Statistica Sinica **12**(2002), 751-767

MODEL-BASED ANALYSIS TO IMPROVE THE PERFORMANCE OF ITERATIVE SIMULATIONS

Chuanhai Liu and Donald B. Rubin

Lucent Technologies and Harvard University

Abstract: Inference using simulation has become a dominant theme in modern statistics, whether using the bootstrap to simulate sampling distributions of statistics, Markov chain Monte Carlo to simulate posterior distributions of parameters, or multiple imputation to simulate the posterior predictive distribution of missing values. Inference via simulations can, in some cases, be greatly facilitated by accompanying methods of analysis based on more traditional mathematical statistical techniques. Here we illustrate this point using one example of such technology: the analysis, based on a Markov-normal model of the stationary distribution underlying an iterative simulation, of parallel simulations before their convergence, thereby allowing a redesign of the simulation for better performance. The potential value of this approach is documented using an example involving censored data.

Key words and phrases: Bayesian methods, the CA-DA algorithm, the DA algorithm, the EM algorithm, the Gibbs sampler, Markov chain Monte Carlo, maximum likelihood estimation, multiple imputation, the PX-EM algorithm.

1. Introduction

There is little doubt that simulation methods have revolutionalized the field of statistics in the last quarter century. The availability of high-speed computing has not only made possible the analysis of complex models heretofore unapplied, but it has also changed the way many statisticians and scientists attack problems. For example, as suggested in Rubin (1984, Section 2.5), the ability to apply models that are scientifically appropriate for the data at hand, without investing in the creation of tedious closed-form mathematical analysis, means that as individuals we are no longer wedded to the use of possibly inappropriate techniques solely because of massive personal investment of time. The resulting scientific freedom when using simulation arises from the frequentist perspective – for example, via the use of the jackknife (e.g., Tukey (1993)) or the bootstrap (e.g., Efron and Tibshirani (1986)), from the Bayesian perspective – for example, via the use of Markov chain Monte Carlo (e.g., Geman and Geman (1984)) or sampling importance resampling (e.g., Rubin (1987a); called importance resampling in Gelman, Carlin, Stern and Rubin (1995)), and from mixed perspectives – for example, via multiple imputation (e.g., Rubin (1987b)).

Of critical importance to the most efficient use of simulation methods, however, is the continuing development of better mathematical statistical methods to analyze the output of simulations. Here we illustrate the use of relatively traditional analytic tools, but in relatively novel ways, to improve MCMC runs. Specifically, we show how a Markov-normal analysis of the output of short, parallel, iterative simulations before their convergence can lead to a redesigned simulation with better performance. The particular technique we describe will not always work, but it does illustrate what we believe is a very important theme in the future of applied statistics: the use of traditional statistical methodology to improve inferences via simulation.

2. Markov-Normal Analysis of Iterative Simulations Before Their Convergence: Redesign for Better Performance

Iterative simulation techniques such as Markov chain Monte Carlo (MCMC) methods have become standard tools for Bayesian computation in the last decade. The vast literature on this topic includes Metropolis and Ulam (1949), Metropolis, Rosenbluth Rosenbluth and Teller (1953), Hastings (1970), Geman and Geman (1984), Tanner and Wong (1987), Gelfand and Smith (1990), and Gelman and Rubin (1992). Recent statistical textbooks describing applications of MCMC techniques include Carlin and Louis (2000) and Gelman, Carlin, Stern, and Rubin (1995). These methods are also popular in the fields of the physical, chemical, and engineering sciences, for example, with reliability growth models (e.g., Erkanli, Mazzuchi and Soyer (1998)) and censored-data models (e.g., Hamada and Wu (1995); Liu and Sun (2000)).

Although MCMC algorithms are typically easy to implement, care must be taken when using them because they can have extremely slow rates of convergence. This problem has been noticed and attacked by many authors, for example, using auxiliary variable methods by Swendsen and Wang (1987), Goodman and Sokal (1988), and Besag and Green (1993); using blocking and grouping by Liu (1994), Liu, Wong and Kong (1994), and Robert and Sahu (1997); using extensions of the ideas of the PX-EM algorithm (Liu, Rubin and Wu (1998)) by Meng and van Dyk (1999), Liu and Wu (1999), and Liu (2001); and using restarted iterative simulations by Liu and Rubin (1996).

