Open Access

Markov Chain Monte Carlo Calculations Allowing Parallel Processing Using a Variant of the Metropolis Algorithm

Guthrie Miller*

Los Alamos National Laboratory, Los Alamos, New Mexico 87544, USA

Abstract: A variant of the Metropolis algorithm is proposed that allows parallel processing. Rather than generating a single candidate point, as in the Metropolis algorithm, for each chain iteration a number of candidates are generated. Energy calculations for each of these candidates can be carried out in parallel. This algorithm would be advantageous in fitting model parameters to data in a Bayesian context, where the forward model calculations (the analog of the energy calculations), are time consuming.

Keywords: Metropolis algorithm, Parallel processing, Markov Chain Monte Carlo (MCMC), Bayesian statistics, Data analysis.

1. INTRODUCTION

Computer processor speed is increased primarily by shrinking the size of the circuitry, and modern processors are pushing physical limits on size and heat dissipation. Increases in computer processing power in the future will thus more and more come about through having multiprocessing systems. To take advantage of such systems for a single problem requires, instead of the traditional sequential algorithm a parallel algorithm, pieces of which can be executed in parallel on many different processors.

Markov Chain Monte Carlo (MCMC) originated with the 1953 paper [1] by Nicolas Metropolis, Adriana Rosenbluth, Marshall Rosenbluth, Agusta Teller, and Edward Teller. This paper surely stands in the first rank of scientific papers from the twentieth century and has had a profound impact on the field of scientific computation [(see, for example, Refs. 2 and 3)].

The problem in statistical mechanics that was being addressed required the evaluation of an integral over a high dimensional configuration space, representing the positions of N particles, of the function exp$(-E/kT)$, where E is the potential energy of the system as a function of the 2N particle coordinates (a two-dimensional space was assumed), the temperature T is a constant, and k is Boltzmann's constant. The authors state:

"It is evidently impractical to carry out a hundred-dimension integral by the usual numerical methods, so we resort to the Monte Carlo method. The Monte Carlo method for many-dimensional integrals consists simply in integrating over a random array of points instead of over a regular array of points. Thus the most naïve method of carrying out the integration would be to put each of the N particles at a random position in the square (this defines a random position in the 2N-dimensional configuration space), then calculate the energy of the system…, and give this configuration a weight of exp$(-E/kT)$. This method, however, is not practical…, since with high probability we choose a configuration where exp$(-E/kT)$ is very small; hence a configuration of very low weight. So the method we employ is actually a modified Monte Carlo scheme, where instead of choosing configuration randomly, and weighting them with exp$(-E/kT)$, we choose configurations with a probability exp$(-E/kT)$ and weight them evenly."

The method of generating configurations with probability exp$(-E/kT)$ was to draw samples from a cleverly constructed Markov Chain, which was run for a large number of iterations. An introduction to the Metropolis method and its applications in statistical mechanics is given in Ref. [4].

The method proposed here is closely related to the method proposed by Barker [5], and for the special case of a single candidate the present method is the same as Barker's method.

In the use of MCMC for statistical modeling the work of Hastings [6] is usually cited along with that of Metropolis. Hastings discusses both the Metropolis and Barker methods. Only a single candidate is considered.

For Bayesian statistical applications (e.g. fitting model parameters to data), the "energy" function is the negative of the log of the combined likelihood function (for independent data, the sum of the log likelihoods for all the individual data points-a simple heuristic discussion of the likelihood function is given in Appendix A). The prior is usually incorporated into the parameterization (so that the prior becomes effectively uniform). Two times the energy is an effective chi-square function indicating how well the model fits the data. In these kinds of applications the evaluation of the energy (likelihood) function often involves time-consuming numerical solution of the forward model (from parameters to data), and most of the computation time is involved in this activity.

In the present work, following closely the approach of the original Metropolis *et al*. paper, a variant of Markov-Chain algorithm that allows parallel processing is proposed. Being able to carry out these forward model calculations in

*Address correspondence to this author at the Group RP-2: Health Physics Measurements, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA; Tel: 505-665-6064; Fax: 505-665-6071;
E-mail: guthrie@lanl.gov

1876-3898/10 2010 Bentham Open

parallel allows, in the same total computation time, *P* times more of them, where *P* is the number of parallel threads used.

