BAYESIAN STATISTICS 5, pp. 599–607
J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (Eds.)
© Oxford University Press, 1996

Efficient Metropolis Jumping Rules

A. GELMAN*, G. O. ROBERTS** and W. R. GILKS***
**University of California, USA, **University of Cambridge, UK* and
****Medical Research Council, UK*

SUMMARY

The algorithm of Metropolis *et al.* (1953) and its generalizations have been increasingly popular in computational physics and, more recently, statistics, for sampling from intractable multivariate distributions. Much recent research has been devoted to increasing the efficiency of simulation algorithms by altering the jumping rules for Metropolis-like algorithms. We study a very specific question: What are the most efficient symmetric jumping kernels for simulating a normal target distribution using the Metropolis algorithm? We provide a general theoretical result as the dimension of a class of canonical problems goes to ∞ and numerical approximations and simulations for low-dimensional Gaussian target distributions that show that the limiting results provide extremely accurate approximations in six and higher dimensions. For a d-dimensional spherical multivariate normal problem, the optimal symmetric jumping kernel has the following properties: (1) its scale is approximately $2.4/\sqrt{d}$ times the scale of the target distribution; (2) the acceptance rate of the associated Metropolis algorithm is approximately 44% for $d = 1$ and declines to 23% as $d \to \infty$; and (3) the efficiency of the Metropolis algorithm, compared to independent samples from the target distribution, is approximately $0.3/d$.

Keywords: ACCEPTANCE RATE; BAYESIAN COMPUTATION; ITERATIVE SIMULATION; MARKOV CHAIN MONTE CARLO; METROPOLIS ALGORITHM; NORMAL DISTRIBUTION.

1. INTRODUCTION

Iterative simulation methods have recently become popular tools in statistical analysis, especially in the calculation of posterior distributions arising in Bayesian inference. For reviews of the theory and its applications, see Besag and Green (1993), Gilks *et al.* (1993), and Smith and Roberts (1993). The goal of Markov chain Monte Carlo is to estimate a (typically multivariate) *target distribution*, $\pi(\theta)$, by generating a Markov chain $\theta^{(1)}, \theta^{(2)}, \ldots$ whose stationary distribution is π. A particularly important algorithm, on which we shall focus, is that of Metropolis *et al.* (1953). This is characterized by a symmetric *jumping density*, $J(\theta, \theta')(= J(\theta', \theta))$, the density of proposing a candidate θ' when the current iteration $\theta^{(t)}$ takes the value θ. The candidate is accepted or rejected according to an acceptance probability,

$$\alpha(\theta, \theta') = \min\left(\frac{\pi(\theta')}{\pi(\theta)}, 1\right).$$

If the candidate is accepted the next iteration $\theta^{(t+1)}$ takes the value of the candidate θ'; if rejected $\theta^{(t+1)}$ takes the old value θ: that is, the chain stands still.

Practical implementations of the Metropolis algorithm often suffer from slow mixing and therefore inefficient estimation, for at least two reasons: (1) the jumps are so short that the simulation moves very slowly through the target distribution; or (2) the jumps are nearly all into low-probability areas of the target density, causing the Markov chain to stand still most of the time. When simulations are slow, it is often possible to improve mixing by properly adjusting the jumping distribution. Various heuristic rules have been suggested for fixing these problems during a simulation, either by monitoring the distance of each jump or the frequency of acceptances in the simulation.

In this paper, we explore the efficiency of Metropolis's algorithm and the way in which it depends on the chosen jumping kernel. We mainly confine our attention to the commonly used spherically symmetric random walk Metropolis algorithm, where $J(\theta, \theta')$ takes the special form, $J(|\theta' - \theta|)$.

Our most important theoretical result appears in Section 3. This considers a sequence of canonical algorithms, for which the asymptotic limit provides considerable insight into the behavior of approximately optimal Metropolis algorithms in high dimensional problems. In practice, the result gives rise to the following heuristic strategy, which is extremely easy to implement: *Choose the scaling of $J(\cdot, \cdot)$ so that the average acceptance rate of the algorithm is roughly 1/4.*

In Section 2 we consider one-dimensional examples where various measures of efficiency can be calculated. These examples provide some motivation for the asymptotic results of Section 3. In Section 3.3 we present a simulation study to show that even for relatively small dimensional problems (6–8 were enough in our study), the asymptotic result is accurate, and the above heuristic produces efficient results.