More specifically, Liu and Rubin (1996) proposed obtaining the maximum likelihood (ML) estimates of the target distribution from multiple sequences *before* their convergence, under the assumption that the target distribution of the simulated Markov chain is normal. They provided the needed technology and suggested that the pre-convergence normal-based ML estimates of the target distribution be used to define a restarting distribution for the simulation. They showed that the Markov-normal restarting procedure can be computationally extremely advantageous when the target distribution is nearly normal, especially in massively parallel or distributed computing environments where many sequences can be run for the same effective cost as one sequence.

Here, we demonstrate and extend this proposal in the context of an example involving censored lifetime data. First, as in Liu and Rubin (1996), the application shows that the normal-based ML estimate of the target distribution can be used to define a restarting distribution for the simulation. Second, it shows how this normal-based ML estimate of the target distribution can be used to provide simple but practical guidance for assessing rates of slowly-converging sequences. Third, the example shows how this estimate provides information on which components or functions of the components are converging slowly, thereby guiding the choice of methods for speeding the underlying MCMC algorithm.

3. Normal-based ML Estimation of the Target Distribution – Review and Extension

3.1. Normal-based ML estimation

We consider the situation with a d-dimensional parameter being simulated by m independent MCMC runs, all starting from common starting distribution $P^{(0)}(X)$ with common target distribution $P(X)$. Formally, suppose that the m independent d-dimensional sequences $\{X_j^{(t)} : t = 0, \ldots, n_j\}$, indexed by j with possibly different length n_j, are simulated, with $X_j^{(0)}$ iid $P^{(0)}(X)$, which has the same support as the target distribution $P(X)$, for $j = 1, \ldots, m$ and with the common transition distribution

$$X_j^{(t)} | (X_j^{(t-1)}, \ldots, X_j^{(0)}) \sim \mathrm{N}_d(\beta X_j^{(t-1)} + \gamma, \Delta), \quad t = 1, 2, \ldots, \tag{1}$$

where the $(d \times d)$ matrix β, d-dimensional vector γ, and $(d \times d)$ non-negative matrix Δ are unknown parameters. Assuming that the Markov chain converges to the target distribution, we have, first, that the target distribution is normal, because (1) is the AR(1) multivariate normal time-series model, and, second, that the mean vector and covariance matrix of this normal distribution $\mathrm{N}_d(\mu, \Psi)$ are given by

$$\mu = \beta\mu + \gamma \quad \text{and} \quad \Psi = \beta\Psi\beta' + \Delta, \tag{2}$$

respectively.

Liu and Rubin (1996) considered the case of $n_j = n$ for $j = 1, \ldots, m$ and provided closed-form expressions for the ML estimates of β, γ, and Δ. The above simple extension allowing for m sequences of various lengths provides certain flexibilities so that ML estimation can be applied in a sequential fashion to incorporate restarted multiple sequences. This simple extension also allows massively parallel or distributed computer processes to generate multiple sequences with different lengths. The ML estimates of the parameters from m sequences of various lengths can also be obtained from (1) in closed form as follows:

$$\hat{\beta}=\Bigg[\sum_{j=1}^{m}\sum_{t=1}^{n_j}\Big(X_j^{(t)}-\frac{1}{N}\sum_{j=1}^{m}\sum_{t=1}^{n_j}X_j^{(t)}\Big)\Big(X_j^{(t-1)}-\frac{1}{N}\sum_{j=1}^{m}\sum_{t=1}^{n_j}X_j^{(t-1)}\Big)'\Bigg]$$
$$\Bigg[\sum_{j=1}^{m}\sum_{t=1}^{n_j}\Big(X_j^{(t-1)}-\frac{1}{N}\sum_{j=1}^{m}\sum_{t=1}^{n_j}X_j^{(t-1)}\Big)\Big(X_j^{(t-1)}-\frac{1}{N}\sum_{j=1}^{m}\sum_{t=1}^{n_j}X_j^{(t-1)}\Big)'\Bigg]^{-1}, \quad (3)$$

