Proc. SPIE **4322**, pp. 456-467 (2001) LA-UR-01-1016
Medical Imaging: Image Processing, M. Sonka and K. M. Hanson, eds.

Markov Chain Monte Carlo posterior sampling with the Hamiltonian method

Kenneth M. Hanson

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

ABSTRACT

The Markov Chain Monte Carlo technique provides a means for drawing random samples from a target probability density function (pdf). MCMC allows one to assess the uncertainties in a Bayesian analysis described by a numerically calculated posterior distribution. This paper describes the Hamiltonian MCMC technique in which a momentum variable is introduced for each parameter of the target pdf. In analogy to a physical system, a Hamiltonian H is defined as a kinetic energy involving the momenta plus a potential energy φ, where φ is minus the logarithm of the target pdf. Hamiltonian dynamics allows one to move along trajectories of constant H, taking large jumps in the parameter space with relatively few evaluations of φ and its gradient. The Hamiltonian algorithm alternates between picking a new momentum vector and following such trajectories. The efficiency of the Hamiltonian method for multidimensional isotropic Gaussian pdfs is shown to remain constant at around 7% for up to several hundred dimensions. The Hamiltonian method handles correlations among the variables much better than the standard Metropolis algorithm. A new test, based on the gradient of φ, is proposed to measure the convergence of the MCMC sequence.

Keywords: Markov Chain Monte Carlo, Hamiltonian method, hybrid MCMC, Metropolis method, statistical efficiency, Bayesian analysis, posterior distribution, uncertainty estimation, convergence test

1. INTRODUCTION

In Bayesian analysis, the posterior probability distribution characterizes the uncertainty in the model parameters estimated from a given set of measurements. It has become evident that the Markov Chain Monte Carlo (MCMC) technique provides a straightforward way to explore the posterior, and hence characterize the uncertainty in parameters.[1–4] MCMC effectively generates a sequence of model realizations, randomly drawn from the posterior distribution.

Most Bayesian analyses make use of one of two standard MCMC algorithms. In the Gibbs approach, each variable of the target pdf is changed one at a time. The variable is chosen from the conditional probability for that variable, with all other variables held fixed. It is usually assumed that the conditional probability is known and easy to make random draws from. In common usage of the Metropolis algorithm, all the parameters are varied at once. The parameter vector is perturbed from the current sequence point by adding a trial step drawn randomly from a symmetric pdf. This proposed trial position is either accepted or rejected on the basis of the probability at the trial position relative to the current one. The Metropolis algorithm is often employed because of its simplicity. One discouraging property of the Metropolis algorithm is that its optimal efficiency for Gaussian distributions drops as $0.3/n$, where n is the number of variables,[5] which I have confirmed in previous work.[6] This loss of efficiency for high dimensional models is a severe disadvantage when the function evaluations are expensive.

In this paper I focus on a promising MCMC technique that I call the Hamiltonian method.[7,8] It is often referred to as simply the hybrid method because it alternates between Gibbs and Metropolis steps. However, that name does not distinguish it from any number of other algorithms that employ a combination of Gibbs and Metropolis steps. In 1980 Andersen[9] proposed a Monte Carlo approach to simulating a system of particles, such as in a gas. Each particle in the physical system is described in terms of a position and momentum. A Hamiltonian H is defined as a kinetic energy (sum over the square of each parameter's momentum divided by two times its fictitious mass) plus a potential energy φ. The goal is to draw random samples from the pdf proportional to exp(-H). The algorithm consists

Further author information – E-mail: kmh@lanl.gov; WWW: http://www.lanl.gov/home/kmh/; Telephone: +1 505 667 1402

of drawing the particles' momenta from a known distribution and then following their trajectories of constant H. Hamiltonian dynamics allows one to move along those trajectories using an algorithm such as the leapfrog technique.

Several years later, Duane et al.[7] put Andersen's simulation into an MCMC context; φ is taken to be minus the logarithm of the target pdf. Then, for each parameter in the problem, an auxiliary parameter is introduced, which represents the parameter's conjugate momentum variable. Duane et al. introduced a Metropolis test at the end of each such Hamiltonian trajectory to maintain detailed balance. After such a Metropolis step, Gibbs sampling is used to pick a new momentum vector, which is easy because the conditional pdf is an uncorrelated Gaussian in the momenta. The use of Hamiltonian dynamics facilitates large steps in the parameter space with only a few evaluations of φ and the gradient of φ. This algorithm and its refinements have been relied on to accomplish critical calculations in quantum field theory.[10] Note that the gradient of φ can often be done in a time comparable to the (forward) calculation of φ by applying adjoint differentiation to the computer code used to calculate φ.[11]

In this paper I show that when the target pdf is an isotropic Gaussian distribution, the efficiency of the Hamiltonian technique is nearly independent of the number of parameters. Only minor difficulties are encountered for more general Gaussian pdfs, for example, those with unequal variance for different parameters and with correlations among parameter uncertainties.

