Published in *Proc. SPIE* **3338**, pp. 371-382 (1998) LA-UR-98-1518
Medical Imaging: Image Processing, K. M. Hanson, ed.

Posterior sampling with improved efficiency

Kenneth M. Hanson and Gregory S. Cunningham

Los Alamos National Laboratory, MS P940
Los Alamos, New Mexico 87545 USA

ABSTRACT

The Markov Chain Monte Carlo (MCMC) technique provides a means to generate a random sequence of model realizations that sample the posterior probability distribution of a Bayesian analysis. That sequence may be used to make inferences about the model uncertainties that derive from measurement uncertainties. This paper presents an approach to improving the efficiency of the Metropolis approach to MCMC by incorporating an approximation to the covariance matrix of the posterior distribution. The covariance matrix is approximated using the update formula from the BFGS quasi-Newton optimization algorithm. Examples are given for uncorrelated and correlated multidimensional Gaussian posterior distributions.

Keywords: adaptive Markov Chain Monte Carlo, statistical efficiency, BFGS optimization, Bayesian analysis, uncertainty estimation

1. INTRODUCTION

In Bayesian data analysis the posterior probability distribution characterizes the uncertainty in the model estimated from a given set of data. One way to explore the posterior and hence characterize parameter uncertainty is to employ the Markov Chain Monte Carlo (MCMC) technique, which effectively generates a random sequence of model realizations that sample the posterior.

The usefulness of MCMC in Bayesian inference is well established.[3–5] In our own work we solved the difficult problem of tomographic reconstruction from two views by basing the reconstruction on deformable boundaries[6] and showed how samples from the posterior can be used to estimate uncertainties in the location of the boundary of the reconstructed object.[7,8] MCMC provided the means to verify the reliability of the reconstruction. Conventional approaches to uncertainty estimation are inadequate to treat this problem because of the nonlinear relation between the data and the model parameters and, hence, the potential nonGaussian nature of the posterior distribution.

Statistical efficiency is the central issue in MCMC, particularly when the evaluation of the posterior requires a lengthy calculation. Our ultimate goal is to treat problems involving large simulations, for example, ocean[9] or atmospheric models, 3D tomographic reconstruction,[10] aerodynamics, and hydrodynamics. Thus, it is essential to reduce the number of steps taken by an MCMC algorithm to reach a specified degree of accuracy in estimating the uncertainties in these models.

The simplest MCMC approach is to use the Metropolis algorithm to construct the sequence. In the Metropolis algorithm, one tries to move from the current position in parameter space by randomly selecting a trial step from a symmetric probability distribution. That trial step is either accepted or rejected on the basis of the probability of the new position relative to the previous one. This algorithm is widely employed because of its simplicity. Most commonly, the steps are chosen independently for each parameter. However, most posterior distributions possess some degree of correlation between the parameters. By not taking these correlations into account, the independent step distribution can lead to substantial calculational inefficiency. Our purpose is to improve the efficiency of the Metropolis algorithm by using information about the correlations.

Our approach to improving the efficiency of MCMC calculations is to obtain an approximation of the covariance matrix of the posterior probability distribution. The approximate covariance is then used to generate trial steps. The covariance matrix is developed with the aid of the gradient of $\varphi = -\log(\text{posterior})$. The gradient can be efficiently calculated using the Adjoint Differentiation In Code Technique (ADICT),[11] which can be implemented

Further author information –
K.M.H.(correspondence): E-mail: kmh@lanl.gov; WWW: http://www.lanl.gov/home/kmh/; Telephone: 505-667-1402
G.S.C.: E-mail: cunning@lanl.gov; Telephone: 505-667-2562; minor corrections made Sept. 21, 2000

for any computational code for which the output pdf is differentiable with respect to the variables in the problem. Many other schemes for improving the efficiency of MCMC exist,[12,13] most of which are adaptive. One that is related to our approach is called reparameterization in which the parameters are translated, rotated, and scaled into a different coordinate system with the purpose of making the covariance matrix in the reparameterized coordinate system uncorrelated and isotropic.[12]

2. BAYESIAN INFERENCE WITH MARKOV CHAIN MONTE CARLO

Bayesian inference is related to the characterization of the posterior probability, which summarizes uncertainty about model parameters. Essentially all inference about uncertainties in models, the reliability of their potential predictions, and so forth, stems from the posterior. The MCMC technique provides a means to generate a set of random samples from an arbitrary probability density function (pdf), which in Bayesian analysis is the posterior. Given a set of N_k random parameter vectors $\{\mathbf{x}_k\}$ drawn from a pdf $p(\mathbf{x})$, one can estimate the expectation value of any function $f(\mathbf{x})$

$$\langle f(\mathbf{x})\rangle = \int f(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{N_k}\sum_{k=1}^{N_k} f(\mathbf{x}_k) \quad . \tag{1}$$

For example, the mean value of the parameters ($f(\mathbf{x}) = \mathbf{x}$) is an estimator for the parameters $\mathbf{x}$. Furthermore, the variance of the estimate $\hat{\mathbf{x}}$ is

$$var(\hat{\mathbf{x}}) = \left\langle (\mathbf{x}-\hat{\mathbf{x}})^2 \right\rangle = \int (\mathbf{x}-\hat{\mathbf{x}})^2\, p(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{N_k}\sum_{k=1}^{N_k} (\mathbf{x}_k - \hat{\mathbf{x}})^2 \quad , \tag{2}$$

is a measure of its uncertainty. Correlations between the uncertainties in the components of $\hat{\mathbf{x}}$ are crucial for making inferences and these can also be estimated from the sequence $\{\mathbf{x}_k\}$.

The MCMC technique makes it feasible to perform some of the difficult technical calculations required by probability theory[5,3] (normalization of pdfs, marginalization, computation of expectation integrals, model selection) in a computer. The MCMC technique has opened up the possibility of applying Bayesian analysis to complex analysis problems. A desirable attribute of MCMC is that there are generally no restrictions on the types of pdfs that can be sampled; no functional form for the pdf is required. In its basic form, MCMC only requires that one be able to calculate $\varphi = -\log(\text{posterior})$, although sometimes the gradient of φ is needed, as in the case of the present algorithm.