A natural use of parallel processing with the Metropolis algorithm is to run multiple independent chains, in parallel, and to appropriately average the results at the end of the runs [7]. This is not completely straightforward, however, particularly when *P* is large, because of the difficultly of estimating "initialization bias" caused by the influence of the chain initialization, which is taken care of by deleting an initial "burn-in" fraction of the chain. See also Ref. [8] for a review of MCMC parallelization using multiple chains.

Parallel processing can also be incorporated into the likelihood function calculation for single-chain calculations. One general method of doing this is "pre-fetching", [9] where all the possible required likelihood function calculations, extrapolated several iterations into the future, are done in advance using parallel processing, even though not all of them will actually be used--clearly an inefficient procedure.

2. THE PROBLEM

There is a possibly multidimensional parameter space with parameter point denoted by Θ. Each of the dimensions *j* of Θ has domain 0 to 1. The problem of interest may be stated in terms of multidimensional integration to evaluate expectation values of a given probability distribution. The goal is to calculate expectation value integrals of the form

$$E\left[F(\Theta)\right] = \frac{\int d\Theta e^{-E(\Theta)} F(\Theta)}{\int d\Theta e^{-E(\Theta)}} \quad , \tag{1}$$

where *F* is an arbitrary function of Θ, the probability distribution $P(\Theta)$ is proportional to $e^{-E(\Theta)}$, and the given "energy" function $E(\Theta)$ is bounded from below.

3. THE METHOD

The method is Markov Chain Monte Carlo integration, where an algorithm produces a Markov Chain (a sequence of parameter points Θ_t) with stationary state $P(\Theta) \propto e^{-E(\Theta)}$. The expectation value integral given by Eq. (1) is then approximated by

$$E\left[F(\Theta)\right] \cong \frac{1}{T} \sum_{t=0}^{T} F(\Theta_t) \quad , \tag{2}$$

for large $T \rightarrow \infty$, where *t* enumerates the chain iteration.

4. THE ALGORITHM

The Markov Chain algorithm is a rule telling the computer how to select the next point Θ' in parameter space, given that the chain is at a current point Θ. The rule is the following. First *N* candidates for the new point, labeled by *i = 1, N,* are generated from a conditional probability distribution $q(\Theta' \mid \Theta)$ (read as "the probability distribution of Θ' given, or conditioned on, Θ"). This "candidate" probability distribution is most often chosen to be a random walk ($\Theta' = \Theta + \Delta(x - 1/2)$, where x is a random number uniformly distributed between 0 and 1, and Δ is a fixed parameter, for all or some subset of the dimensions of Θ'); however, the single requirement on this distribution is that it allow every point in parameter space to be eventually reached starting from every other point, possibly after many steps. An additional point *i = 0* represents the unchanged current point. The chain is moved to the next point *i = 0, N* with discrete probability $P_i \propto G_i$, where

$$G_i \equiv \frac{e^{-E(\Theta_i')}}{q(\Theta_i' \mid \Theta)} \quad . \tag{3}$$

For *i = 0* in the above, Θ_i' is to be replaced by Θ.

A single chain is run; however, at each iteration, multiple candidates for the next position of the chain are generated. Only a single one of these candidates will be selected as the next position of the chain.

5. STATIONARY STATE

The arguments follow closely those of Metropolis *et al.* [1]. Consider a very large ensemble *N* of chains that have been run a long time from different starting points. The number of chains in volume element $d\Theta$ is denoted by $n(\Theta)$. By the rule given by Eq. (3), the number of chains that move from Θ to Θ' in one time step is proportional to $n(\Theta)e^{-E(\Theta')}$ while the corresponding number that move from Θ' to Θ is proportional to $n(\Theta')e^{-E(\Theta)}$. If

$$\frac{n(\Theta)}{n(\Theta')} < \frac{e^{-E(\Theta)}}{e^{-E(\Theta')}} \quad , \tag{4}$$

there will be a net flow from Θ' to Θ, which will tend to remove this inequality. The ensemble will approach a stationary state with

$$n(\Theta) \propto e^{-E(\Theta)} \quad . \tag{5}$$

More discussion is warranted. Consider, for simplicity, a discrete parameter space. In general, the number of chains in state *k* satisfies the equation

$$\frac{dn_k}{dt} = -n_k \sum_l p_{kl} + \sum_l n_l p_{lk} = -n_k + \sum_l n_l p_{lk} \quad , \tag{6}$$

where p_{kl} is the probability that the chain will transition from state *k* to *l*, and the first term on the right-hand side of Eq. (6) represents chain transitions where the chain moves *from* state *k* and the second term transitions where the chain moves *to* state *k*. If there exists a configuration n_k that satisfies the detailed balance condition

$$n_k p_{kl} = n_l p_{lk} \quad , \tag{7}$$

this configuration is a steady state from Eq. (6), however a steady state from Eq. (6) does not necessarily satisfy detailed balance. The additional fact that needs to be emphasized is that for an irreducible, ergodic chain (where every point can be reached from every other point after a finite number of steps), there is a single, unique steady state independent of chain starting position [10], so a steady state obtained by detailed balance is *the* steady state.