This paper is part of a broader effort to develop methods for conducting Bayesian inference using large simulation codes.[11] Thus, of particular interest are methods that can cope with large numbers of parameters, say hundreds or more, in a context where a function evaluation can take several hours or even days to perform on state-of-the-art computer systems. Our ultimate goal is to treat problems involving large simulations, for example, ocean[12] or atmospheric models, 3D tomographic reconstruction,[13] aerodynamics, and hydrodynamics. Thus, it is essential to reduce the number of steps taken by an MCMC algorithm needed to reach a specified degree of accuracy in estimating the uncertainties in these models.

1.1. Bayesian Inference with MCMC

The MCMC technique facilitates Bayesian inference by providing a means to generate a set of random samples from the posterior distribution. Given a set of N_k random parameter vectors $\{\mathbf{x}_k\}$ drawn from a pdf $q(\mathbf{x})$, one can easily estimate the expectation value of any function $f(\mathbf{x})$:

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

For example, the posterior mean estimate of the parameters $\bar{\mathbf{x}}$ is obtained using $f(\mathbf{x}) = \mathbf{x}$. With $f(\mathbf{x}) = (\mathbf{x} - \bar{\mathbf{x}})^2$, one obtains the variance. Other measures of parameter uncertainty can similarly be determined from the MCMC sequence.

The MCMC technique makes it feasible to perform many of the difficult technical calculations required in Bayesian analysis,[1,2] such as normalization of pdfs, marginalization, computation of expectation integrals, and model selection. The MCMC technique has opened up the possibility of applying Bayesian analysis to complex analysis problems.

2. TRADITIONAL MARKOV CHAIN MONTE CARLO ALGORITHMS

In MCMC the objective is to generate a sequence of parameter sets that mimic a specified target pdf $q(\mathbf{x})$ where $\mathbf{x}$ is a vector of parameters in the relevant parameter space. MCMC is useful 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, a complex simulation can provide for a specific $\mathbf{x}$ the value of $N\, q(\mathbf{x})$ where N is an unknown, but fixed, normalization constant. See Ref. 2 for an excellent introduction to MCMC and review of its use in statistics applications.

The process of exploring $q(\mathbf{x})$ is somewhat like feeling one's way in the dark; nothing is known until one tries to take a step and determines q at the new position. In that context, it would clearly help to know the gradient of $q(\mathbf{x})$ with respect to $\mathbf{x}$, because then one would at least know which way the terrain is sloping.

2.1. Metropolis Algorithm

One of the simplest algorithms used in MCMC calculations is due to Metropolis et al.[14] In this algorithm, one makes a trial perturbation 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.

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 much of current MCMC research and works remarkably well,[15,2] especially when one takes into account correlations among parameters[16,6]

2.2. Gibbs Algorithm

In Gibbs sampling, typically one parameter is varied at a time, holding all others fixed. The parameter is to be randomly drawn from the conditional pdf, the probability distribution of one parameter, given all other parameters; $q(x_i|\mathbf{x}_{-i})$, where $\mathbf{x}_{-i}$ is the full set of parameters excluding only the single component x_i. It is usually assumed that one can easily draw a random sample from this conditional pdf.

2.3. Statistical Efficiency of an MCMC Sequence

Equation (1) permits one to evaluate posterior expectation values of desired quantities from an MCMC sequence. A crucial issue is the degree of uncertainty in these estimates. The statistical efficiency of an MCMC sequence is defined as the ratio of the number of independent draws from the target pdf to the number of MCMC iterations required to achieve the same variance in an estimated quantity.

Suppose that we are given a sequence of N_k samples v_k drawn from a pdf of a scalar quantity v and that the samples are drawn from that pdf by a stationary process. The estimated value of v is given by the sample mean,

$$\hat{v} = \frac{1}{N_k}\sum_{k=1}^{N_k} v_k \quad . \tag{2}$$

If the v_k represent independent random draws from the underlying pdf, we know that the variance in the estimated value $\hat{v}$ is

$$\sigma_{\hat{v}}^2 = var(\hat{v}) = \frac{var(v)}{N_k} \quad , \tag{3}$$

where $var(v)$ is the variance of the pdf. If the variance of the estimate $\hat{v}$ is determined from numerous repetitions of the MCMC process, the statistical efficiency of the sequence generation procedure is then

$$\eta = \frac{var(v)}{N_k \sigma_{\hat{v}}^2} \quad . \tag{4}$$

The focus of this study is the estimation of the variance of the variables from MCMC sequences. Therefore, I will choose v to be the variance of a component of $\mathbf{x}$. The variance of the variance of a Gaussian distribution is twice its squared value. Thus, the efficiency of the MCMC algorithm for estimating the variance $var(\hat{v}_i)$ in x_i is

$$\eta_i = \frac{2(var(v_i))^2}{N_k\, var(\hat{v}_i)} \quad , \tag{5}$$

where $v_i = var(x_i)$ and $\hat{v}_i$ is its value estimated from an MCMC sequence of length N_k. The variance of $\hat{v}_i$ is estimated from many runs of the MCMC algorithm being tested.