$$\hat{\gamma}=\frac{1}{N}\sum_{j=1}^{m}\sum_{t=1}^{n_j}X_j^{(t)}-\frac{1}{N}\hat{\beta}\sum_{j=1}^{m}\sum_{t=1}^{n_j}X_j^{(t-1)}, \quad (4)$$

$$\hat{\Delta}=\frac{1}{N}\sum_{j=1}^{m}\sum_{t=1}^{n_j}\left(X_j^{(t)}-\hat{\beta}X_j^{(t-1)}-\hat{\gamma}\right)\left(X_j^{(t)}-\hat{\beta}X_j^{(t-1)}-\hat{\gamma}\right)', \quad (5)$$

$N=\sum_{j=1}^{m}n_j$, the total sample size for estimating the parameters. With the ML estimates of the parameters β, γ, and Δ given in (3)−(5), the ML estimates of the mean vector and covariance matrix of the normal-based target normal distribution can be computed from (2) using standard methods, including simple iterative methods. As a result, the parameters of the target distribution of $X_j^{(t)}$ following the AR(1) model can be well estimated before the convergence of the sequences $X_j^{(t)}$. The consistency of the ML estimate of the transition parameters β, γ and Δ of the Markov chain $AR(1)$ as m goes to the infinity (see, for example, Liu and Rubin (1996)) provides a relevant theoretical justification for the procedure.

3.2. Applications of the normal-based ML estimates

First, as was suggested by Liu and Rubin (1996), the normal-based ML estimate of the target distribution can be used to restart the Markov chains. For example, by inflating the covariance matrix we create an over-dispersed starting distribution for running multiple sequences that allows for easy assessment of the convergence of the simulated sequences using the method of Gelman and Rubin (1992).

Secondly, the estimate of β can be used to assess the rate of convergence of the underlying MCMC scheme. Liu, Wong and Kong (1994) proved that the rate of convergence of the Data Augmentation (DA) algorithm is determined by the auto-correlation of the sequences created by the algorithm. In current practice, this auto-correlation can only be well estimated using moment-based methods from converged sequences with satisfactory rates of convergence. Our normal-based ML estimate $\hat{\beta}$, obtained by conditioning on the starting points, however, can provide useful information about the rate of convergence even before the convergence of the simulated sequences, so that we can assess, before starting a full run, whether the underlying MCMC scheme will have problematically slow convergence. When the underlying MCMC scheme has a severely slow rate of convergence, it generally does not help to restart the iterative simulations. Rather, it may be necessary to modify (i.e., redesign) the MCMC scheme.

Thirdly, the normal-based ML estimates $\hat{\beta}$ and $\hat{\Delta}$ also allow the identification of components, or functions of the components of X, that have the slowest rates of convergence. This information helps find methods to accelerate the MCMC algorithm.

3.3. Needed technology

From (1) we have

$$(X^{(t)} - \mu) = \beta(X^{(t-1)} - \mu) + \varepsilon^{(t)}, \tag{6}$$

where $\varepsilon^{(t)} \sim \mathrm{N}_d(0, \Delta)$ is independent of the sequence $\{X^{(t)} : t = 0, \ldots, t{-}1\}$. The covariance matrix between $X^{(t)}$ and $X^{(t-1)}$, for the converged sequence $\{X^{(t)}\}$, is $\mathrm{Cov}\,(X^{(t)}, X^{(t-1)}) = \beta\Psi$, and hence the correlation matrix between normalized $\Psi^{-1/2}X^{(t)}$ and $\Psi^{-1/2}X^{(t-1)}$ is $\mathrm{Cor}(\Psi^{-1/2}X^{(t)}, \Psi^{-1/2}X^{(t-1)}) = \Psi^{-1/2}\beta\Psi^{1/2}$.