One needs to know how long the chain need be run for convergence to the steady state. This can be assessed in practice by running two independent chains with different starting points and different random number seeds and comparing the results. The comparison of the desired distribution (of $F(\Theta)$) calculated from the two chains directly gives the degree of convergence. As usual, to minimize the effects of transient states ("initialization bias"), some initial fraction (the "burn-in" fraction) of each chain is discarded. Convergence is obtained by increasing the total number of iterations, with the same fraction of initial iterations discarded, until the two independent chains give an acceptable agreement for the distributions of the quantities of interest.

6. NUMERICAL EXAMPLE ONE

Consider a purely artificial one-dimensional problem. The motivation for this example is to have the actual distribution that is being approximated using MCMC be something that is easy to describe. The energy function has an infinite rectangular well of width 0.4 centered on the point $\Theta = 0.75$. A candidate point is obtained by generating a random number x uniformly distributed between 0 and 1 and letting $\Theta' = x^2$. Because $d\Theta' = 2xdx$ and dx is numerically equal to the probability dP that the random number will be in this interval,

$$\frac{dP}{d\Theta'} = q(\Theta' \mid \Theta) = \frac{1}{2\sqrt{\Theta'}} \quad . \tag{8}$$

For each chain iteration, 950 candidate points *i* = 1, 950 are generated. The current point Θ is included as *i = 0*, and the probabilities of moving the chain to the points *i* = 0 to 950 are set up to be proportional to (for *i = 0* let $\Theta_i' = \Theta$)

$$\sqrt{\Theta_i'}e^{-E(\Theta_i')} \quad . \tag{9}$$

An integer *i* is generated from this discrete probability distribution (by partitioning the interval from 0 to 1 with the *i* = 0, 950 normalized probabilities, generating a uniform random number from 0 to 1, and selecting the *i* of the interval containing the random number). The chain is then moved to the point Θ_i' . This process is continued for a large number of steps.

The result is shown in Fig. (**1**).

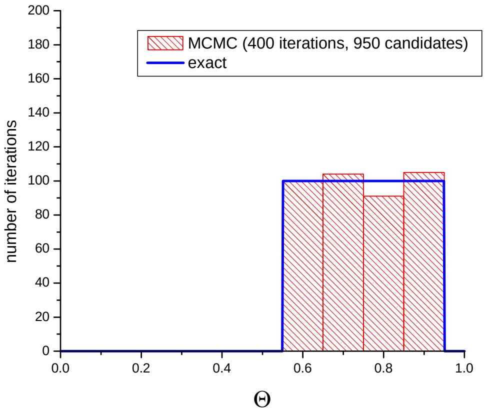

Fig. (1). MCMC run for the example problem with an infinite height rectangular energy well. Each iteration involved 950 candidate points. The energies of each of these points could be calculated in parallel. In the entire run 380000 candidates were calculated.

Because the correct result is known, convergence has to do with the departure of the solution from the correct result, which is accounted for by the Poisson statistics (±10 iterations) of the number of events in each bin shown in Fig. (**1**).

In contrast, Fig. (**2**) shows the same MCMC run of 400 iterations, except that only a single candidate is generated.

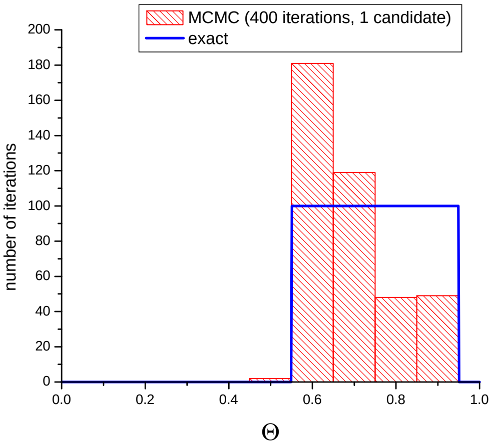

Fig. (2). MCMC run for the example problem with an infinite height rectangular energy well as shown in Fig. **1** except that in this case only a single candidate is generated. In the entire run 400 candidates were calculated.