While the above method for estimating the efficiency of an MCMC algorithm differs from that used in Ref. 6, it is derived from the same definition for statistical efficiency. The results of two methods should generally agree. However, because the efficiency of the Hamiltonian method is so high, it is difficult to use the autocorrelation function to measure it.

2.4. MCMC Issues

Two important practical issues in MCMC are convergence and burn in.[2] 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. The samples obtained during this "burn in" period must be discarded for subsequent analysis as it does not represent the pdf. One good way to determine convergence is to run multiple sequences starting each with disparate parameter values.[17] The sequences are taken to have converged when they coalesce into a common distribution. I will introduce a new convergence test in Sect. 3.2. For more detailed information about MCMC, the reader is referred to the excellent book edited by Gilks et al.[2]

3. HAMILTONIAN MCMC ALGORITHM

As mentioned in the Introduction, the Hamiltonian method is based on an analogy to physical systems. For each parameter x_i, a additional parameter p_i is introduced, which represents the parameter's associated momentum.[8] A Hamiltonian is constructed as a potential energy term, $\varphi = -\log(q(\mathbf{x}))$, plus a kinetic energy term:

$$H = \varphi(\mathbf{x}) + \sum_i \frac{p_i^2}{2m_i} \quad , \tag{6}$$

where $\varphi = -\log(q(\mathbf{x}))$ and m_i is a fictitious mass. The goal is to draw random samples from the new pdf that is proportional to $\exp(-H)$.

Each iteration of the algorithm starts with a Gibbs sampling to pick a new momentum vector from the uncorrelated Gaussian in the momenta corresponding to the second term in H. Then the trajectory in the $(\mathbf{x}, \mathbf{p})$ space that maintains a constant H value is followed using the leapfrog technique, which consists of the following three substeps:

$$p_i(t + \frac{\tau}{2}) = p_i(t) - \frac{\tau}{2} \left. \frac{\partial \varphi}{\partial x_i} \right|_{\mathbf{x}(t)} \quad , \tag{7}$$

$$x_i(t + \tau) = x_i(t) + \frac{\tau}{m_i} p_i(t + \frac{\tau}{2}) \quad , \tag{8}$$

$$p_i(t + \tau) = p_i(t + \frac{\tau}{2}) - \frac{\tau}{2} \left. \frac{\partial \varphi}{\partial x_i} \right|_{\mathbf{x}(t+\tau)} \quad , \tag{9}$$

where τ represents the time increment for the leapfrog step. The first and third updates in momentum are half time steps and have the effect of making the scheme accurate to second order in τ. After m leapfrog steps corresponding to a total trajectory time of $T = m\tau$, a Metropolis acceptance/rejection decision is made to guarantee that the sequence is in statistical equilibrium with q. This deterministic approach allows large steps in the parameter space to be taken with only a few evaluations of φ and the gradient of φ. The ability to take large steps is the essential feature of the Hamiltonian method that makes it attractive. The Metropolis test is required to maintain detailed balance in the MCMC sequence, that is, to guarantee that the probability of moving from the starting position to the final position exactly equals the reverse jump.

Let's count the number of function evaluations required by the leapfrog technique. The first and third leapfrog substeps require the evaluation of $\nabla\varphi$. However, the gradient in the third step is at the same location as the beginning of the next leapfrog step, so long as there is no Metropolis rejection. The Metropolis test requires the evaluation of φ at the end of the trajectory. In the ADICT scheme described in the following section, this evaluation of φ is typically done as part of the gradient calculation. The bottom line is that m leapfrog steps will typically require m φ evaluations and m evaluations of $\nabla\varphi$.

Figure 1 shows a typical sequence of H trajectories for a one-dimensional Gaussian distribution with unit variance. The vertical jumps correspond to the Gibbs sampling of momentum from the Gaussian pdf, $\exp(-p^2)$, for unit mass.

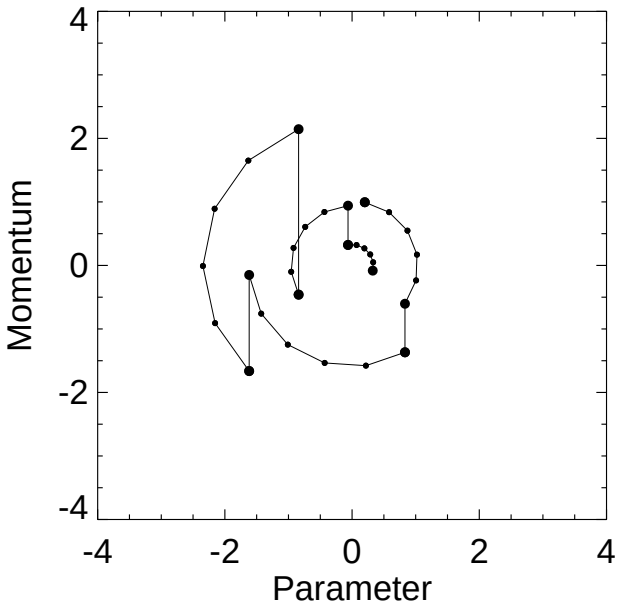

Figure 1. Example of a sequence of several trajectories in the momentum-parameter space for the Hamiltonian method for a 1D Gaussian distribution. The motion is always clockwise. Small dots are placed at end of each of the five leapfrog steps and large dots at the end of each Hamiltonian trajectory.