2. UNIVARIATE EXAMPLES

2.1. *Notation and Measures of Efficiency*

Assume for the time being that θ is one-dimensional. We shall consider two measures of efficiency of the Markov chain. The first measure is based on the asymptotic variance of the sample mean, $\overline{\theta} = \frac{1}{N}\sum_{t=1}^{N} \theta^{(t)}$. Under independent sampling of θ, $\mathrm{var}(\overline{\theta}) = \tau^2/N$, where τ^2 is the variance of the target density. The asymptotic efficiency of the Markov chain sampling for $\overline{\theta}$ is thus,

$$\mathrm{eff}_{\overline{\theta}} = \frac{\tau^2}{V_{\overline{\theta}}} = [1 + 2(\rho_1 + \rho_2 + \rho_3 + \ldots)]^{-1}, \tag{1}$$

where $V_{\overline{\theta}} = \lim_{N\to\infty} N\mathrm{var}(\overline{\theta})$ is the limiting scaled sample variance from the Markov chain output, and ρ_t is the autocorrelation of the Markov chain at lag t. In other words, $\mathrm{eff}_{\overline{\theta}}$ is the reciprocal of the integrated autocorrelation time for measuring the mean of θ.

An alternative measure of efficiency is given by a bound of $\mathrm{eff}_{\overline{\theta}}$ derived from the eigenvalue decomposition of the algorithm. If $1 = \lambda_1, \lambda_2, \ldots$ denote the eigenvalues of the transition kernel of the algorithm (at least so that λ_2 is the second largest), a well known inequality for $\mathrm{eff}_{\overline{\theta}}$ (see, for example, Besag and Green, 1993) is the following:

$$\mathrm{eff}_{\overline{\theta}} = \left\{ \sum_{s=1}^{\infty} a(i) \frac{1+\lambda_i}{1-\lambda_i} \right\}^{-1} \geq \frac{1-\lambda_2}{1+\lambda_2}, \tag{2}$$

which we define as the efficiency based on the second eigenvalue:

$$\mathrm{eff}_{eig} \equiv \frac{1-\lambda_2}{1+\lambda_2}.$$

Here $a(\cdot)$ is a probability measure on the positive integers. Unfortunately, it is rarely possible to easily estimate eff_{eig}. In the next subsection, we shall begin to consider how measures of efficiency can be related to more easily monitored quantities.

One of the easiest characteristics of a Metropolis algorithm to monitor is the frequency of "acceptance" in the Metropolis step—which we label p_{jump}. It has been claimed that, for a wide variety of problems, optimal rules have acceptance probabilities near 0.5 (see, for example, Muller, 1993).

2.2. *Numerical Results*

For our one-dimensional examples, we compute the measures of efficiency of Metropolis's algorithm for a unit normal target distribution under a variety of symmetric jumping kernels. For this we use a discrete approximation to π, replacing the sample space of θ by a discrete array of 100 points spread evenly between -6 and 6; we compute the normal density at these points and then renormalize. We compute eff$_{eig}$ directly from the transition matrix and eff$_{\bar{\theta}}$ using an asymptotic formula (Peskun, 1973; Kemeny and Snell, 1969). After performing all our computations, we check the effects of the discrete approximation by repeating some computations on the grid of 200 points spread evenly between -8 and 8, obtaining the same results to two decimal places.

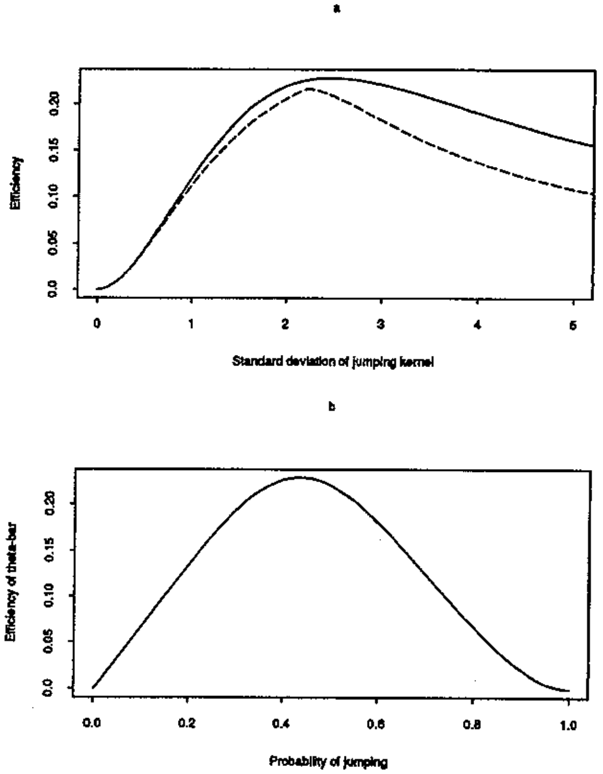

Figure 1. *Univariate normal target distribution, normal jumping kernel.*

2.2.1. *Normal jumping kernels*

We start by considering symmetric jumping rules based on a normal kernel with standard deviation σ: that is, the jumping, or candidate, distribution is

