46	1.77	2.14	2.59	3.13	3.79	4.59	5.55	6.70 $\cdots$

Table 3: Selected temperature values near $T_c = 1.021$ for the Ising model.

Each version of MCMCMC was run for 5 million iterations. The average squared jump distance in inverse-temperature space was 0.0095 for the 23% rule and 0.0049 for the geometric spacing, i.e. the 23% rule was almost twice as efficient.

We also present the trace plots (subsampled every 1,000 iterations) of the function $\{\mathcal{E}(X_n),\ n \geq 0\}$ during the simulation for each of the two algorithms in Figure 6, together with their autocorrelation functions. While the trace plots themselves appear similar, the autocorrelations indicate that the 23% rule has resulted in a faster mixing (less correlated) sampler in the $\mathcal{X}$-space as well. We confirm this by calculating the empirical average square jump distance in the $\mathcal{X}$ space, $SJD(\mathcal{X}) = n^{-1}\sum_{j=1}^{n} |\mathcal{E}(X_j) - \mathcal{E}(X_{j-1})|^2$, which works out to 22.11 for the 23% rule, and 13.76 for the geometric sequence, i.e. the 23% rule is 1.6 times as efficient by this measure.

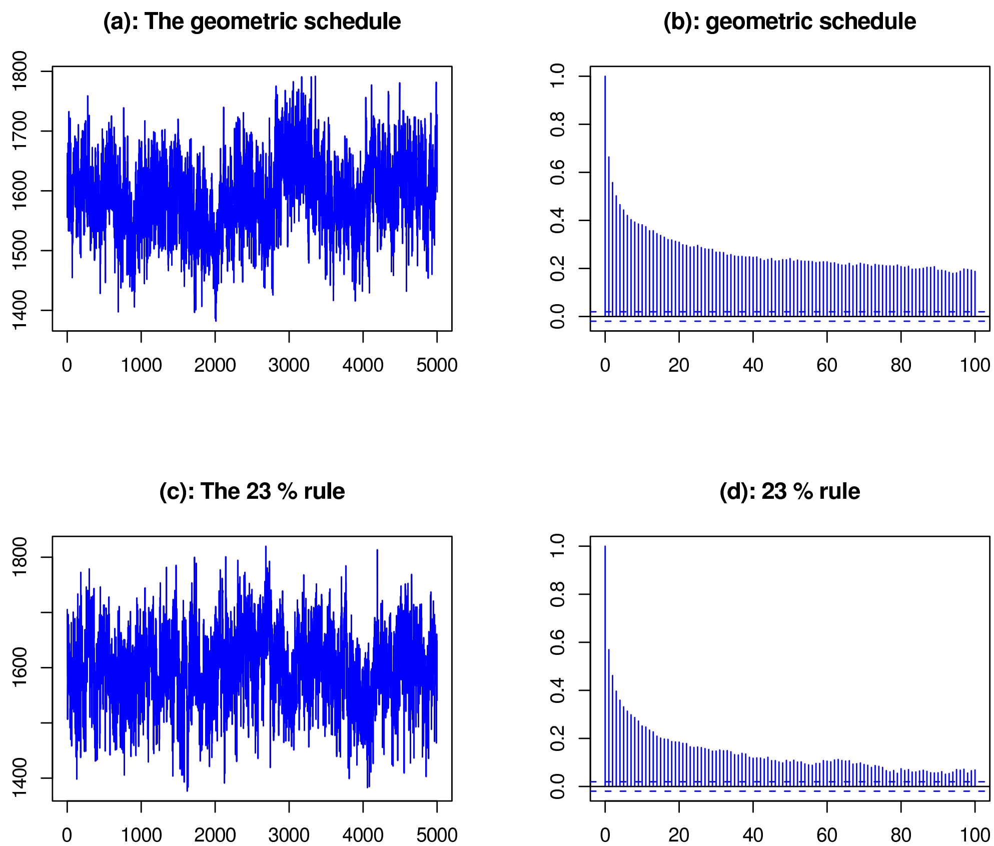

Figure 6: (a),(c) trace plots and (b),(d) autocorrelation functions of $\mathcal{E}(X_n)$ for the Ising model, subsampled every 1,000 iterations, with (a),(b) geometric temperature spacing, and (c),(d) the 23% rule.