Let $P^{(t)}(X)$ be the distribution of $X^{(t)}$. Results on the rate of convergence of $P^{(t)}(X)$ to the target distribution $P(X)$ have appeared in various places. For example, Roberts and Sahu (1997) showed that the rate of convergence is the spectral radius of the matrix β, that is, the greatest of the absolute eigenvalues of β. Although this result is theoretically useful in terms of comparing different algorithms, it is difficult to use the result to design faster algorithms. For example, the eigenvalue corresponding to the spectral radius can be a complex number and thereby the corresponding eigenvector can be a vector of complex numbers.

Alternatively, we consider the square-root, denoted by ρ, of the spectral radius of the non-negative definite matrix

$$\begin{aligned} M &\equiv \left[\mathrm{Cor}(\Psi^{-1/2}X^{(t)}, \Psi^{-1/2}X^{(t-1)})\right] \left[\mathrm{Cor}(\Psi^{-1/2}X^{(t)}, \Psi^{-1/2}X^{(t-1)})\right]' \\ &= \Psi^{-1/2}\beta\Psi\beta'\Psi^{-1/2}. \end{aligned}$$

Because ρ can be viewed as the rate of convergence of $P^{(t)}(X)$ to $P(X)$, we call the matrix M the "squared convergence rate" matrix. Similarly, we call the matrix

$$S \equiv I - M = I - \Psi^{-1/2}\beta\Psi\beta'\Psi^{-1/2} = \Psi^{-1/2}\Delta\Psi^{-1/2} \tag{7}$$

the squared speed matrix of the underlying MCMC scheme.

From a Bayesian perspective, S in (7) is related to the fraction of observed information or fraction of missing information (Rubin (1987)), that is, the variance of X relative to the variance of X given the observed data. Intuitively, the bigger the within-iteration variance-covariance matrix Δ with respective to the target variance-covariance matrix, Ψ, the faster the algorithm converges because the fraction of missing information is less.

More specifically, let $Z^{(t)} = \Psi^{-1/2}(X^{(t)} - \mu)$. Then $Z^{(t)}|Z^{(t-1)} \sim \mathrm{N}_d\,(0, S)$. All the eigenvalues of S are in the interval $[0,1]$. When all the eigenvalues of S are one, the MCMC sequence converges approximately in one step, and when all the eigenvalues of S are zero, the MCMC sequence will never converge. In general, the MCMC sequence converges faster in the subspaces determined by the eigenvectors corresponding to large eigenvalues of S than those corresponding to small eigenvalues. Thus, by computing the eigenvalues and the corresponding eigenvectors of Δ with respect to Ψ, that is, of $S = \Psi^{-1/2}\Delta\Psi^{-1/2}$, we can identify the slowly-converging subspace. More formally, let $\lambda_1 \le \ldots \le \lambda_d$ be the d eigenvalues of $S = \Psi^{-1/2}\Delta\Psi^{-1/2}$, and let $s_1, \ldots, s_d$ be the corresponding eigenvectors. Then the slowest-converging k-dimensional $(1 \le k \le d)$ subspace is $\mathcal{L}(s_1, \ldots, s_k)$, the subspace spanned by the k eigenvectors $s_1, \ldots, s_k$.

The use of the rate of convergence $\rho = (1 - \lambda_1)^{1/2}$ as practical guidance is based on the following result. It follows easily from routine algebraic operations.

Result 1. *Suppose the* (*posterior*) *mean of a function of* X, $y = f(X)$, *is of interest. Suppose also that the moment estimator* $\bar{y}_{n_0,n} = \frac{1}{n}\sum_{t=1}^{n} y_{n_0+t}$ *is to be used, where* $y_{n_0+t} = f(X^{(n_0+t)})$ *for* $t = 1, \ldots, n$ *and* $X^{(n_0+t)} \sim P(X)$, *and that the sequence* $y_{n_0+1}, \ldots, y_{n_0+n}$ *is an* $AR(1)$ *process with the variance* σ^2 *and first-order autocorrelation* $r > 0$. *Then* (*i*) *the rate of convergence of the sequence* $y_{n_0+1}, \ldots, y_{n_0+n}$ *is* $\rho = r$, *and* (*ii*) *the variance of the moment estimate* $\bar{y}_{n_0,n}$ *of the mean of* y *is*

$$var(\bar{y}_{n_0,n}) = \frac{\sigma^2}{n}\left[\frac{1+r}{1-r} - \frac{2r(1-r^n)}{n(1-r)^2}\right] \approx \frac{1+r}{1-r}\frac{\sigma^2}{n},$$

when n *is sufficiently large.*

Thus, as was noted by Tierney (1994), we have the following result.