$$J(\theta, \theta') \propto \exp\left(-\frac{1}{2}\frac{(\theta-\theta')^2}{\sigma^2}\right).$$

If σ is too low, the Metropolis steps are too short and move too slowly through the target distribution; if σ is too high, the algorithm almost always rejects and stays in the same place. The optimal σ is somewhere in between. In a general context, Tierney (1991) suggests setting the scale of the jumping kernel at 0.5 or 1 times the scale of the target distribution, and Muller (1993) suggests $\sigma = 1$ in general. We compute the two efficiency measures, eff_{eig} and $\text{eff}_{\overline{\theta}}$, for normal jumping kernels with different choices of σ, in all cases simulating the unit normal target distribution.

Figure 1a displays eff_{eig} and $\text{eff}_{\overline{\theta}}$ as a function of σ, the standard deviation of the jumping rule. The optimal value of σ under either efficiency criterion is about 2.4, a surprisingly high value. However, this is consistent with the recommendations of Besag and Green (1993) in the rejoinder to the discussion of their paper.

The optimal efficiency, using either measure, is just below 0.25. (The "corner" at the maximum of the eff_{eig} line occurs when the second and third largest eigenvalues are equal.) Interestingly, if one cannot be optimal, it seems better to use too high a value of σ than too low; $\sigma = 5$ is better than $\sigma = 1$.

For the normal kernel and normal target distribution, the acceptance rate, or average jumping probability, can be determined analytically:

$$p_{jump} = \frac{2}{\pi}\arctan\left(\frac{2}{\sigma}\right).$$

Figure 1b plots the efficiency measure $\text{eff}_{\overline{\theta}}$ as a function of acceptance rate; the leftmost point on the curve corresponds to $\sigma \to \infty$, and the rightmost point to $\sigma = 0$. At least for this example, the folklore seems correct; an acceptance rate near (but slightly below) 0.5 is optimal.

2.2.2. *Other kernels*

To broaden our understanding, we repeated the above procedure with other symmetric jumping rules. The simplest alternative is the symmetric uniform kernel, which we parameterize as having width $\sqrt{12}\sigma$ and thus a variance of σ^2. We determine the efficiency measures by matrix computations on a discrete grid, as before, and estimate the probability of jumping for each σ by simulation. The results (not shown) are nearly identical to Figures 1a–b.

We also tried a bimodal kernel—a mixture of two normal densities, each with standard deviation $\sigma/\sqrt{2}$, set a distance $\sigma\sqrt{2}$ apart. As with the previous jumping rules, this kernel has a variance of σ^2. The idea was to reduce the number of very small jumps that detract from the efficiency of the Metropolis simulations. Once again, the results were remarkably close to the efficiencies of the other kernels, whether parameterized by kernel variance or acceptance rate.

3. MULTIVARIATE TARGET DISTRIBUTIONS

3.1. *Limiting Diffusion Process Approximation*

We now consider multivariate θ. We state an asymptotic result motivated by the investigations above, a special case of a slightly more general result proved in Roberts, Gelman, and Gilks (1994). We drop the assumption of normality of π, and merely assume that π is d-dimensional, and has the product form,

$$\pi(\theta) = \prod_{i=1}^{d} f(\theta_i), \tag{3}$$

for some one-dimensional density f. We shall assume various regularity conditions on the form of π which are not mentioned explicitly here but are detailed in Roberts, Gelman, and Gilks (1994). Suppose we use a multivariate normal Metropolis proposal kernel centered about the current point, but with variance-covariance matrix $\sigma_d^2 I_d \equiv (\phi^2/d)I_d$. Let $Y_t^d = \theta_1^{[td]}$ be a speeded up, continuous time version of the d-dimensional Metropolis algorithm. Here $[td]$ denotes the integer part of td. In other words, Y_d is a process in continuous time which remains constant for a time interval $1/d$, before making a jump according to the Metropolis algorithm with proposal variance $\sigma_d^2 I_d$. We are only considering the first component of θ, which is in general not a Markov process in its own right. Remarkably, however, the limit of Y^d as $d \to \infty$ is Markov. More precisely, we have the following