4. Simulated Tempering

We now consider the Simulated Tempering algorithm [19, 9]. This is somewhat similar to MCMCMC, except now there is just *one* particle which can jump either within-temperature or between-temperatures. Again we focus on the between-temperatures move, and keep similar notation to before.

The state space is now given by $\{\beta_0, \beta_1, \ldots, \beta_n\} \times \mathcal{X}$ and the target density is proportional to $f_d(\beta, \mathbf{x}) = \prod_{i=1}^d e^{K(\beta)} f^\beta(x_i) = e^{dK(\beta)} f_d^{(\beta)}(\mathbf{x})$ for some choice of "normalizing" constants $K(\beta)$. Letting $\beta = \beta_j$ and $\beta + \epsilon = \beta_k$, a proposed temperature move from $(\beta, \mathbf{x})$ to $(\beta + \epsilon, \mathbf{x})$ is accepted with probability

$$\alpha \;=\; 1 \wedge \left(\frac{e^{dK(\beta+\epsilon)} f_d^{\beta+\epsilon}(\mathbf{x})}{e^{dK(\beta)} f_d^{\beta}(\mathbf{x})} \right) .$$

We again make the simplifying assumptions (5) and (6), and again let $\epsilon \searrow 0$ according to (7). We further assume that we have chosen the "true" normalising constants,

$$K(\beta) = -\log\left(\int f^\beta(x) dx \right), \tag{15}$$

so that $f_d(\beta, \cdot)$ is indeed a normalised density function. This choice makes the stationary distribution for β be uniform on $\{\beta_0, \beta_1, \ldots, \beta_n\}$; this appears to be optimal since it allows the β values to most easily and quickly migrate from the cold temperature β_0 to the hot temperature β_n and then back to β_0 again, thus maximising the influence of the hot temperature on the mixing of the cold chain.

4.1. **Main Result.** Under the above assumptions, we prove the following analog of Theorem 1:

Theorem 2. *Consider the Simulated Tempering algorithm as described above, assuming (5), (6), (7), and (15). Then as $d \to \infty$, the ESJD of (2) is maximized when ℓ is chosen to maximize $\ell^2 \times 2\,\Phi\left(-\ell I(\beta)^{1/2}/2\right)$. Furthermore, for this optimal choice of ℓ, the corresponding probability of accepting a proposed swap is given (to three decimal points) by 0.234. In fact, the relationship between ESJD of (2) and ACC of (3) is given by*

$$ESJD \;=\; (4/dI(\beta)) \times ACC \times \left(\Phi^{-1}(ACC/2)\right)^2 \tag{16}$$

(see Figure 7). Finally, this choice of ℓ also maximises the ESJD(h) of (4), for any differentiable function $h : [0,1] \to \mathbb{R}$.

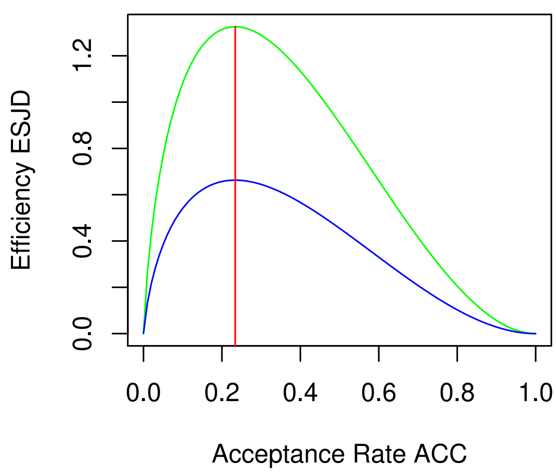

Figure 7: A graph of the relationship between the expected squared jumping distance (ESJD) and asymptotic acceptance rate (ACC), as described in equations (16) (top, green) and (8) (bottom, blue), in units where $dI(\beta) = 1$.