2.1. Statistical Efficiency of a MCMC Sequence

The uncertainty in estimates of quantities derived from an MCMC sequence of random samples is a central issue. Suppose that we are given a sequence of N_k samples v_k representing a pdf of a scalar quantity v and that the samples are assumed to be drawn from that pdf by a stationary process. The estimated value of v is given by the sample mean,

$$\hat{v} = \frac{1}{N_k}\sum_{k=1}^{N_k} v_k \quad . \tag{3}$$

The expected variance in $\hat{v}$ is the expectation over the ensemble of all such sequences:

$$\sigma_{\hat{v}}^2 = var(\hat{v}) = \mathrm{E}\left\{(\hat{v} - E\{\hat{v}\})^2\right\} = \mathrm{E}\left\{ \frac{1}{N_k}\sum_j (v_j - \bar{v}) \frac{1}{N_k}\sum_k (v_k - \bar{v}) \right\} = \frac{1}{N_k^2}\sum_{jk} \mathrm{E}\left\{(v_j-\bar{v})(v_k-\bar{v})\right\} \quad , \tag{4}$$

where $\bar{v} = E\{v\} = E\{\hat{v}\}$. The autocovariance of a sequence is defined as $E\{(v_k - \bar{v})(v_{k+l} - \bar{v})\}$. The normalized autocovariance is $\rho(l) = (\sigma^2)^{-1}E\{(v_k - \bar{v})(v_{k+l} - \bar{v})\}$, where σ^2 is the variance of v and $\rho(l)$ does not depend on k, by stationarity. When the sequence is long compared to the length of the nonzero normalized autocovariance values,

$$\sigma_{\hat{v}}^2 = \frac{1}{N_k^2}\sum_l \sum_{k=1}^{N_k - l} \mathrm{E}\left\{(v_k-\bar{v})(v_{k+l}-\bar{v})\right\} = \frac{\sigma^2}{N_k}\sum_{l=-\infty}^{\infty} \rho(l) \quad , \tag{5}$$

because the v_k are from a stationary sequence. We note that the normalized autocovariance is a symmetric function, i.e., $\rho(-l) = \rho(l)$.

If the sequence has sufficiently converged to the target pdf, the variance of the distribution is approximately the variance of the samples:

$$\sigma_v^2 \approx S^2 = \frac{1}{N_k - 1} \sum_{j=1}^{N_k} (v_k - \hat{v})^2 \ , \tag{6}$$

and the normalized autocovariance may be estimated from the sequence:

$$\rho(l) \approx \frac{1}{S^2} \frac{1}{(N_k - l - 1)} \sum_{k=1}^{N_k - l} (v_k - \hat{v})(v_{k+l} - \hat{v}) \ , \tag{7}$$

for lag $l \geq 0$.

The statistical efficiency of an MCMC sequence is defined as the reciprocal of the ratio of the number of MCMC trials needed to achieve the same variance in an estimated quantity as are required for independent draws from the target probability distribution. For the estimation of the mean, the variance for independent sampling would be σ_v^2/N_k in our present case. We see that the statistical efficiency is

$$\eta = \left[\sum_{l=-\infty}^{\infty} \rho(l) \right]^{-1} = \left[1 + 2 \sum_{l=1}^{\infty} \rho(l) \right]^{-1} \ . \tag{8}$$

For stationary Markov chains, because the value v_k depends in a probabilistic way only on the value of the preceding element in the chain, i.e., v_{k-1}, the normalized autocovariance function has an exponential behavior.

2.2. Gaussian Probability Distributions

We recall a few important relationships for a multivariate Gaussian pdf. Writing the pdf as

$$p(\mathbf{x}) = Z^{-1} exp(-\varphi(\mathbf{x})) \ , \tag{9}$$

then for a Gaussian distribution

$$\varphi(\mathbf{x}) = \tfrac{1}{2} (\mathbf{x} - \mathbf{x}_0)^{\mathrm{T}} \, \mathbf{B} \, (\mathbf{x} - \mathbf{x}_0) \ , \tag{10}$$

where $\mathbf{B}$ is the Hessian, i.e., the second derivative matrix of φ with respect to the vector of parameters $\mathbf{x}$, and $\mathbf{x}_0$ is the central position of the Gaussian.

The connection between the Hessian and the covariance matrix $\mathbf{C}$ of the probability distribution $p(\mathbf{x})$ is:

$$\mathbf{C} = \left\langle (\mathbf{x} - \mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)^{\mathrm{T}} \right\rangle_p = \mathbf{B}^{-1} \ . \tag{11}$$

3. METROPOLIS ALGORITHM

In MCMC the objective is to generate a sequence of parameter sets that mimic a specified target pdf, let's call it $q(\mathbf{x})$, where $\mathbf{x}$ is a vector of parameters in the relevant parameter space. MCMC is resorted to in cases in which the functional nature of $q(\mathbf{x})$ is unknown for which analytic methods of analysis are precluded. This situation often occurs when complex models are required to predict the measurements. To clarify further, the only thing that one can know from a complex simulation is the value of $q(\mathbf{x})$ for a specific $\mathbf{x}$, or perhaps the gradient of $-\log[q(\mathbf{x})]$ with respect to $\mathbf{x}$. The process of exploring an unknown pdf is somewhat like feeling one's way in the dark; nothing is known until one tries to take a step.

The goal of the MCMC algorithm is to create a sequence that is in statistical equilibrium with the target pdf $q(\mathbf{x})$. This goal can be achieved when the MCMC sequence satisfies the condition of detailed balance[14,15]:

$$q(\mathbf{x}) \, T(\mathbf{x} \rightarrow \mathbf{x}') = q(\mathbf{x}') \, T(\mathbf{x}' \rightarrow \mathbf{x}) \ , \tag{12}$$

where $T(\mathbf{x} \rightarrow \mathbf{x}')$ is the transition probability of moving from $\mathbf{x}$ to $\mathbf{x}'$. Detailed balance essentially requires that, in a very long sequence, the number of moves from $\mathbf{x}$ to $\mathbf{x}'$ is identical to the number from $\mathbf{x}'$ to $\mathbf{x}$.