The circular arcs correspond to the trajectories of constant H followed in five leapfrog steps using $\tau = 0.4$, yielding a total trajectory time of $T = 5\tau = 2$.

One observes in Fig. 1 that it is possible to get into a resonance situation in which the Hamiltonian trajectories move around the circle by a rational fraction number of cycles. This type of behavior would clearly not represent random sampling and could lead to misleading results. Therefore, the length of the Hamiltonian trajectories must be randomized to realize an adequate random sampling of $q(\mathbf{x})$. Thus, for each Hamiltonian trajectory, T is randomly chosen from a uniform distribution from 0 to T_{max}. Once an MCMC sequence has been generated, the properties of $q(\mathbf{x})$ may be characterized by considering just the x_k samples. The momentum contributions to the extended pdf, $\exp(-H)$, are marginalized out because they are independent of the $\mathbf{x}$ dependence.

The choices for τ, T, and m_i are important for achieving the best efficiency. If τ is chosen to be too large, H will not be held constant enough, resulting in rejection of the new position by the Metropolis step. A good rule of thumb is that the length of the leapfrog steps should be kept smaller than the width of the **x**-**p** distribution. On the other hand, it is desirable for the length of the H trajectory, which is proportional to T, to be a large fraction of the width of the target pdf. As pointed out above, the values of T must be randomized to avoid resonance conditions. Thus, T_{max} should be chosen to produce H trajectories that are a few times the width of the target distribution. As with most other MCMC algorithms, one must explore the parameter space to optimize their efficiency for any particular application. It seems reasonable that one would like to pick the mass associated with each component that is about the same as the variance of the target distribution along that component in order to maintain circular trajectories in x-p space. However, as the parameters of the Hamiltonian algorithm have not been optimized in this study in any detail, the above advice should be taken as preliminary.

3.1. Adjoint Differentiation

The Hamiltonian method requires knowing 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.[11] The technique, which we have called Adjoint Differentiation In Code Technique (ADICT), essentially applies the chain rule for differentiation to the forward computer code. The result is an auxiliary code to compute the derivatives, which 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. Compilers are available to automatically create an adjoint code from a forward code, for example, the Tangent linear and Adjoint Model Compiler (TAMC) developed by Ralf Giering[18] for FORTRAN programs. TAMC has successfully been used to generate sensitivities for a 1D hydrodynamics code[19] and for an ocean-modeling code.[12]

It is often the case that the adjoint derivative calculation is done in the same amount of time as the forward calculation. For summarizing the results below, this equality will be assumed. The reader is cautioned that for more complicated forward calculations, the gradient calculation may take somewhat longer than the forward calculation, in which situations the efficiencies stated below would be overestimates of their true values.

3.2. Convergence Test

As mentioned in Sect. 2.4, guaranteeing convergence of a sequence is a major concern in MCMC. The following convergence test seems to be helpful for monitoring convergence in situations where $\nabla\varphi$ is available. The expression for the variance of $q(\mathbf{x})$ along any particular component x_i can be integrated by parts to obtain

$$var(x_i) = \int_{-\infty}^{\infty} (x_i - \bar{x}_i)^2 q(\mathbf{x})\, d\mathbf{x} = \frac{1}{3}\int_{-\infty}^{\infty} (x_i - \bar{x}_i)^3 q(\mathbf{x})\, \frac{\partial\varphi(\mathbf{x})}{\partial x_i}\, d\mathbf{x} + \left.\frac{1}{3}(x_i - \bar{x}_i)^3 q(\mathbf{x})\right|_{-\infty}^{\infty} \quad , \tag{10}$$

where $\bar{x}_i$ is the first moment of $q(\mathbf{x})$ in the x_i direction and $q(\mathbf{x})$ is assumed to be defined over the full infinite interval. The derivative is

$$\frac{\partial\varphi(\mathbf{x})}{\partial x_i} = -\frac{\partial \log(q(\mathbf{x}))}{\partial x_i} = -\frac{1}{q(\mathbf{x})}\frac{\partial q(\mathbf{x})}{\partial x_i} \; . \tag{11}$$

The second term on the right-hand side of Eq. (10) can often be argued to be zero because $q(\mathbf{x})$ usually drops off faster than $|x^3|$ as x approaches $\pm\infty$. Then the two integrals are equal. But these integrals may be evaluated using the MCMC samples drawn from $q(\mathbf{x})$ and checked to see if they indeed are equal, to within sampling uncertainties.