Proof. A proposed temperature move from $(\beta, \mathbf{x})$ to $(\beta + \epsilon, \mathbf{x})$ is accepted with probability

$$\alpha \;=\; 1 \wedge \left(\frac{e^{dK(\beta+\epsilon)} f_d^{\beta+\epsilon}(\mathbf{x})}{e^{dK(\beta)} f_d^{\beta}(\mathbf{x})}\right) \;=\; 1 \wedge e^{d(K(\beta+\epsilon)-K(\beta))} e^{\epsilon \sum_{i=1}^d g(x_i)}$$

$$= \; 1 \wedge e^{d(K(\beta+\epsilon)-K(\beta))+\epsilon d M(\beta)} e^{\epsilon \sum_{i=1}^d \bar{g}(x_i)}\,,$$

where $\bar{g}(x) = g(x) - M(\beta)$.

If we choose $K(\beta) = -\log\left(\int f^{\beta}(x)dx\right)$, then we compute (again using the "quotient rule") by comparison with (9) that $K'(\beta) = -M(\beta)$. Hence, from (11), $K''(\beta) = -M'(\beta) = -I(\beta)$. Therefore, by a Taylor series expansion, $K(\beta+\epsilon) - K(\beta) = \epsilon\, K'(\beta) + \frac{1}{2}\epsilon^2 K''(\beta) = -\epsilon M(\beta) - \frac{1}{2}\epsilon^2 I(\beta)$ for some $\beta_\epsilon \in [\beta, \beta+\epsilon]$. It follows that $d\left(K(\beta+\epsilon) - K(\beta)\right) + \epsilon d M(\beta) = \epsilon^2 d K''(\beta_\epsilon)/2 = (\ell^2/2)K''(\beta_\epsilon) = -(\ell^2/2)I(\beta_\epsilon)$.

Setting $\epsilon = d^{-1/2}\ell$ for some $\ell > 0$ and letting $d \to \infty$, it follows from the above and the central limit theorem that

$$\lim_{d\to\infty} 1 \wedge \left(\frac{e^{dK(\beta+\epsilon)} f_d^{\beta+\epsilon}(\mathbf{x})}{e^{dK(\beta)} f_d^{\beta}(\mathbf{x})}\right) \;=\; \lim_{d\to\infty} 1 \wedge \left(e^{-(\ell^2/2)I(\beta_\epsilon)}\; e^{\epsilon \sum_{i=1}^d \bar{g}(x_i)}\right)$$

$$=\; \lim_{d\to\infty} 1 \wedge \left(e^{-(\ell^2/2)I(\beta)}\; e^{\ell d^{-1/2} \sum_{i=1}^d \bar{g}(x_i)}\right) \;\equiv\; 1 \wedge e^{A}\,,$$

where $A \sim N(-(\ell^2/2)I(\beta),\ \ell^2 \mathrm{Var}^{\beta}(g)) = N(-(\ell^2/2)I(\beta),\ \ell^2 I(\beta))$.

The rest of the argument is standard, just as in Theorem 1, and shows that for large d, the squared jump distance of Simulated Tempering is approximately

$$ESJD \;=\; (\ell^2/2d) \times 2\,\Phi(-\ell I(\beta)^{1/2}/2) \;\equiv\; e_{ST}(\ell)\,, \tag{17}$$

which is again maximised by choosing ℓ such that the acceptance probability is (to three decimal places) equal to 0.234. The argument for (16) is identical to that for Theorem 1. ∎

4.2. **Comparison of Simulated Tempering and MCMCMC.**

Now that we have optimality results for both Simulated Tempering (Theorem 2) and MCMCMC (Theorem 1), it is natural to compare them. We have the following.