Two important practical issues in MCMC are convergence and burn in.[3] Since sequences may be started from an arbitrary point, any particular sequence may take some time to equilibrate with the target pdf, that is, reach convergence. Therefore, one must try to determine when the sequence has reached convergence, a process that is often carried out by monitoring the sequence itself. This "burn in" period must be discarded for subsequent analysis as it does not represent the pdf. One very good way to determine convergence is to run multiple sequences starting each with disparate parameter values.[16] The sequences are taken to have converged when they coalesce into a common distribution. For more detailed information about MCMC, the reader is referred to the excellent book edited by Gilks et al.[3]

3.1. Metropolis Algorithm

One of the simplest algorithms used in MCMC calculations is due to Metropolis et al.[14] This algorithm ensures detailed balance (12) for each step in the sequence. One starts at an arbitrary point in the vector space to be sampled, $\mathbf{x}_0$. The general recursion at any point in the sequence $\mathbf{x}_k$ is to repeat the following cycle many times:

```
(1) Select a new trial position $\mathbf{x}^* = \mathbf{x}_k + \Delta\mathbf{x}$,
    where $\Delta\mathbf{x}$ is randomly chosen from a symmetric step distribution
(2) Calculate the ratio $r = q(\mathbf{x}^*)/q(\mathbf{x}_k)$
(3) Accept the trial position, that is, set $\mathbf{x}_{k+1} = \mathbf{x}^*$,
        if $r \geq 1$,
        or with probability $r$, if $r < 1$,
    otherwise, stay put, $\mathbf{x}_{k+1} = \mathbf{x}_k$ .
```

This algorithm is used in the majority of current MCMC research and works remarkably well.

3.2. Various Approaches to MCMC

The choice of the best MCMC technique for a specific target pdf is a balancing act. In general, it is desirable to incorporate what is known about the target pdf in the design of the MCMC sampling method but that must be done with caution, lest the prior information be false. Almost always, one has to trade off efficiency against robustness. The Metropolis approach is quite robust, allowing for fairly general unimodal target pdfs, but sometimes fairly inefficient. Target pdfs with multiple modes often require special treatment.[12]

In the Metropolis algorithm, trials are limited to steps taken away from the present position, which must be drawn from a symmetric pdf. Almost any symmetric step distribution will work, but some choices will enhance the statistical efficiency of the MCMC process for a particular target pdf. The use of a Gaussian step distribution is standard and will permit the sequence to converge to most nonpathelogical target pdfs.

Hastings suggested an extension to the basic Metropolis algorithm to permit use of an asymmetric step distribution.[17] The Hastings form permits arbitrary proposal distributions. As an example, if the target pdf is known to be Gaussian, one can draw trials from a fixed Gaussian pdf that approximates the target pdf, in the same vein as the well-known importance-sampling approach to variance reduction in evaluating integrals of pdfs.[15] If the proposal distribution matches the target pdf well, nearly all trial moves will be accepted. The resulting behavior will be similar to drawing independent samples from the pdf and the statistical efficiency can approach 100%. However, if the target pdf is not Gaussian, this approach may result in terrible inefficiency. If the target pdf were an exponential, for example, the use of a fixed Gaussian proposal distribution would severely hinder adequate sampling of the tails of the exponential.

It is potentially useful to monitor the MCMC process with the goal of detecting departures from the assumptions on which the MCMC method is based.[13]

4. EXAMPLES OF STANDARD USE OF METROPOLIS ALGORITHM

We now present some examples of the use of the Metropolis algorithm to generate MCMC sequences, starting with the case of a 1D Gaussian distribution. For multidimensional target pdfs, the step distribution most often used in the Metropolis version of MCMC is an uncorrelated, isotropic (same variance for every variable) Gaussian distribution. We demonstrate the performance of this standard choice for the simplest of target pdfs in multidimensions, namely an uncorrelated, isotropic Gaussian. The difficulty with this choice for nonisotropic Gaussians is also described. We emphasize that the choice of Gaussian target pdfs for these examples is only for simplicity of presentation. The

Metropolis-MCMC technique is robust and will handle most other pdfs. The advantages that we claim for our adaptive technique, presented in the next section, should apply to other kinds of pdfs as well.

The examples presented here were obtained using the advanced image-processing language, IDL,[18] which is an excellent tool for developing algorithms. However, as a cautionary note to IDL users, we found that calls to the random-number generation procedures, RANDOMU and RANDOMN, must employ the same seed variable throughout a calculation. Otherwise, the random numbers generated from the two routines possess some degree of correlation. This unexpected behavior is not documented in the IDL manuals, as far as we can tell.

4.1. Two-dimensional Gaussian Target Distribution

Figure 1 shows examples of the MCMC sequences generated for an uncorrelated two-dimensional problem with the Metropolis algorithm for various widths of the step distribution σ_T. For a modest change of the width of the step distribution by only a factor of four, these sequences show quite different behavior. When σ_T is small compared with the width of the target distribution σ_0, the steps are small relative to σ_0. The behavior mimics that of Brownian motion, that is, an unconstrained random walk. The curve actually has a fractal nature,[?] although this realization doesn't seem to have been used to benefit for understanding MCMC. It obviously takes many steps to sample the full width of the target distribution. It is exactly this situation that practitioners of MCMC must guard against, lest they believe that they have reached convergence too early in the sequence. When $\sigma_T \approx \sigma_0$, the movement through the target distribution appears to be much more efficient, approaching that expected for samples drawn independently from the target distribution. When σ_T is somewhat larger than σ_0, the behavior is entirely different again. Because the proposal distribution so much broader than the target distribution, the trial steps are often not accepted, so the movement is sticky. On the other hand, the trial steps that are accepted are usually large enough to jump clear across the target distribution.

The normalized autocovariance functions for the sequences from which the three segments in Fig. 1 are taken are shown in Fig. 2. The normalized autocovariance functions are very nearly exponential, reflecting the property of Markov chains that each state depends only on the previous state. As shown in Sect. 2.1, the statistical efficiency of a MCMC sequence is reciprocally related to the sum under the normalized autocovariance of the sequence. Thus, the efficiency is approximately $(1 + 2\tau)^{-1}$, where τ is the decay length of exponential that describes the normalized autocovariance dependence. From the sum under the normalized autocovariance functions, we find the efficiency is 1.3% for $\sigma_T = 0.25$, 10.1% for $\sigma_T = 1$, and 7.3% for $\sigma_T = 4$. The fraction of proposed steps that are accepted in these cases is 88%, 57%, and 12%, respectively. For this 2D problem, the highest statistical efficiency of 14% is achieved for $\sigma_T = 2$ with an acceptance probability of 31%.