For a sequence of samples $\{\mathbf{x}^k\}$, allegedly drawn from $q(\mathbf{x})$, the proposed test consists of computing the ratio

$$R = \frac{\sum_k (x_i^k - \bar{x}_i)^3 \left.\frac{\partial\varphi}{\partial x_i}\right|_{\mathbf{x}^k}}{3\sum_k (x_i^k - \bar{x}_i^k)^2} \quad , \tag{12}$$

The mean value of $\bar{x}_i$ is determined in the obvious way from the x_i^k samples. The above argument implies that R should be unity. Of course, as these quantities are Monte Carlo estimates, they are subject to statistical fluctuations and R will fluctuate around unity, even for bona fide sample sets. Note that R generally need not be positive.

Experience indicates that R tends to be less than one when $q(\mathbf{x})$ is not adequately sampled. The plausibility of this observation can be understood by considering the behavior of the numerator for a Gaussian distribution. The derivative is proportional to $x_i - \bar{x}_i$ and so the numerator's sum is over $(x_i - \bar{x}_i)^4$. The value of R depends on the samples $\{\mathbf{x}^k\}$ from the MCMC sequence. If the sequence does a good job of spanning the width of the Gaussian target pdf, their distribution will make R come out to approximately unity. However, if the samples don't sufficiently reach to the edges of the Gaussian, then the denominator will tend to be larger than the numerator because of its slower dependence on $(x_i - \bar{x}_i)^2$.

One of the advantages of the above test is that it does not reply on $q(\mathbf{x})$ being Gaussian. However, the power of the test, in the sense of the uncertainty in R is small enough to determine when convergence is not reached, will depend on the functional behavior of $q(\mathbf{x})$. For example, if $q(x)$ is a 1D uniform distribution between two finite limits, the numerator will only get a nonzero contribution when the sample is at one of the limits and that contribution will be infinite.

4. RESULTS FOR MULTIDIMENSIONAL GAUSSIAN DISTRIBUTIONS

We now present some examples of the use of the Hamiltonian algorithm to generate MCMC sequences for multi-dimensional Gaussian distributions. All masses m_i are set to unity in these examples. Some attempt is made to choose optimal values for the parameters τ and T_{max}, but better values might be possible. The efficiencies stated below are derived from Eq. (5) using 1000 runs of the Hamiltonian algorithm for each set of conditions to compute the variance in the estimate of parameter variance made in each run. For this study, each run routinely employs 50

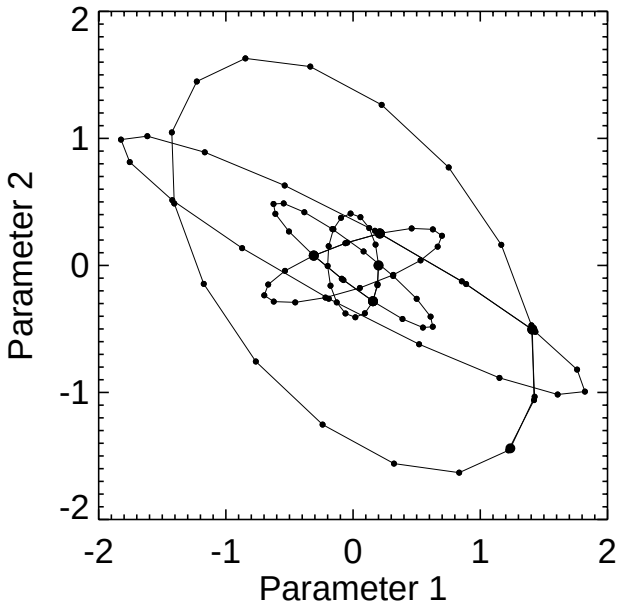

Figure 2. Five iterations of the Hamiltonian algorithm for a two-dimensional isotropic Gaussian pdf. The lengths of the trajectories are purposely chosen to be much longer than normally used to display their elliptical nature. A new elliptical trajectory results when the momentum vector is chosen by means of the Gibbs step.

iterations of the Hamiltonian algorithm. While this number may be far fewer than used in many applications, it is consistent with the stated intent to perform MCMC in situations where the function evaluations are expensive and, therefore, a limited accuracy in the estimated variance is acceptable. I have verified that the efficiencies quoted for 50 iterations remain valid for more iterations.

The examples presented here were obtained using the advanced image-processing language, IDL.[20] Care must be taken in using the random-number generation procedures, RANDOMU and RANDOMN; one must employ the same seed variable throughout a run.[6]

4.1. Two-dimensional Gaussian Distributions

Figure 2 displays five successive H trajectories for an isotropic, univariate two-dimensional Gaussian distribution. The total time for each trajectory is chosen to be abnormally large to display the shape of the H trajectories, $T = 7.2$. With $\tau = 0.4$, there are $m = 18$ leapfrog steps per H trajectory. Each H trajectory forms an ellipse, whose tilt and eccentricity is determined by the coordinates and momentum vector at its starting position. The momentum vector is randomly drawn from a 2D univariate Gaussian distribution.