Corollary 1. *Under assumptions (5), (6), (7), and (15), asymptotically as* $d \to \infty$, *the maximal value of ESJD for Simulated Tempering is precisely* twice *that for MCMCMC (cf. Figure 7). (More generally, the maximal ESJD(h) for Simulated Tempering is precisely* twice *that for MCMCMC, for any differentiable h). Furthermore, the optimal choice of* ℓ *for Simulated Tempering is* $\sqrt{2}$ *times that for MCMCMC.*

Proof. Comparing $e_{MC}(\ell)$ from (12) with $e_{ST}(\ell)$ from (17), we see that $e_{MC}(\ell) = \frac{1}{2} e_{ST}(\ell\sqrt{2})$. In particular, $\sup_\ell e_{MC}(\ell) = \frac{1}{2}\sup_\ell e_{ST}(\ell)$, and also $\mathrm{argsup}_\ell e_{MC}(\ell) = (1/\sqrt{2})\, \mathrm{argsup}_\ell e_{ST}(\ell)$, which gives the result. ∎

Corollary 1 says that, under appropriate conditions, the ESJD of MCMCMC is precisely *half* that of Simulated Tempering. That is, if Simulated Tempering has ideally-chosen normalisation constants $K(\beta)$ as in (15), then in some sense it is twice as efficient as MCMCMC. (For a related inequality about spectral gaps of these two algorithms on finite state spaces, see Theorem 3 of [31].)

However, choosing $K(\beta)$ ideally may be difficult or impossible (though it may be possible in certain cases to learn these weights adaptively during the simulation, see e.g. [4]), and for non-optimal $K(\beta)$ this comparison no longer applies. By contrast, MCMCMC does not require $K(\beta)$ at all.

Furthermore, a single temperature swap of MCMCMC updates *two* different chain values, and thus may be twice as valuable as a single temperature move under

Simulated Tempering. If so, then the two different factors of two precisely cancel each other out.

In addition, it may take more computation to do one within-temperature step of MCMCMC (involving $n+1$ particles) than one step of Simulated Tempering (involving just one particle).

In summary, this comparison of the two algorithms involves various subtleties and is not entirely clear-cut, though Corollary 1 still provides certain insights into the relationship between them.

4.3. **Langevin Dynamics for Simulated Tempering?** One limitation of Simulated Tempering is that it performs a Random Walk Metropolis (RWM) algorithm in the β-space, where each proposal bears a high chance of being rejected if the value of the energy $\sum_{i=1}^{d} g(x_i)$ is not consistent with the proposed value of temperature. Now, it is known [24] that Langevin dynamics (which use derivatives to improve the proposal distribution) are significantly more efficient than RWM when they can be applied. In the context of the inverse temperatures, a Langevin-style proposal distribution could take the form:

$$\mathcal{N}\left[\beta + \frac{\sigma^2}{2}\left(\sum_{i=1}^{d} g(x_i) - d\nabla K(\beta)\right), \sigma^2\right].$$

Furthermore, in this case $\nabla K(\beta) = \mathbb{E}_\beta(g(X))$. Therefore, such a Langevin proposal will compare the current value of $\sum_{i=1}^{d} g(x_i)$ to the average value $d\nabla K(\beta)$. If $\sum_{i=1}^{d} g(x_i) \leq d\nabla K(\beta)$ then smaller temperature are more compatible and are more likely to be proposed (and vice versa).

The main limitation of this idea is that in practice, we do not know the gradient $\nabla K(\beta)$. This can perhaps be estimated during the simulation as the average of the energy $\sum_{j=1}^{d} g(X_n)$ at times n where the inverse-temperature level β is visited, but this needs more investigation and we do not pursue it here.