4.2. Uncorrelated, Isotropic Multidimensional Gaussian Target Distributions

Proceeding in the same vein as the preceding section, we now consider the behavior of the Metropolis algorithm over a range of dimensions from one to 64. In all cases the target pdf is an isotropic Gaussian with unit variance in all directions and without any correlation among the parameters. Figure 3 summarizes the results of this study in terms of the statistical efficiency. In these examples, we do not experience any problems of convergence and so it is not necessary to burn in the sequences. The boxes in Fig. 3 show the expected efficiencies and optimum widths of the step distribution derived from Langevin diffusion theory[19] as a function of the dimension of the problem n. These operating points, which obey the scaling laws $\eta = 0.3\, n^{-1}$ and $\sigma_T = 2.4\, n^{-\frac{1}{2}}$, agree well with our numerical results. The challenge for dealing with very large problems is to find a way to overcome this lamentable n^{-1} drop off in statistical efficiency.

During the MCMC procedure, the easiest thing to monitor is the jump or acceptance probability, that is, the fraction of trials steps that are accepted. We show the efficiency as a function of the acceptance for this study in Fig. 4, which confirms the rule of thumb that for a moderate number of dimensions the best operating point for the Metropolis-MCMC algorithm is at an acceptance of about 25%.

4.3. Nonisotropic Gaussian Target Distributions

Consider a two-dimensional problem in which the pdf is an uncorrelated 2D Gaussian, but with the rms width in one direction four times larger than in the other. Because the pdf is uncorrelated, the behavior in one direction is largely uncoupled from the other direction. From the earlier example in 2D, we recognize that if one were to use an isotropic 2D Gaussian for the step distribution, a major difficulty will ensue. The rms width for the best performance

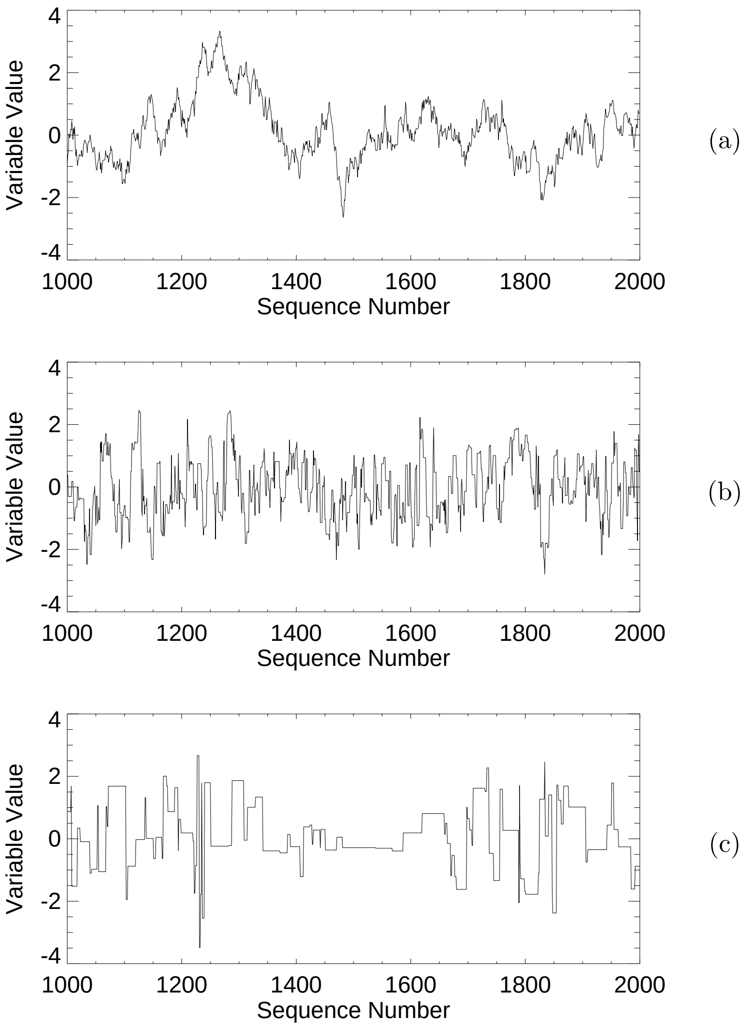

Figure 1. Examples of MCMC sequences generated for a two-dimensional Gaussian target distribution of unit rms width, showing 1000 consecutive values for a step distribution of rms width (a) $\sigma_T = 0.25$, i.e., one quarter the width of the target distribution; (b) $\sigma_T = 1$ and (c) $\sigma_T = 4$.

in the narrower direction will be too small to achieve good efficiency in the wider direction. In fact, the normalized autocovariance function will have a different behavior in the two directions. Hence, the efficiency in this conceptual problem will differ for each component. To generalize this observation, the statistical efficiency of an estimate derived from an MCMC sequence depends on the components on which that estimate depends and their possibly different efficiencies.

In all of these previous examples, we have dealt with uncorrelated Gaussian distributions. Clearly, correlations between parameters will complicate matters further. However, these correlations are tremendously important in making inferences so that it is crucial for us to learn how to deal with them. An adaptive approach provides a way to cope with asymmetric and correlated target pdfs.

5. AN ADAPTIVE APPROACH TO MCMC

The basic idea behind the proposed algorithm is to use an approximation to the covariance matrix of the target probability distribution for the step distribution in a Metropolis algorithm. This algorithm is adaptive in the