Figure 3 shows the behavior of the Hamiltonian MCMC algorithm for a two-dimensional anisotropic Gaussian distribution with a standard deviation of 4 for x_1 and 1 for x_2. The total length of each H trajectory is randomly chosen from a uniform distribution between 0 and $T_{max} = 5$. For this example, the maximum value for τ is 0.4, resulting in up to 13 leapfrog steps per H trajectory. Thus, some trajectories are short and some are long. Because of the anisotropy in the target pdf, the H trajectories are no longer elliptical, but appear to be similar to Lissajous curves. For this example involving 15 H trajectories, all Metropolis tests are accepted. The MCMC sequence consists of the 15 points at the end of each H trajectory, shown in Fig. 3 as larger dots. A new momentum vector is chosen at these points, which starts the next H trajectory off in a new direction.

Table 1 summarizes the properties of the test statistic given by Eq. (12) seen in 1000 runs of the Hamiltonian MCMC algorithm. The target pdf is the same 2D distribution described in the previous paragraph. The H trajectories employ $\tau = 0.2$ and a rather small $T_{max} = 2$, chosen to limit the extent to which the algorithm samples the target pdf. It is observed that R_i provides a reasonably good indication of how well the MCMC algorithm has sampled the

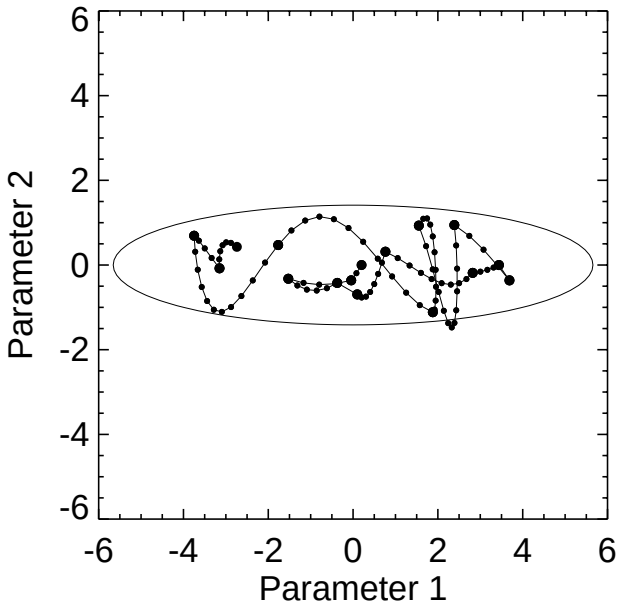

Figure 3. Hamiltonian trajectories for a two-dimensional anisotropic uncorrelated Gaussian pdf, demonstrating the ability of the Hamiltonian trajectories to readily transverse the length of the target pdf in just a few steps.

Table 1. Properties of the test statistic R given by Eq. (12) as a function of number of H trajectories for the 2D target distribution shown in Fig. 3. The degree of sampling of the Hamiltonian method is curtailed by choosing $T_{max} = 2$. The last column lists the average estimated variance from the 1000 runs used. The actual variances for the first and second components are 16 and 1, respectively.

Number of	Average R		Rms Deviation in R		Mean Variance	
Iterations	Comp. 1	Comp. 2	Comp. 1	Comp. 2	Comp. 1	Comp. 2
10	0.071	0.486	0.085	0.363	2.47	0.870
20	0.139	0.665	0.124	0.373	3.93	0.913
40	0.268	0.808	0.189	0.344	6.44	0.945
80	0.430	0.901	0.243	0.272	9.32	0.980
160	0.629	0.949	0.304	0.214	12.38	0.987
320	0.766	0.964	0.300	0.156	13.73	0.991
640	0.870	0.984	0.258	0.118	14.97	0.994

Table 2. Characteristics of Hamiltonian method for isotropic univariate Gaussian distributions as a function of their dimensionality. The Hamiltonian calculations involve 50 iterations with the parameters of the H trajectories set at: $T_{max} = 2$, $\tau = 0.4$, $m = 1$. Listed are the fraction of H iterations accepted, the efficiency per Hamiltonian iteration, and the efficiency per function evaluation. For comparison, the efficiency of the Metropolis algorithm is given.

Dimension	Acceptance	Efficiency/Iteration	Efficiency/Funct Eval	
n			Hamiltonian	Metropolis
4	0.984	0.447	0.075	0.075
16	0.968	0.417	0.070	0.019
64	0.931	0.394	0.066	0.0047
256	0.867	0.352	0.058	0.0012
1024	0.738	0.247	0.041	0.00029

target pdf. The behavior is different in the two components. The mean estimated variance for the second component is estimated to within 2% for 80 iterations, at which point $R = 0.90 \pm 0.27$. For 80 iterations, the estimated variance for first component is just a little over half of what it should be and $R = 0.43 \pm 0.24$. Even at 640 iterations, the variance of the first component is on the average 4% too low. The test statistic has almost reached unity; $R = 0.87 \pm 0.26$.