5. Discussion and Future Work

This paper has presented certain results about optimal inverse-temperature spacings. In particular, we have proved that for MCMCMC (Theorem 1) and Simulated Tempering (Theorem 2), it is optimal (under certain conditions) to space the inverse-temperatures so that the probability of accepting a temperature swap or move is approximately 23%. Our theorems were proved under the restrictive assumption (5), but we have seen in simulations (Section 3) that they continue to approximately apply even if this assumption is violated.

Theorems 1 and 2 were stated in terms of the expected squared jumping distances (ESJD) of the inverse-temperatures, assuming the chain begins in stationarity. While this provides useful information about optimising the algorithm, it provides less information about the algorithm's long-term behaviour. In this section, we consider various related issues and possible future research directions.

5.1. **Inverse-Temperature Process.** It is possible to define an entire *inverse temperature process*, $\{y_n\}_{n=0}^{\infty}$, living on the state space $\mathcal{Y} = \{\beta_0, \beta_1, \ldots, \beta_n\}$. For Simulated Tempering (ST), this process is simply the inverse-temperature β at each iteration. For MCMCMC, this process involves "tracking" the influence of a particular inverse-temperature's chain through the temperature swaps, i.e. if $y_n = \beta_j$, and then the β_j and β_{j+1} chains successfully swap, then $y_{n+1} = \beta_{j+1}$ (otherwise $y_{n+1} = \beta_j$).

In general, this $\{y_n\}$ process will not be Markovian. However, if we assume that the ST/MCMCMC algorithm does an automatic i.i.d. "reset" into the distribution $f^{(\beta)}$ immediately upon entering the inverse-temperature β (i.e., where the chain values associated with each inverse-temperature β are taken as i.i.d. draws from $f_d^{(\beta)}$ at each iteration), then $\{y_n\}$ becomes a Markov chain. Indeed, under this assumption, it corresponds to a simple random walk (with holding probabilities) on $\mathcal{Y}$.

In that case, it seems likely that as $d \to \infty$, if we appropriately shrink space by a factor of $\sqrt{d}$ and speed up time by a factor of d, then as in the RWM case [23, 25], the $\{y_n\}$ will converge to a limiting Langevin diffusion process. In that case, maximising the speed of the limiting diffusion is equivalent to optimising the original algorithm (according to any optimisation measure, see [25]), and this would provide another justification of the values of ℓ_{opt} in Theorems 1 and 2.

In terms of this $\{y_n\}$ Markov chain, ESJD from (2) is simply the expected squared jumping distance of this process. Furthermore, ESJD(h) from (4) is then the *Dirichlet form* corresponding to the Markov operator for $\{y_n\}$. Thus, Theorems 1 and 2 would also be saying that the given values of ℓ_{opt} maximise the associated Dirichlet form.

Of course, this "reset" assumption that the moves within each inverse temperature β_j result in an immediate i.i.d. reset to the density f^{β_j} is not realistic, since in true applications the convergence within each fixed temperature chain would occur only gradually (especially for larger β_j). However, it is not entirely unrealistic either, for a number of reasons. For example, some algorithms might run many within-temperature moves for each single attempted temperature swap, thus making the within-temperature chains effectively mix much faster. Also, some within-temperature algorithms (e.g. Langevin diffusions, see [24]) have convergence time of smaller order ($O(d^{1/3})$) than that of the temperature-swapping random-walk behaviour ($O(d)$), so effectively the within-temperature chains converge immediately on a relative scale, which is equivalent to the "reset" assumption.

Proving convergence to limiting diffusions can be rather technical (see e.g. [23, 24]), so we do not pursue it here, but rather leave it for future work. In any case, assuming the limiting diffusion does hold (as it probably does), this provides new insight and context for the optimal behaviour of MCMCMC and ST algorithms.

5.2. **Joint Diffusion Limits?** If we do not assume the "reset" assumption as above, then the process $\{y_n\}$ is not Markovian, so a diffusion limit seems far less certain.