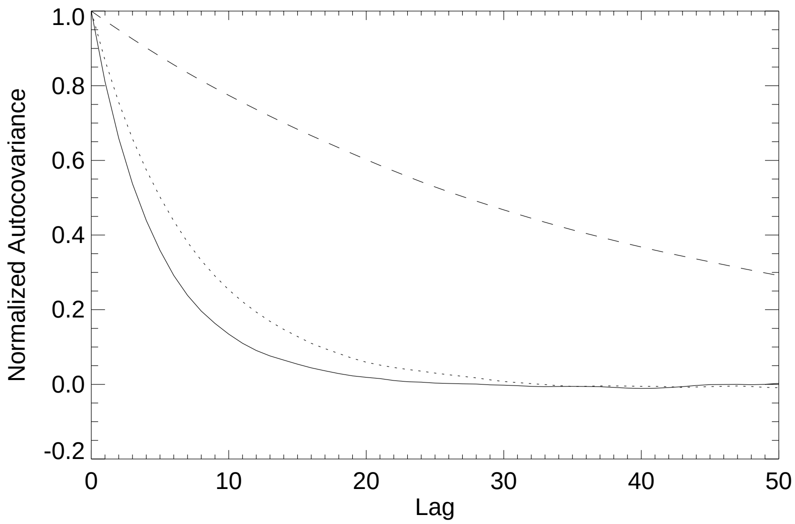

Figure 2. The normalized autocovariance function for the three MCMC sequences represented in Fig. 1. The solid line is for a width of the step distribution σ_T equal to 1, the dashed line for 0.25, and the dotted line for 4. As the efficiency of the MCMC algorithm is inversely proportional to the sum under the normalized autocovariance from $-\infty$ to ∞, $\sigma_T = 1$ is the most efficient choice of these widths in this situation.

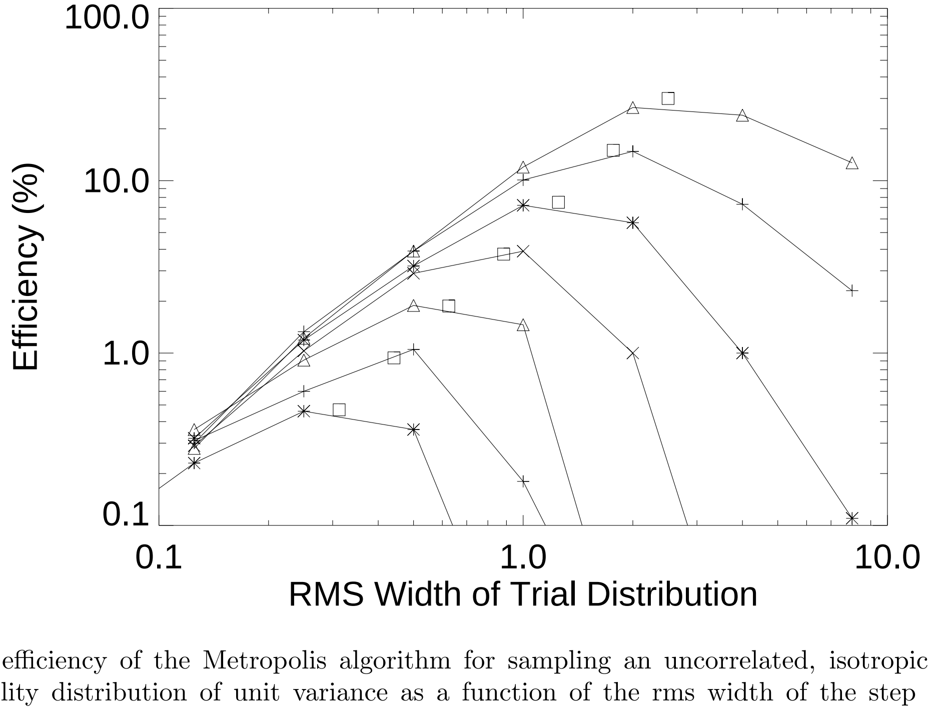

Figure 3. The efficiency of the Metropolis algorithm for sampling an uncorrelated, isotropic multidimensional Gaussian probability distribution of unit variance as a function of the rms width of the step distribution. The dimension of the problem varies by factors of two from 1 for the top curve to 64 for the bottom. The scaling law discussed in the text, indicated by the boxes, correctly predicts the efficiency at the optimum operating points.

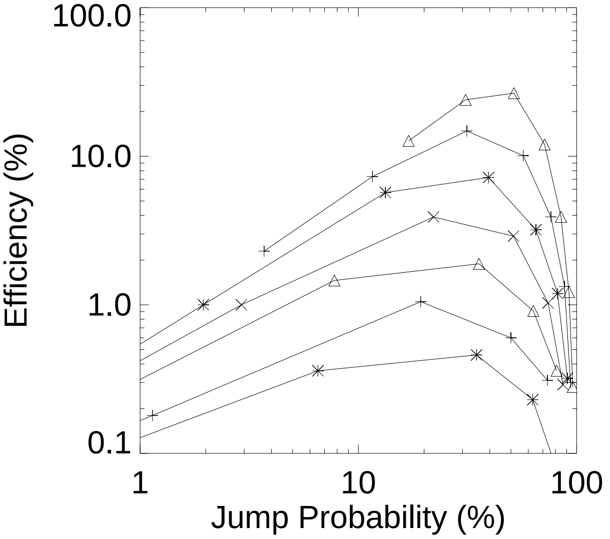

Figure 4. The efficiency of the Metropolis algorithm for sampling an uncorrelated, isotropic multidimensional Gaussian probability distribution of unit variance as a function of the acceptance probability for proposed steps. The dimension of the problem varies by factors of two from 1 for the top curve to 64 for the bottom.

sense that an early phase of the MCMC sequence is used to approximate the covariance matrix borrowed from an optimization algorithm described next. The full covariance matrix is approximated, including off-diagonal elements, i.e., correlations between various parameters. The trial steps are then drawn from a Gaussian with the approximated covariance. In the present approach, the proposed steps are centered on the present position $\mathbf{x}_k$. Other possibilities exist, of course, and they may be useful. For example, one alternative approach is to use the gradient of φ to offset the step distribution so that it coincides better with the target pdf, as in the so-called Langevin-Hastings algorithm.[12]

Adaptive approaches are employed in many MCMC algorithms.[12,13] Furthermore, it has been shown that adaptive Monte Carlo algorithms simulating particle transport can be constructed in which the variance drops exponentially with the number of iterations n[?] instead of with the usual n^{-1} behavior.

5.1. BFGS Optimization Algorithm

There are a variety of optimization techniques that make use of the gradient of the function to be optimized φ. The general idea behind the so-called quasi-Newton optimization techniques is to build up an approximate expression for the Hessian (the second-derivative matrix of φ with respect to all the variables), or the inverse Hessian, and use it to take an approximate Newton step. The most recently developed version of the quasi-Newton methods is the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm, which has largely replaced the earlier Davidon-Fletcher-Powell (DFP) algorithm. The BFGS algorithm has the advantage over DFP in that it does not require accurate line minimizations along the quasi-Newton directions to build up the approximate Hessian.[?] Thus, BFGS potentially reduces the number of function evaluations required to complete an optimization procedure.