The statistical efficiency (5) is about 0.33/iteration for all of the above conditions.

4.2. Isotropic Multidimensional Gaussian Distributions

The main concern of this paper is whether the Hamiltonian method can avoid the decrease in efficiency for high dimensions, which has already been observed for the Metropolis method for isotropic Gaussian distributions.[6] Table 2 shows that the efficiency of the Hamiltonian method remains high, even for quite large dimensions. The Metropolis efficiency listed is derived from the formula $\eta = 0.3/n$, which is appropriate for isotropic Gaussian target pdfs.[6] Per function evaluation, the efficiency of the Hamiltonian method remains constant at about 7% up to several hundred dimensions. The efficiency of the Metropolis algorithm starts at 7.5% at four dimensions and drops precipitously for higher dimensions. The Hamiltonian method provides clearly superior performance at high dimensions.

The average value of the variance for these runs is around 0.96, quite close to the actual value of 1.00.

4.3. Correlated Multidimensional Gaussians

To demonstrate another property of the Hamiltonian MCMC algorithm, I use an example presented in Ref. 6. The target distribution is a 16-dimensional Gaussian distribution with a high degree of correlation. The form of the target pdf is motivated by a type of regularization typically used to solve ill-posed problems, that of a smoothness prior or regularizer. See Ref. 6 for the derivation of the covariance matrix, a 6×6 piece of which is:

4.97	3.98	2.50	1.24	0.42	-0.02
3.98	4.97	3.98	2.50	1.24	0.42
2.50	3.98	4.97	3.98	2.50	1.24
1.24	2.50	3.98	4.97	3.98	2.50
0.42	1.24	2.50	3.98	4.97	3.98
-0.02	0.42	1.24	2.50	3.98	4.97

This covariance matrix clearly indicates a high degree of positive correlation over several neighboring elements, which was shown to lead to inefficiencies for the standard Metropolis-MCMC algorithm. Using an isotropic Gaussian step distribution, the best achievable efficiency for this problem was 0.11%. The efficiency was increased to 1.6% by tailoring the trial step distribution to approximately match the covariance matrix estimated from a preliminary MCMC run.

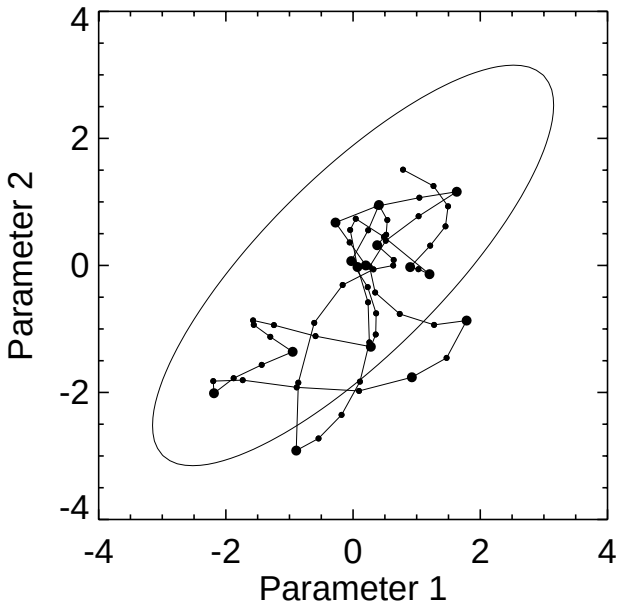

Figure 4. A consecutive set of 15 Hamiltonian trajectories in a 2D subspace from a correlated 16-dimensional Gaussian pdf. The contour shown is at the two-standard-deviation level for the marginalized distribution.

Table 3. Characteristics of Hamiltonian method for correlated multidimensional Gaussian distribution as a function of their dimensionality. The Hamiltonian calculations employ 50 iterations with the parameters of the H trajectories: $T_{max} = 8$, $\tau = 0.4$, $m = 1$. Listed are the fraction of H iterations accepted, the efficiency per Hamiltonian iteration, and the efficiency per function evaluation.

Dimension, n	Acceptance	Efficiency/Iteration	Efficiency/Funct Eval
16	0.919	0.453	0.022
64	0.831	0.391	0.019
128	0.765	0.352	0.017

The Hamiltonian calculations for this problem employ 50 iterations with the parameters of the H trajectories set to $T_{max} = 8$, $\tau = 0.4$, $m = 1$. A larger value for T_{max} is used than in earlier examples because of the larger variance of the target pdf. Figure 4 shows a typical set of 15 iterations for this problem. The larger dots, representing the actual sample points at the ends of each H trajectory, do a pretty good job of sampling this subspace.

Table 3 lists the efficiencies observed in 1000 runs of the Hamiltonian algorithm for various dimensionalities. Recall that the best efficiency obtained with the Metropolis algorithm was 1.6%, and that was only after adapting the trial step distribution to the problem. The efficiency of the simple Metropolis was 0.11%. The Hamiltonian algorithm achieves better efficiency at 16 dimensions without any folderol. What is more, the efficiency of the Hamiltonian method does not drop much with increasing dimensionality. The best achievable efficiency for the Metropolis algorithm for 128 dimensions, after adapting it to the shape of the target pdf, would be 0.23%, about 7 times worse than the Hamiltonian algorithm achieves!