The comparison of Fig. (**1**) and Fig. (**2**) clearly shows the advantage of this method. If 950 processors were available (and other overheads were small) both calculations would take the same time to complete, and the Fig. (**1**). result is certainly preferable.

7. CHOOSING THE CANDIDATE DISTRIBUTION

Only uniform, random-walk candidate distributions are considered in this section. The question is: "How should one choose the Δ's?"

Assume initially that all the coordinates of Θ are varied (coordinate parameters for all dimensions).

Consider the total volume of parameter space explored by the candidate distribution

$$V = \prod_j \Delta_j \quad ,$$

(*j* runs over all the dimensions of Θ). Initially, when the chain begins at an arbitrary point far from a minimum energy point, the volume *V* should be the entire volume of parameter space. However, the chain then tends to find a lower energy point and remain there a long time without movement, because the candidate points are so widely spaced in parameter space. With this type of behavior in mind, the algorithm can have an initial "adaptive phase" where, if the chain has remained stationary for a period of time (say for n_{Same} iterations), the total volume *V* searched is decreased by a factor of $n_{Same} \times N$ (by proportionally decreasing the Δ's), where *N* is the number of candidate points. The rationale for this is that if the chain has remained stationary for n_{Same} iterations, the minimum energy point is likely to be located inside a volume $v = V/(n_{Same} \times N)$ surrounding the stationary point.

Thus there is no reason to continue searching outside this volume, and the volume v can become the total volume V for the next iterations. In practice a safety factor F (e.g. 3) is introduced so that the new volume $V = F\ v$.

The other parameters of this adaptive algorithm that are specified in advance are, n_{Same} (e.g. 2), the number of consecutive times the chain remains stationary before the search volume is decreased, $n_{NotSame}$ (e.g. 5), the number of consecutive times the chain moves before the adaptive phase ends, and a minimum value of Δ for each parameter. The minimum Δ (e.g. a factor of 1.2) is based on the quality of the data. The adaptive phase ends when the chain has consecutively moved for all the previous $n_{NotSame}$ iterations. If the search volume is no longer being decreased because the Δ's have reached their minimum values, it becomes a matter of waiting for a run of $n_{NotSame}$ iterations with consecutive moves. The Δ's obtained with this adaptive algorithm result in a low *rejection rate* (the rate at which the candidate is rejected; quoting Hastings [6]: "high rejections rates are to be avoided"). However, going along with a low rejection rate, the candidate distribution is usually exploring only a minute fraction of the entire parameter space, which can be a problem if there are widely spaced additional energy minima. To handle this situation one needs to move only single parameters or subgroups of parameters at each iteration rather than all the parameters at once, which allows sampling of larger volumes with acceptably low rejection rates. Also, in this case one can alternate between random walk sampling (with some Δ's) and sampling of the entire subspace, with specified fractions of time spent on each.

A question in practice is how to choose the Δ's when different sized groups of parameters are used. To analyze this situation (which is done very roughly here), assume that the posterior probability is given by a multivariate log normal, with the logarithms used as parameters (alternatively, imagine a multivariate normal distribution). Assume that the parameters j are grouped into groups G_k. The number of candidate points is N , and we desire an approximately equal number of these points N_1 to land in the support region of the posterior, of size σ_j. Then,

$$N_1 \propto N \prod_{j \ni G_k} \frac{\sigma_j}{\Delta_j} \ ,$$

so that

$$\prod_{j \ni G_k} \frac{\Delta_j}{\sigma_j} \cong const \ .$$

For example, consider a problem where there are 8 parameters that are naturally grouped together, because of strong correlations between them. Assume another group of 4 parameters that are relatively independent of the first group. Assume that the width of the candidate distributions for the 8-parameter group is a factor of 1.5, while the width of the posterior (for all parameters) is estimated to be a factor of 1.2 because of the strength of the data. Then the width of the candidate distribution for the 4-parameter group needs to be

$$1.2^{\left(\frac{\log(1.5)}{\log(1.2)}\right)^{8/4}} \cong 2.5 \ ,$$

showing how the "size" of the space sampled by the candidate distribution can be compared across subsystems of different dimensionality.