In the kth iteration of the BFGS optimization procedure, the change in the parameter vector from its present value $\mathbf{x}_k$ is based on the gradient $\mathbf{g}_k = \frac{\partial \varphi}{\partial \mathbf{x}}|_{\mathbf{x}_k}$ and the present estimate of the inverse Hessian $\mathbf{C}_k$, which is the covariance matrix of the corresponding Gaussian probability distribution, as indicated in Sect. 2.2. The position of the minimum is estimated by the Newton formula: $\mathbf{x}^*(\alpha) = \mathbf{x}_k - \alpha\,\mathbf{C}_k\mathbf{g}_k$. Starting with $\alpha = 1$, a line search is conducted to find the value of $\alpha = \alpha_k$ that minimizes $\varphi(\mathbf{x}^*)$, which yields the new estimate of the position of the minimum $\mathbf{x}_{k+1} = \mathbf{x}^*(\alpha_k)$. Designating the change in position by $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and the corresponding change in the gradient by $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, the BFGS update formula for the covariance matrix is

$$\mathbf{C}_{k+1} = \mathbf{V}_k^{\mathrm{T}}\mathbf{C}_k\mathbf{V}_k + c_k\,\mathbf{s}_k\mathbf{s}_k^{\mathrm{T}}\ , \qquad (13)$$

where $\mathbf{V}_k = \mathbf{I} - c_k\, \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}}$ and $c_k = (\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k)^{-1}$. It is easy to show that $\mathbf{C}_{k+1}$ satisfies the Newton condition for the last step $\mathbf{C}_{k+1}\, \mathbf{y}_k = \mathbf{s}_k$, but this relation may not necessarily hold for earlier steps, i.e., $\mathbf{C}_{k+1}\, \mathbf{y}_j = \mathbf{s}_j$ for $j < k$. The process usually begins by taking $\mathbf{C}_1$ as a diagonal matrix, often just the identity matrix.

Important to note for applications involving many variables is that the covariance matrix does not have to be stored explicitly as a matrix. Rather, whenever the product of the covariance matrix with a vector is desired, for example to estimate the next step, the expansion (13) may be used. Therefore, only the parameter vector and the gradient at each iteration of the optimization need to be stored. However, the present approach to MCMC requires making random draws from the correlated Gaussian distribution with the estimated covariance matrix. For this, the square root of the covariance matrix is needed, as described below. Thus, an algorithm for finding that square root needs to be found in the case that one has only an expansion for the covariance matrix. When the number of terms in the BFGS expansion (13) is not too large, the square root can be obtained for the subspace spanned by the displacement vectors $\{\mathbf{s}_k\}$ using the approach we relate below, that is, by SVD.

5.2. Adjoint Differentiation

The BFGS update equation requires the derivatives of φ with respect to all the parameters. Fortunately, there is a technique to efficiently calculate these gradients, even for complicated forward calculations. The technique, which we have called Adjoint Differentiation In Code Technique (ADICT), essentially evaluates the chain rule for differentiation through the use of an auxiliary code that effectively reverses the data flow of the forward calculation. This approach typically generates the gradient of φ in a computation time comparable to the forward calculation. A new application called the Tangent linear and Adjoint Model Compiler (TAMC) developed by Ralf Giering[20] provides an automatic means to create the adjoint code for FORTRAN programs. TAMC has successfully been used to generate sensitivities for a 1D hydrodynamics code[21] and for an ocean-modeling code.[9]

5.3. Random Draws from a Correlated Gaussian Distribution

Once one has an approximate covariance matrix $\mathbf{C}$, generation of a random vector γ with a Gaussian distribution that possess that covariance is accomplished by the well-known means of evaluating

$$\gamma = \mathbf{C}^{\frac{1}{2}} \xi \ , \tag{14}$$

where ξ is a vector whose components are random numbers independently drawn from a unit-variance Gaussian distribution and $\mathbf{C}^{\frac{1}{2}}$ is the unique square-root matrix[?] satisfying $\mathbf{C}^{\frac{1}{2}} \mathbf{C}^{\frac{1}{2}} = \mathbf{C}$. This matrix is calculated by performing a Singular Value Decomposition (SVD) on $\mathbf{C}$:

$$\mathbf{C} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^{\mathrm{T}} \ , \tag{15}$$

where $\mathbf{U}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a diagonal matrix with the singular values on the diagonal, i.e., $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, ...)$. The square root of the covariance matrix is then:

$$\mathbf{C}^{\frac{1}{2}} = \mathbf{U} \boldsymbol{\Lambda}^{\frac{1}{2}} \mathbf{U}^{\mathrm{T}} \ , \tag{16}$$

where $\boldsymbol{\Lambda}^{\frac{1}{2}} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, ...)$.

5.4. Example - Nonisotropic, Correlated Multidimensional Gaussian

To demonstrate the usefulness of our adaptive MCMC approach, we present an example consisting of a target distribution that is a 16-dimensional Gaussian distribution with a high degree of correlation. This example is motivated by a typical type of regularization used in solving ill-posed problems. The kind of constraint considered is often used to promote smoothness for an inversion problem, namely the integral over the square of the second derivative of the field being reconstructed. In the context of one dimension, consider a function $y(x)$. The smoothness of y is promoted by trying to minimize

$$\pi(y) = \int \left| \frac{d^2 y}{dx^2} \right|^2 dx \ . \tag{17}$$

In a Bayesian analysis, this kind of function would be thought of as the $-\log$ (prior). In a discretely sampled representation, for which the vector $\mathbf{y}$ represents samples of y as a function of x, (17) becomes

$$\pi(\mathbf{y}) = \tfrac{1}{2} \sum_i \left| \tfrac{1}{2} y_{i-1} - y_i + \tfrac{1}{2} y_{i+1} \right|^2 \ . \tag{18}$$

This expression is similar to the smoothness constraint that we used to control the deformable boundary in our reconstruction of shape from two views.[8]