Theorem 3.1. *As $d \to \infty$, assuming the starting values for the components of θ, $\theta_1, \theta_2, \ldots$ are all distributed independently according to f, then the process Y^d converges weakly to the limiting Langevin diffusion, which satisfies the stochastic differential equation,*

$$dY_t = \frac{f'(Y_t)h(\phi)}{2f(Y_t)}dt + h(\phi)^{1/2}dB_t, \tag{4}$$

where

$$h(\phi) = 2\phi^2\Phi\left(\frac{-\phi F^{1/2}}{2}\right), \tag{5}$$

and

$$F = \int_{-\infty}^{\infty} \frac{(f'(x))^2}{f(x)}dx \tag{6}$$

is a Fisher's information measure for f ($F = 1$ for standard normal f). The limiting value of p_{jump} for this sequence of problems is $h(\phi)/\phi^2$.

The speed of the diffusion $h(\phi)$ is maximized by the choice $\phi = \tilde{\phi} = 2.38/F^{1/2}$. Therefore the asymptotically optimal jumping kernel has variance-covariance matrix $(\tilde{\phi}^2/d)I_d$, with jumping probability approximately 0.234.

For the diffusion process limit, all measures of efficiency are equivalent (up to a multiplicative constant). Thus optimizing (1) for any functional of θ is equivalent to optimizing $h(\phi)$. However, it is natural to ask what relevance this has to finite dimensional problems. The simulation study below demonstrates that the asymptotic optimality of accepting approximately $1/4$ of proposed moves is approximately true for dimension as low as 6.

All the distributional assumptions made in the statement of Theorem 3.1, that is the form given in (3) and the conditions on the starting values of the algorithms, can be weakened considerably. In fact, a modified version of the result (in which the optimal acceptance rate remains 0.234) remains true as long as a technical phase transition-free condition on π holds. See Roberts, Gelman, and Gilks (1994) for details.

In practice, the result can be used by monitoring the acceptance rate of iterations as suggested in the introduction and discussed in more detail in Section 4. Figure 2 shows how h varies with the jumping kernel scale factor $\phi = \sigma_d\sqrt{d}$, and with the acceptance rate p_{jump}, assuming $F = 1$. Here we see clearly that efficiency is maximised by setting $\phi = 2.38$ (Figure 2a) or by setting $p_{jump} = 0.234$ (Figure 2b).

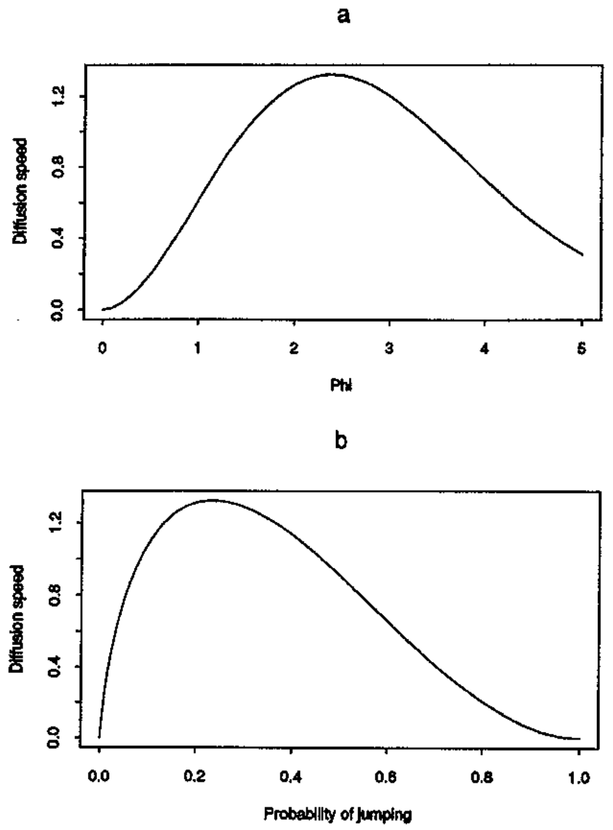

Figure 2. *Diffusion speed $h(\phi)$ related to jumping kernel scale factor ($\sigma = \phi/\sqrt{d}$) and acceptance rate p_{jump}.*

3.2. *Simulation Results for Finite d*

In the previous sections we have presented calculations for $d = 1$ and $d \to \infty$ dimensions. In this section we present the results of simulations for $d = 1, \ldots, 10$ dimensions for the multivariate standard normal target distribution, $\pi(\theta) = N_d(0, I_d)$, and multivariate normal jumping kernel, $J(\theta, \theta') = N_d(0, \sigma_d^2 I_d)$.