This new algorithm also allows stratified sampling (see Appendix), which would seem to be advantageous in some situations, although the author has not yet been able to demonstrate this.

8. NUMERICAL EXAMPLE TWO

This example involves actual bioassay data (urine and whole body counts) from an individual who was involved in the Goiania radiation accident. Details of the data and the problem background are given in Ref. [11], which is freely available online. This example is also calculated using WinBugs [12] in the soon-to-be-available National Council on Radiation Protection (NCRP) report on uncertainties in internal dosimetry [13].

The passage of ingested, highly soluble, cesium through the body is assumed to be described by the compartmental model recommended by the International Commission on Radiation Protection (ICRP) shown in Fig. (**3**).

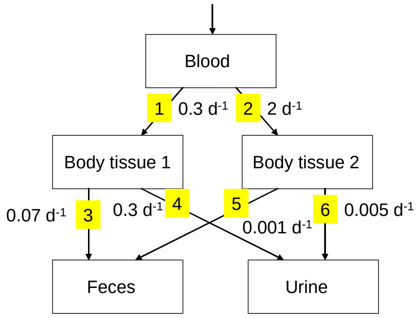

Fig. (3). Biokinetic model describing ingestion of highly soluble cesium recommended by the ICRP. There are six rate coefficients labeled 1 through 6. The parameters Θ_j for j = 1, 6 are the scaled natural logarithms of these rate coefficients. The log-space uniform prior on each rate coefficient was centered on 0.1 d^{-1} with a range of variation ÷1000 to ×1000 (a factor of 1000000 total variation).

The values of the rate coefficients indicated in Fig. (**3**) are values recommended by the ICRP for an adult. Somewhat different central values are obtained from the data in this case (case MPA in Ref. 10), which consisted of 15 24-hr urine excretion measurements in the time interval from 14 to 69 days post intake (assumed to be a single acute intake occurring on 20-Sep-1987) and 6 measurements of whole body content in the time interval from 66 to 418 days post intake. These data and other details are given in Ref. [10].

The energy function depends on the intake amount I as well as the biokinetic parameters Θ. This is replaced by a function of only the biokinetic parameters by performing the integration over intake amount numerically,

$$E(\Theta) = E_0 - \log\left(\int e^{E_0} L(\Theta, I) P(I) dI\right) = E_0 - \log\left(\int e^{-(E(\Theta,I)-E_0)} P(I) dI\right) \ ,$$

where E_0 is a constant that might be needed to avoid having the integration for all the N candidates numerically underflow to zero, $L(\Theta, I)$ is the combined likelihood function for all the data (the product of the likelihood functions for all individual data points), and *P(I)* is the prior probability of intake amount (a broad lognormal assumed). Because the biokinetic models are linear in intake amount *I*, the integration over *I* is accomplished by first capturing all the *N* candidate biokinetic models in interpolation tables (giving, for a range of times, the bioassay quantities for unit intake) and then straightforwardly and relatively quickly performing the integration over *I* numerically. By far most of the computer time is spent on the solution of the differential equations of the forward model.

The Markov Chain Monte Carlo analysis used N = 950 candidate distributions, calculated in parallel using 8 threads, which required about 20 seconds of computer time per iteration (3 GHz processor speed). The adaptive method discussed in Sec. 6, starting with a search of the entire space was used. In each iteration, all 6 parameters were varied. A run of 100 iterations was made, with the initial 25 iterations discarded as being either part of the adaptive phase or being influenced by the chain initial conditions. A scatter plot of parameter 5 versus parameter 6 for the remaining 75 iterations is shown in Fig. (**4**).

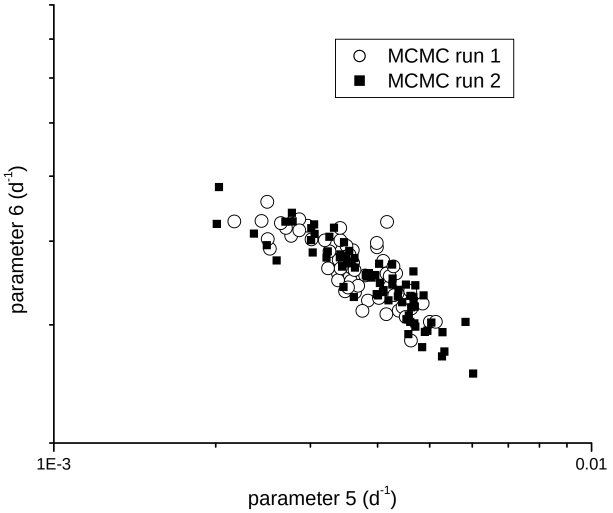

Fig. (4). Scatter plot of parameter 5 versus parameter 6 showing the uncertainties of these parameters including correlation (central values of both parameters about 0.004 d^{-1}). This plot should be compared with Fig. (**6**) of Ref. [10], and one finds them to be essentially identical.