The second derivative of (18) is a Hessian with values [0.25, -1, 1.50, -1, 0.25] centered on the diagonal element of each row. For simplicity we construct the Hessian to be circulant, which is appropriate for a periodic sequence y_k. The inverse of the Hessian yields the covariance matrix. The singularity of this Hessian is avoided by adding a small value (0.05) to the diagonal, which may be thought of as arising from an additional contribution to $\pi(\mathbf{y})$ from $|\mathbf{y}|^2$. A 5×5 piece from the 16×16 covariance matrix for this problem is:

$$\begin{matrix} 4.97 & 3.98 & 2.50 & 1.24 & 0.42 \\ 3.98 & 4.97 & 3.98 & 2.50 & 1.24 \\ 2.50 & 3.98 & 4.97 & 3.98 & 2.50 \\ 1.24 & 2.50 & 3.98 & 4.97 & 3.98 \\ 0.42 & 1.24 & 2.50 & 3.98 & 4.97 \end{matrix}$$

There is clearly a high degree of correlation, which, from our earlier discussion, should lead to inefficiencies for a standard Metropolis-MCMC calculation. In fact, after adjusting the width of an isotropic step distribution to obtain the highest efficiency $\sigma_T = 0.5$, the best achievable efficiency for this problem is 0.11%. This efficiency is a lot smaller than predicted by the Langevin scaling law, $0.3/16 = 1.9\%$, presumably because of the high degree of correlation in the target pdf. The rms difference between the elements of the actual covariance matrix and that estimated directly from a 100000-long MCMC sequence based on this adaptive technique is 0.27, or 5.4% relative to the maximum elements.

For the learning phase of our adaptive MCMC algorithm, we used the first 100 distinct steps of an MCMC sequence based on an isotropic Gaussian step distribution with an assumed rms width of two in each variable. The initial covariance is taken to be $\mathbf{C}_1 = \mathrm{diag}(4, 4, 4, ...)$. A 5×5 piece from the full 16×16 covariance matrix estimated using the BFGS updating formula is

$$\begin{matrix} 3.79 & 2.93 & 1.78 & 0.92 & 0.34 \\ 2.93 & 4.07 & 3.40 & 2.22 & 1.12 \\ 1.78 & 3.40 & 4.63 & 3.72 & 2.25 \\ 0.92 & 2.22 & 3.72 & 4.59 & 3.49 \\ 0.34 & 1.12 & 2.25 & 3.49 & 4.34 \end{matrix}$$

We observe that this estimated matrix is not the same as the actual known covariance matrix given above, but it does possess a similar amount of correlation, as determined by the off-diagonal elements. The rms difference between the estimated covariance matrix and the actual one is 0.28, or 5.6% of the maximum elements. For the principal MCMC run, we draw random steps from this estimated covariance matrix in the Metropolis algorithm and multiply them by a factor of 0.5 to obtain the best efficiency. The efficiency with this adaptive approach is 1.62%, a vast improvement over the standard nonadaptive Metropolis algorithm by over an order of magnitude. This efficiency is almost the same as for uncorrelated Gaussians, which makes sense because the use of the approximate covariance matrix essentially removes the effect of correlation in the target pdf. The rms difference between the elements of the actual covariance matrix and that estimated directly from a 100000-long MCMC sequence based on this adaptive technique is 0.070, or 1.4% relative to the maximum elements. This result is about four times more accurate than for the nonadaptive approach for the same number of sequence samples, consistent with the ratio of the efficiencies.

In separate runs, we find that the accuracy of the BFGS updating formula improved with more training steps. For 1000 distinct steps, the rms difference between the estimated covariance matrix and the actual one is reduced to 0.003, i.e., by a factor of 100 compared to above. However, this more accurate covariance matrix does not improve the accuracy of the covariance matrix estimated from the MCMC sequence with 100000 samples, mainly because that is limited by the sampling accuracy.

A visual comparison of the functioning of these two MCMC calculations is shown in Fig. 5. The long-range correlations as a function of sequence number (displayed in the horizontal direction) in the results using the standard Metropolis algorithm are obvious. These correlations are drastically reduced in Fig. 5(b) through the use of the estimated covariance matrix for the step distribution. Of course, the best efficiency (100% by definition) is obtained by drawing independent samples directly from the known pdf, as shown in Fig. 5(c), which cannot normally be done because the pdf is not usually known. Otherwise, MCMC would not be needed.

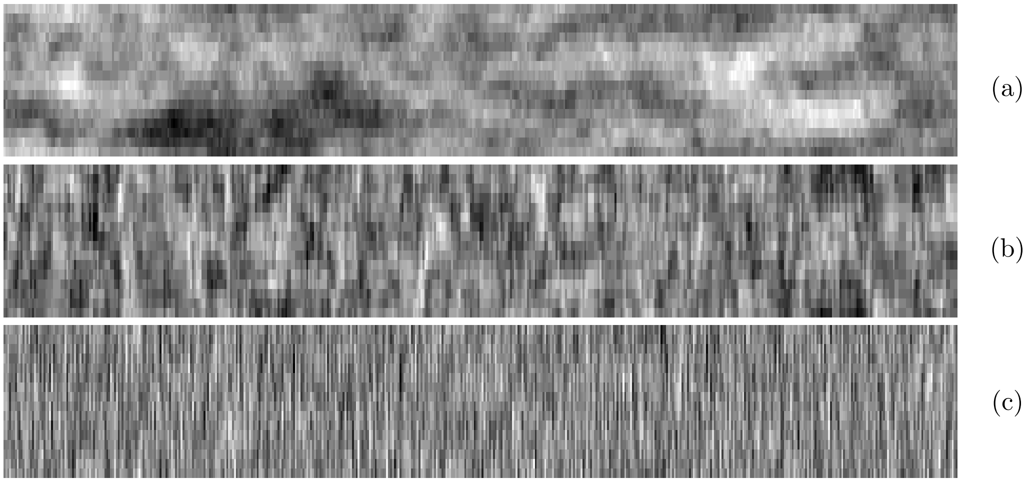

Figure 5. A gray-scale display of portions of MCMC sequences representative of a 16-parameter pdf with a high degree of correlation between parameters. The 16 parameter values are displayed vertically while the sequence numbers run horizontally. Every tenth step from a 5000-step portion of each sequence is displayed for (a) the standard nonadaptive Metropolis algorithm ($\eta = 0.11\%$), (b) our adaptive Metropolis algorithm ($\eta = 1.62\%$), and (c) independent random samples drawn directly from the pdf ($\eta \equiv 100\%$).