However, it is still possible to jointly consider the chain $\{\mathbf{x}_n\}_{n=0}^{\infty}$ living on $\mathcal{X}$ (for ST) or $\mathcal{X}^d$ (for MCMCMC), *together* with the inverse temperature process $\{y_n\}$. The *joint* process $(\mathbf{x}_n, y_n)$ must be Markovian, and it is quite possible that it has its own joint diffusion limit and joint optimal scalings.

This would become quite technical, and we do not pursue it here, but we consider it an interesting topic for future research.

5.3. **Models For How the Algorithms Converge.** Related to the above is the question of a deeper understanding of how MCMCMC and ST make use of the tempered distributions to improve convergence. Significant progress was made in this direction in previous research (e.g. [29]), but more remains to be done.

One possible simplified model is to assume the process does not move at all in the within-temperature direction, except at the single inverse-temperature $\beta = \bar{\beta}$ when it does an immediate i.i.d. reset to $f_d^{(\bar{\beta})}$. At first we thought that such a model is insufficient since it would then be exponentially difficult for the algorithm to climb from $\beta = \bar{\beta}$ to $\beta = \beta_0$, but the work of [29] shows that this is in fact not the case. So, we may explore this model further in future work.

A compromise model is where the state space $\mathcal{X}$ is partitioned into a finite number of modes, and when entering any $\beta > 0$ the process does a "reset" into $f^{(\beta)}$ conditional on remaining in the current mode. Such a process assumes fast within-mode mixing at all temperatures, but between-mode mixing only at the hot temperature when $\beta = 0$. We believe that in this case, there is a joint diffusion limit of the joint (β,mode) process. If so, this would be interesting and provide a good model for how the hot temperature and intermediate temperature mixing all works together.

5.4. **State space invariance.** Our results do not depend in any way on the update mechanism within the state space $\mathcal{X}$. Moreover, although our results are proved for the case of IID components, this observation immediately generalises our findings to the very large class of distributions which we can write as functions of IID components. In fact the only thing which stops this being a completely general $d-$dimensional state space result is the fact that ST on the transformed variables is a different method from the transformed ST algorithm. From a practical point of view, this might even suggest improved tempering schemes (rather than simply considering powered densities).

5.5. **Implementation.** Finding practical ways to implement MCMCMC and ST in light of our results has not been a focus of this paper, though this is an important issue for future consideration. The approach adopted in Subsection 2.2 is designed to be a thorough methodology to investigate the effects of Theorem 1, but cannot be considered a practical approach in general. Intriguing is the propect of integrating our theory within an *adaptive MCMC* framework as in for example [3, 1, 27, 26].

5.6. **Parallelisation.** As computational power become more and more widespread, there is increasing interest in running Monte Carlo algorithms in *parallel* on many machines, perhaps even using Graphics Processing Units (GPUs), see e.g. [10, 28, 17]. It would be interesting to consider, even in simple situations, how to optimise parallel computations asymptotically as the number of processors goes to infinity.

Acknowledgements: We thank the editors and anonymous referees for very helpful reports that significantly improved the presentation of our results.

References

[1] C. Andrieu and E. Moulines (2007) On the ergodicity properties of some Markov chain Monte Carlo algorithms. Ann. Appl. Prob. **44**, 2, 458–475.

[2] C. Andrieu and C.P. Robert (2001), Controlled MCMC for optimal sampling. Preprint.

[3] Y.F. Atchade, An adaptive version of the Metropolis adjusted Langevin algorithm with a truncated drift, Meth. and Comp. in Appl. Prob., 2006, **8**, 2, 235–254.

[4] Y.F. Atchade and J.S. Liu (2004), The Wang-Landau algorithm in general state spaces: applications and convergence analysis. Statistica Sinica, to appear.

[5] B. Cooke and S.C. Schmidler (2008), Preserving the Boltzmann ensemble in replica-exchange molecular dynamics. J. Chem. Phys. **129**, 164112.