The estimated variance (diagonal elements of the covariance matrix) for these runs is about 4.75, reasonably close to the actual value of 4.97.

5. DISCUSSION

The above results demonstrate the superiority of the Hamiltonian method to the Metropolis algorithm for generating a sequence of random samples from a calculated target pdf, especially for more than 6 dimensions. The caveat is that one must be able to calculate not only φ, but also $\nabla\varphi$. The efficiencies quoted are based on the assumption that one can calculate the gradient as quickly as φ itself, which is possible for many calculations using adjoint differentiation (ADICT). The Hamiltonian method not only maintains its high efficiency for high dimensions, but it also handles anisotropic and correlated distributions in a robust manner.

There are several ways in which the Hamiltonian method might be improved. The crux of the method is the use of the Hamiltonian trajectories. Because this aspect of the algorithm is deterministic, it seems reasonable to try to improve the ability of the calculated trajectory to keep H constant, thereby avoiding Metropolis rejection, or, in fact, the need for the Metropolis test. The usefulness of the algorithm would be improved if one could adaptively adjust the leapfrog step size τ to the local properties of the target pdf in order to maintain the accuracy in H. The difficulty is to balance the H accuracy against any extra calculation needed. As with it any MCMC method, it is also possible to improve the performance of the Hamiltonian method for correlated and anisotropic pdfs through the usual means of adapting the algorithm to include estimates of the covariance structure of the target pdf.[2]

It is desirable to investigate the performance of the Hamiltonian method for highly anisotropic distributions. Because of the deterministic aspect of the H trajectories, it can be imagined that one might find ways to cope with anisotropies by adapting the leapfrog method to the varying characteristics of the Hamiltonian dynamics along the trajectory.

The unadorned leapfrog method relies solely on the gradient of φ. In real-world calculations, the gradient might not be calculated very accurately. One area of research that could prove useful is to develop methods to handle inaccurate calculations of $\nabla\varphi$. For example, corrections to the leapfrog H trajectories could be made on the basis of calculated values of φ. These corrections might even be useful when the gradients are accurately calculated to overcome the shortcomings of the leapfrog method for larger step sizes, which are desirable for maintaining high efficiency of the Hamiltonian algorithm.

My experiments with momentum persistence,[8,10] in which one nudges the momentum instead of completely changing it with a random draw from the momentum distribution, have not provided a compelling reason to use it.

ACKNOWLEDGEMENTS

I would like to thank the following for their help in understanding MCMC and its potential usefulness: Greg Cunningham, Frank Alexander, David Haynor, David Higdon, Julian Besag, Malvin Kalos, John Skilling, James Gubernatis, and Richard Silver. This work was supported by the United States Department of Energy under contract W-7405-ENG-36.

REFERENCES

1. J. Besag, P. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Stat. Sci.* **10**, pp. 3–66, 1995.
2. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.
3. M.-H. Chen, Q.-M. Shao, and J. G. Ibrahim, *Monte Carlo Methods in Bayesian Computation*, Springer, New York, 2000.
4. C. P. Robert and G. Casella, *Monte Carlo Statistical Method*, Springer, New York, 1999.
5. 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.
6. K. M. Hanson and G. S. Cunningham, "Posterior sampling with improved efficiency," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE* **3338**, pp. 371–382, 1998.
7. S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Phys. Lett. B* **195**, pp. 216–222, 1987.
8. R. M. Neal, *Bayesian Learning for Neural Networks*, Springer, New York, 1996.
9. H. C. Andersen, "Molecular dynamics simulations at constant pressure and/or temperature," *J. Chem. Phys.* **72**, pp. 2384–2393, 1980.

10. A. M. Horowitz, "A generalized guided Monte Carlo algorithm," *Phys. Lett. B* **268**, pp. 247–252, 1991.
11. K. M. Hanson, G. S. Cunningham, and S. S. Saquib, "Inversion based on computational simulations," in *Maximum Entropy and Bayesian Methods*, G. E. et al., ed., pp. 121–135, Kluwer Academic, Dordrecht, 1998 (to be published).
12. G. Burgers, R. Giering, and M. Fischer, "Construction of the adjoint of the HOPE OGCM," *Ann. Geophysicae.* **C14**, p. 390, 1996.
13. 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.
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. 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.
17. A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences (with discussion)," *Statist. Sci.* **7**, pp. 457–511, 1992.
18. R. Giering, "Tangent linear and Adjoint Model Compiler," Tech. Rep. TAMC 4.7, Max-Planck-Institut für Meteorologie, 1997 (e-mail: giering@dkrz.de).
19. 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).
20. Interactive Data Language, Research Systems, Inc., 2995 Wilderness Place, Boulder, CO 80301.