Two MCMC runs are shown in Fig. (**4**) to demonstrate convergence. In one, the prior is centered on 0.1 d^{-1} as described in Fig. (**4**). In the other, the prior is centered on 0.3 d^{-1}, and a different random number seed is used.

One finds Fig. (**4**) to be essentially identical with Fig. (**6**) of Ref. [4], which used a totally different method of calculation. The Markov Chain method is thus of comparable feasibility for this type of problem. It is advantageous in its conceptual simplicity and that the "tape" of chain parameters versus iteration simply and directly returns the parameter uncertainties and correlation.

DISCUSSION AND CONCLUSIONS

In the Metropolis algorithm, a single candidate is generated. If the energy is decreased, the chain is moved. If the energy is increased, the chain is moved with probability equal to the ratio of the smaller divided by the larger of the two exponentials. For a single candidate, the algorithm described here is the same as the Barker algorithm, and it does not necessarily move the chain, even if the candidate has a lower energy. The chain moves probabilistically in all cases. The Barker algorithm is thus distinct from the Metropolis algorithm; however, it is found to perform very similarly.

The conclusions of this paper are: 1) a new MCMC algorithm has been derived, and it's advantage is that it allows more than one candidate to be considered, with the energy functions for these candidates evaluated simultaneously, using parallel processing; 2) the algorithm has been demonstrated for a statistical data fitting example that has recently been considered in two other publications, with good agreement obtained; 3) an adaptive-phase algorithm for choosing the width of the candidate distributions is proposed; 4) with multiple candidates, it would be of interest to investigate other sampling schemes besides random sampling, for example, stratified sampling.

ACKNOWLEDGEMENTS

The author would like to acknowledge assistance with multithread FORTRAN programming kindly provided by Peter Jacobson of Absoft Corporation.

The author is also grateful for the helpful suggestions and literature references provided by the reviewers that greatly improved this paper.

APPENDIX A-HEURISTIC DISCUSSION OF THE LIKELIHOOD FUNCTION

Consider a simple one-dimensional problem where one is attempting to estimate a single parameter θ from a single measurement with value *y*. The forward model calculation gives the predicted value of the measurement $f(\theta)$ if the true value of the parameter were known to be θ. If the true value of the parameter were θ, and the measurement was repeated a large number of times, the measurement values would have some distribution that could be normalized to form a probability distribution $P(y|\theta)dy$, (read as the probability that *y* will be found to be in the interval *dy* given θ). The problem at hand is to statistically interpret a single measurement that yields a single measurement value *y*. By the elementary rules of conditional probability (Bayes theorem),

$$P(\theta \mid y)d\theta \propto P(y \mid \theta)P(\theta)d\theta \ .$$

In words, the desired probability that the true value of the parameter is in the interval $d\theta$, given the measurement result that has been obtained, is proportional to $P(y|\theta)$ times the prior $P(\theta)$, times $d\theta$. This relationship gives an unnormalized probability; but it is easily converted to a proper probability distribution by normalization

$$P(\theta \mid y) = \frac{P(y \mid \theta)P(\theta)}{\int P(y \mid \theta)P(\theta)d\theta} \ .$$

The likelihood function is defined up to a multiplicative constant by the relationship

$$L(\theta) \propto P(y \mid \theta) \ .$$

In many cases of interest, where the statistical variations of the measurement are caused by variations of a large number of factors (e.g. urine excretion), the likelihood function is approximately log-normal

$$L(\theta) \cong C \exp\left(-\frac{1}{2}\left(\frac{\ln(y)-\ln(f(\theta))}{\sigma}\right)^2 \right) ,$$

where σ is a constant representing measurement variability and C is an arbitrary constant factor. In measurements of urine excretion involving counting (for example counting radioactive decays when measuring the excretion of a radioactive material), the log normal approximation is only valid when the counting statistics variations are relatively small. A more general model of the likelihood function (the Poisson mixture model) is discussed in Ref. [14].