[6] J.-F. Delmas and B. Jourdain (2009), Does waste recycling really improve the multi-proposal Metropolis-Hastings algorithm? An analysis based on control variates. J. Appl. Prob. **46(4)**, 938–959.

[7] D.J. Earl and M.W. Deem (2005), Parallel tempering: theory, applications, and new perspectives. J. Phys. Chem. B **108**, 6844.

[8] D. Frenkel (2006), Waste-recycling monte carlo. In *Computer simulations in condensed matter: from materials to chemical biology*. Lecture Notes in Physics **703**, Springer, Berlin.

[9] C.J. Geyer (1991), Markov chain Monte Carlo maximum likelihood. In Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, 156–163.

[10] P.W. Glynn and P. Heidelberger (1991), Analysis of Parallel, Replicated Simulations under a Completion Time Constraint. ACM Trans. on Modeling and Simulation **1**, 3–23.

[11] P.J. Green and A. Mira (2001), Delayed rejection in reversible jump Metropolis-Hastings. Biometrika **88(3)**, 1035–1053.

[12] W.K. Hastings (1970), Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109.

[13] Y. Iba (2001), Extended ensemble Monte Carlo. Int. J. Mod. Phys. C **12(5)**, 623–656.

[14] D.A. Kofke (2002), On the acceptance probability of replica-exchange Monte Carlo trials. J. Chem. Phys. **117**, 6911. Erratum: J. Chem. Phys. **120**, 10852.

[15] D.A. Kofke (2004), Comment on reference [21]. J. Chem. Phys. **121**, 1167.

[16] A. Kone and D.A. Kofke (2005), Selection of temperature intervals for parallel-tempering simulations. J. Chem. Phys. **122**, 206101.

[17] A. Lee, C. Yau, M.B. Giles, A. Doucet, and C.C. Holmes (2009), On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo Methods. Preprint.

[18] N. Madras and Z. Zheng (2003), On the swapping algorithm. Random Struct. Alg. **22**, 66–97.

[19] E. Marinari and G. Parisi (1992), Simulated tempering: a new Monte Carlo scheme. Europhys. Lett. **19**, 451–458.

[20] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953), Equations of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1091.

[21] C. Predescu, M. Predescu, and C.V. Ciobanu (2004), The incomplete beta function law for parallel tempering sampling of classical canonical systems. J. Chem. Phys. **120**, 4119.

[22] H. Robbins and S. Monro (1951), A stochastic approximation method. Ann. Math. Stat. **22**, 400–407.

[23] G.O. Roberts, A. Gelman, and W.R. Gilks (1997), Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Appl. Prob. **7**, 110–120.

[24] G.O. Roberts and J.S. Rosenthal (1998), Optimal scaling of discrete approximations to Langevin diffusions. J. Roy. Stat. Soc. B **60**, 255–268.

[25] G.O. Roberts and J.S. Rosenthal (2001), Optimal scaling for various Metropolis-Hastings algorithms. Stat. Sci. **16**, 351–367.

[26] G.O. Roberts and J.S. Rosenthal (2009), Examples of Adaptive MCMC. J. Comp. Graph. Stat. **18(2)**, 349–367.

[27] G.O. Roberts and J.S. Rosenthal (2007), Coupling and ergodicity of Adaptive MCMC. J. Appl. Prob. **44**, 458–475.

[28] J.S. Rosenthal (2000), Parallel computing and Monte Carlo algorithms. Far East J. Theor. Stat. **4**, 207–236.

[29] D.B. Woodard, S.C. Schmidler, and M. Huber (2009), Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. Ann. Appl. Prob. **19**, 617–640.

[30] D.B. Woodard, S.C. Schmidler, and M. Huber (2009), Sufficient conditions for torpid mixing of parallel and simulated tempering. Elec. J. Prob. **14**, 780–804.

[31] Z. Zheng (2003), On swapping and simulated tempering algorithms. Stoch. Proc. Appl. **104**, 131–154.