For each d, and for each of 21 values of σ_d between $2/\sqrt{d}$ and $3/\sqrt{d}$, we simulated one million iterations of the Metropolis algorithm with starting values drawn from $\pi(\theta)$. For each run and for $i = 1 \ldots d$, $\text{eff}_{\bar{\theta}_i}$ was estimated using the method of batching (see, for example, Hastings, 1970, or Ripley, 1987), where $\overline{\theta}_i$ denotes the sample mean for the ith coordinate of θ. With batch size 100, first-order autocorrelations in batch means were always less than 0.05. Essentially the same results were obtained with batch size 1000.

In Table 1 we have reported for each d the value of σ_d giving largest average efficiency $\mathrm{eff}_{\overline{\theta}_1}$, together with the corresponding empirical proportion of accepted jumps p_{jump}. To reduce the impact of random variation in determining the optimal σ_d, we averaged over the efficiency $\Sigma_i \mathrm{eff}_{\overline{\theta}_i}$ for $i = 1, \ldots, d$ and smoothed over the 21 values of σ_d.

Table 1. *Optimal scale factor σ_d and optimal efficiency for normal jumping kernel and standard normal target distribution in low dimensions, compared to theoretical values based on Theorem 3.1.*

Dimension, d	Optimal σ_d	$\mathrm{eff}_{\overline{\theta}_1}$	p_{jump}	$2.38/\sqrt{d}$	$0.331/d$
1	2.40	0.233	0.441	2.38	0.331
2	1.70	0.136	0.352	1.68	0.166
3	1.39	0.098	0.316	1.37	0.110
4	1.25	0.076	0.279	1.19	0.083
5	1.10	0.062	0.275	1.06	0.066
6	1.00	0.053	0.266	0.97	0.055
7	0.93	0.047	0.261	0.90	0.047
8	0.87	0.041	0.255	0.84	0.041
9	0.80	0.037	0.261	0.79	0.037
10	0.74	0.034	0.267	0.75	0.033

The results show that the asymptotically optimal $\sigma_d = 2.38/\sqrt{d}$ (from Section 3.1) applies for d as low as 1, and the asymptotic acceptance rate of 0.234 and efficiency of $0.331/\sqrt{d}$ are attained approximately by $d = 6$. Thus Theorem 3.1 accurately predicts the behavior of the optimal spherically symmetric multivariate normal jumping kernel in low dimensions.

The theory and the simulation study both support the use of an over-dispersed proposal distribution, as recommended by Besag and Green (1993) for one-dimensional sampling in multivariate problems. However for higher dimensional problems, it is advisable to have proposals with smaller variances in relation to those of the target density.

4. PRACTICAL IMPLICATIONS

Our results for the normal distribution suggest some heuristics for adaptive Metropolis simulation of more complicated distributions. In general, we can imagine a multivariate target distribution for which we have constructed a starting distribution and an iterative simulation procedure such as the Metropolis-Hastings algorithm (Hastings, 1970). After running the simulations for a while, we can monitor the convergence of the simulated sequences, perhaps using the method of Gelman and Rubin (1992), Liu, Liu, and Rubin (1992), or Roberts (1995). If the convergence is slow, it may be worthwhile to try to increase the efficiency of the simulations using whatever information is available from the simulations that have been produced so far. This idea of *adaptive* simulation has been suggested by many researchers. For example, Hills and Smith (1992) use a rough estimate of the target density to reparameterize in the Gibbs sampler; Liu and Rubin (1993) estimate the time series behavior of multiple simulated sequences in order to create a new starting distribution for an improved iterative simulation; and Muller (1993) suggests adaptively altering a Metropolis jumping rule. Incidentally, the simulations produced by an adaptive "Markov chain" simulation are not, in general, themselves a Markov chain, because the transition probabilities can depend on the results of earlier iterations (see, for example, Gelfand and Sahu, 1993).