There is an effective chi-square interpretation of the log likelihood function, but it requires that the maximum of the log likelihood function for a single data point is 0 (the constant C is one). In that case, for the lognormal likelihood function,

$$-2\ln\left(L(\theta)\right) = \chi^2 = \left(\frac{\ln(y)-\ln(f(\theta))}{\sigma}\right)^2 .$$

That is, -2 times the log of the likelihood function (the energy function) is a measure of how much the data departs from the predicted value.

Note the power of Bayes theorem to relate statistical variation of the measurement if the true value of the parameter is known to the desired quantity, which is the statistical variation of the parameter when the measurement value is known.

APPENDIX B-STRATIFIED SAMPLING

With this algorithm, there is the possibility of using stratified sampling. Imagine that in the run phase (rather than the adaptive phase) parameters are varied in groups, rather than all parameters being varied simultaneously. With a fairly small number of parameters in a group, the values of each parameter may be divided into strata such that the total number of candidate values N is the product of the number of strata. The jth coordinate of Θ_i for $i = 1, N$ is then generated using the formula

$$\Theta_{ij} = \frac{i_{strat}(i,j)-x}{n_{strat}(j)} ,$$

where x is a random number uniformly distributed between 0 and 1, $n_{strat}(j)$ is the number of strata for coordinate j, and $i_{strat}(i,j) = 1,...n_{strat}(j)$ is the stratum number for coordinate j associated with candidate i. This is illustrated in Table **1** for $N = 6$ candidates generated using stratified sampling. There are two coordinate parameters, the first divided into 3 strata and the second into 2 strata.

Table 1. The Function $i_{strat}(i,j)$ for N = 6 Candidates. Values of the First Parameter (j = 1) are Divided into $n_{strat}(1) = 3$ Strata and the Second (j = 2) into $n_{strat}(2) = 2$ Strata

i	*j* =	1	2
1		1	1
2		2	2
3		3	1
4		1	2
5		2	1
6		3	2

REFERENCES

[1] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculation by fast computing machines. J Chem Phys 1953; 21(6): 1087-92.

[2] Gilks WR, Richardson S, Spiegelhalter DJ. Markov chain monte carlo in practice. New York: Chapman and Hall 1996.

[3] Newman MEJ, Barkema GT. Monte carlo methods in statistical physics. Oxford: Oxford University Press 2002.

[4] Hammersley JM, Handscomb DC. Monte carlo methods. London: Methuen 1964.

[5] Barker AA. Monte carlo calculations of the radial distribution functions for a proton-electron plasma. Aust J Phys 1965; 18: 119-33.

[6] Hastings WK. Monte carlo sampling methods using markov chains and their applications. Biometrika 1970; 57(1): 97-109.

[7] Glynn PW, Heidelberger P. Analysis of initial transient deletion for parallel steady-state simulations. SIAM J Sci Stat Comput 1992;13(4): 904-22.

[8] Rosenthal J. Parallel computing and monte carlo algorithms. Far East J Theor Stat 2000; 4: 207-36.

[9] Brockwell AE. Parallel markov chain monte carlo simulation by pre-fetching. J Comput Graph Stat 2006; 15(1): 246-61.

[10] Feller W. An introduction to probability theory and its applications, Chapter XV, "Markov Chains". 2nd ed. New York: John Wiley & Sons Inc 1957.

[11] Miller G, Melo D, Martz H, Bertelli L. An empirical multivariate gaussian distribution representing uncertainty of biokinetic parameters for ^{137}Cs. Radiat Prot Dosim 2008; 131(2): 198-211.

[12] The BUGS (Bayesian inference Using Gibbs Sampling) Project. Available From: http://www.mrc-bsu.cam.ac.uk/bugs/.

[13] National Council on Radiation Protection, Uncertainties in Internal Radiation Dosimetry, NCRP report SC 6.3 (in press) status report available from: http://www.ncrponline.org/Current_Prog/SC_6-3.html [Accessed 7 February 2009].

[14] Miller G, Justus A, Vostrotin V, Dry D, Bertelli L. Poisson mixture model for measurements using counting. Radiat Prot Dosim 2009. Available From: http://rpd.oxfordjournals.org/content/early/2009/11/26/rpd.ncp268

Received: December 30, 2009 Revised: February 25, 2010 Accepted: March 01, 2010