6. DISCUSSION

We have shown that it is possible to estimate the covariance matrix that characterizes a pdf by using a formula from the BFGS optimization algorithm. This approach requires evaluations of the gradient of $-\log(\text{posterior})$ at a set of parameter coordinates. Use of the approximate covariance matrix for the step distribution in the Metropolis algorithm can improve its efficiency compared to the standard approach of using an isotropic, uncorrelated distribution.

We noted above that the BFGS update formula for the covariance matrix only satisfies the Newton condition for the last update. Thus, after 100 learning steps in our example above, the covariance matrix is only approximately determined. In fact, 16 linearly independent steps should be sufficient to completely determine the covariance matrix. Obviously, our algorithm could be improved by determining the covariance matrix that satisfies the Newton condition for all the learning steps by solving the set of linear equations implied by the corresponding Newton conditions.

Even though the correlations in a target pdf may be taken into account with our adaptive approach, the version of the Metropolis algorithm that employs a symmetric pdf to take trial steps from the present position is quite inefficient for high-dimensional problems. The best that one can do is to attain the efficiency from Langevin diffusion theory, $\eta = 0.3/n$, for n parameters, applicable for uncorrelated, isotropic target pdfs. New approaches are required for many dimensions.

One way to beat the Metropolis performance is to place the proposal distribution at the estimated peak position and use the Metropolis-Hastings algorithm, as mentioned in SEct. 3.2. If the target distribution is truly Gaussian and the covariance matrix and peak position are accurately estimated, the efficiency of a Metropolis-Hastings algorithm approaches 100%. Of course, if these condition do not hold, the efficiency may drop tremendously. However, corrective action may be taken on the basis of what one learns in the course the MCMC sequence. Alternatively, the proposal distribution may be placed somewhere between the peak position and present location, which would help avoid the difficulties that might obtain when these conditions didn't hold but at the same time lower the efficiency.

The hybrid Monte Carlo method[2] seems worth investigating. That algorithm requires calculation of the gradient of $\varphi = -\log(\text{posterior})$, just as our adaptive algorithm does. The claim for the hybrid algorithm is that it can take large steps through the target probability distribution and still maintain high acceptance probabilities. It is a generalization of the Hastings-Langevin MCMC algorithm[12] in which the gradient of φ is used to offset the proposal distribution to better align it better with the target pdf. Approximate knowledge of the covariance matrix might improve the efficiency of these algorithms.

ACKNOWLEDGEMENTS

Many people have helped us understand MCMC and its potential usefulness, including Julian Besag, - Kalos, John Skilling, James Gubernatis, and Richard Silver. David Schmidt discovered an insidious property of the random number generator in IDL that initially resulted in small systematic defects in MCMC sequences. This work was supported by the United States Department of Energy under contract W-7405-ENG-36.

REFERENCES

1. K. M. Hanson and G. S. Cunningham, "Posterior sampling with improved efficiency," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE* **3338**, 1998 (to be published).
2. R. M. Neal, *Bayesian Learning for Neural Networks*, Springer, New York, 1996.
3. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.
4. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman & Hall, London, 1995.
5. J. Besag, P. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Stat. Sci.* **10**, pp. 3–66, 1995.
6. K. M. Hanson, G. S. Cunningham, G. R. Jennings, Jr., and D. R. Wolf, "Tomographic reconstruction based on flexible geometric models," in *Proc. IEEE Int. Conf. Image Processing, vol. II*, pp. 145–147, IEEE, 1994.
7. K. M. Hanson, "Bayesian reconstruction based on flexible prior models," *J. Opt. Soc. Amer. A* **10**, pp. 997–1004, 1993.
8. K. M. Hanson, G. S. Cunningham, and R. J. McKee, "Uncertainty assessment for reconstructions based on deformable models," *Int. J. Imaging Systems and Technology* **8**, pp. 506–512, 1997.
9. G. Burgers, R. Giering, and M. Fischer, "Construction of the adjoint of the HOPE OGCM," *Ann. Geophysicae.* **C14**, p. 390, 1996.
10. G. S. Cunningham, K. M. Hanson, and X. L. Battle, "Three-dimensional reconstructions from low-count SPECT data using deformable models," *Opt. Express* **2**, pp. 227–236, 1998.
11. K. M. Hanson, G. S. Cunningham, and S. S. Saquib, "Inversion based on computational simulations," in *Maximum Entropy and Bayesian Methods*, G. Erickson, ed., Kluwer Academic, Dordrecht, 1998 (to be published).
12. W. R. Gilks and G. O. Roberts, "Strategies for improving MCMC," in *Markov Chain Monte Carlo in Practice*, W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, eds., pp. 89–114, Chapman and Hall, London, 1996.
13. A. E. Raftery and S. M. Lewis, "Implementing MCMC," in *Markov Chain Monte Carlo in Practice*, W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, eds., pp. 115–130, Chapman and Hall, London, 1996.
14. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machine," *J. Chem. Phys.* **21**, pp. 1087–1091, 1953.
15. M. H. Kalos, *Monte Carlo Methods - Vol. 1: Basics*, John Wiley and Sons, New York, 1986.
16. A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences (with discussion)," *Statist. Sci.* **7**, pp. 457–511, 1992.
17. W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika* **57**, pp. 97–109, 1970.
18. Interactive Data Language, Research Systems, Inc., 2995 Wilderness Place, Boulder, CO 80301.
19. A. Gelman, G. O. Roberts, and W. R. Gilks, "Efficient Metropolis jumping rules," in *Bayesian Statistics 5*, J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, eds., Oxford University Press, 1996.
20. R. Giering, "Tangent linear and Adjoint Model Compiler," Tech. Rep. TAMC 4.7, Max-Planck-Institut für Meteorologie, 1997 (e-mail: giering@dkrz.de).
21. M. L. J. Rightley, R. J. Henninger, and K. M. Hanson, "Adjoint differentiation of hydrodynamic codes," in *CNLS Research Highlights*, Center for Nonlinear Studies, Los Alamos National Laboratory, April, 1998 (WWW: http://cnls.lanl.gov/Publications/highlights.html).