To fix ideas, consider the following adaptive scheme for a problem with continuous parameter space: run several parallel sequences of a Metropolis algorithm, starting with samples from some approximate starting distribution, and, at some point, stop and estimate the (multivariate) target distribution and monitor the convergence of the simulations. If the simulations are still far from convergence (in the terminology of Gelman and Rubin (1992), if the potential scale reduction is much greater than 1), use the estimated target distribution to alter the Metropolis jumping rule in two ways: first, by reparameterizing, so that the target distribution is approximately spherical; and second, by setting the scale of the jumping kernel to approximately $2.38/\sqrt{d}$ times the conditional standard deviation of the target distribution along the jumping direction. This is approximately the procedure suggested by Muller (1993) (see also Tierney, 1991), but with a different variance for the jumping kernel.

While the Metropolis algorithm is running, it can be fine-tuned: Muller (1993) monitors the frequency of acceptances of the Metropolis algorithm, if the acceptance rate is much less or much more than 0.5, altering the jumping kernel by decreasing or increasing its variance, respectively. Care has to be taken when adopting this approach, since adaptation to information from previous iterations can compromise the stationarity of the target density. However, such an approach is acceptable as part of a pilot sample analysis, where adaptation stops after a fixed number of exploratory iterations.

Our computations provide some justification for such an adaptive approach. For higher dimensional jumping rules, however, a lower acceptance rate near 0.25 is preferable. Moreover, Theorem 3.1 implies that an average acceptance rate of between 0.15 and 0.4 yields at least 80% of the maximum efficiency obtainable (see Figure 2). In practice therefore, adaptation cannot be recommended when acceptance rates are within this range. Even the folklore figure of 0.5 produces reasonable results (approximately 75% of maximum possible efficiency).

The application of Theorem 3.1 is not restricted to Metropolis sampling in the full dimensional space. It remains relevant when the Metropolis step forms part of a larger dimensional (perhaps Gibbs style) algorithm. This enables the ideas to be used, for example, for posterior distributions in many hierarchical models.

Finally, we emphasize that an acceptance rate of around 0.25 does not guarantee efficiency of the algorithm. In particular, a different approach may be required to sample efficiently from highly multimodal distributions. However, when an efficient scaling does exist, it is often sufficient to only loosely tune the proposal distribution in order to obtain satisfactory results.

ACKNOWLEDGEMENTS

We thank Jun Liu and Jiming Jiang for helpful discussions and the American Lung Association for financial support. In addition, Yuval Peres independently proved a special case of Theorem 3.1.

REFERENCES

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *J. Roy. Statist. Soc. B* **55**, 25–37.

Gelfand, A. E. and Sahu, S. (1993). On Markov chain Monte Carlo acceleration. *J. Comp. Graph. Statist.* (to appear).

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statist. Sci.* **7**, 457–511.

Gilks, W. R., Clayton, D. G., Spiegelhalter, D. J., Best, N. G., McNeil, A. J., Sharples, L. D. and Kirby, A. J. (1993). Modelling complexity: applications of Gibbs sampling in medicine. *J. Roy. Statist. Soc. B* **55**, 39–52.

Green, P. J. and Han, X. (1991). Metropolis methods, Gaussian proposals and antithetic variables. *Lecture Notes in Statistics* **74**, 142–164.

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

Hills, S. E. and Smith, A. F. M. (1992). Parameterization issues in Bayesian inference. *Bayesian Statistics 4* (J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, eds.). Oxford: University Press, 227–246 (with discussion).

Liu, C. and Rubin, D. B. (1993). Markov-normal analysis of iterative simulations before their convergence. *Tech. Rep.*, Department of Statistics, Harvard University.

Liu, C., Liu, J. and Rubin, D. B. (1992). A variational control variable for assessing the convergence of the Gibbs sampler. *Proceedings of the Statistical Computing Section, American Statistical Association*, 74–78.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092.

Muller, P. (1993). A generic approach to posterior integration and Gibbs sampling. *Tech. Rep.*, ISDS, Duke University.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika* **60**, 607–612.

Ripley, B. D. (1987). *Stochastic Simulation.* New York: Wiley.

Roberts, G. O. (1995). Methods for estimating L^2 convergence of Markov chain Monte Carlo. *Bayesian Statistics and Econometrics: Essays in honor of Arnold Zellner* (D. Berry, K. Chaloner and J. Geweke, eds.). New York: Wiley(to appear).

Roberts, G. O., Gelman, A. and Gilks, W. R. (1994). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Tech. Rep.*, Statistical Laboratory, University of Cambridge.

Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J. Roy. Statist. Soc. B* **55**, 3–23 (with discussion).

Tierney, L. (1991). Exploring posterior distributions using Markov chains. *Computing Science and Statistics* **23**, 563–570.