Corollary 1. *The asymptotic standard deviation of the sample mean* $\bar{y}_{n_0,n}$ *is* $\frac{\sigma}{n^{1/2}}(\frac{1+r}{1-r})^{1/2}$, *and thus, for the sample mean, the equivalent effectively-independent-sample-size in a run of length* n *is asymptotically* $n(1-r)/(1+r)$.

For convenience, we call

$$F = \frac{1+r}{1-r} \tag{8}$$

the equivalent sample size factor. In general, for a multivariate AR(1) process (6), the equivalent sample size factor in (8) can be adjusted based on the following results for the covariance matrix of $\bar{X}_{n_0,n} = \frac{1}{n}\sum_{t=1}^{n} X^{(n_0+t)}$ after effective convergence of $X^{(n_0)}$. Routine algebraic operations lead to the following result.

Result 2. Let $B = \Psi^{-1/2}\beta\Psi^{1/2}$. Then $S = I - BB'$ and, for large n,

$$\begin{aligned}
&\mathrm{Cov}\,(\Psi^{-1/2}\bar{X}_{n_0,n}, \Psi^{-1/2}\bar{X}_{n_0,n}) \\
&= \frac{1}{n}(I-B)^{-1}(I-BB')(I-B')^{-1} - \frac{1}{n^2}\Big[(I-B)^{-2}(I-B^n)B \\
&\quad +B'(I-(B')^n)(I-B')^{-2}\Big] \\
&\approx \frac{1}{n}(I-B)^{-1}(I-BB')(I-B')^{-1}. \qquad (9)
\end{aligned}$$

Corollary 2. *Let λ be the maximum eigenvalue of* (9). *To obtain the same precision for the moment estimate $\bar{X}_{n_0,n}$ from an independent sample of size n, the length of the sequence of a run after reaching equilibrium needs to be larger than n roughly by the factor λ.*

Thus, before the convergence of the MC simulation, we can use the results from the Markov-normal analysis and the conclusion of Corollary 2 to obtain a rough estimate of the ultimate length of sequence, *after effective convergence,* needed for estimation via MCMC methods. This result also provides guidance on the choice of many independent runs versus a few long runs.

4. Example

4.1. The data and model

The data in Table 1 came from a router bit experiment reported originally by Phadke (1986). Hamada and Wu (1995) considered fitting a normal regression model with 23 effects to the log-lifetime data consisting of 14 left-censored, 10 interval-censored, and 8 right-censored values. The 23 effects consist of the intercept, the seven two-level factor main effects A, B, C, F, G, H and I, three two-level pseudo-factors D1, D2 and D3 for the main effect of the factor D, and twelve interactions AF, AH, AI, BF, BG, BI, CG, CH, CI, FI, GI and HI. Letting X and Y be the (32×23) design matrix and the 32-dimensional vector of the log-lifetime in the underlying complete dataset, respectively, letting β be the regression coefficients, and letting σ^2 be the variance of the errors, we write the complete-data model as $Y|(\beta,\sigma^2) \sim \mathrm{N}_{32}(X\beta, \sigma^2 I_{23})$. Hamada and Wu (1995) used a prior distribution for (β,σ^2) of the form: $\sigma^{-2} \sim \mathrm{Gamma}(\nu_0/2, \nu_0 s_0^2/2)$ and $\beta|\sigma^2 \sim N_{23}(\beta_0, \sigma^2 A_0^{-1})$, with $\nu_0 = 1$, $s_0^2 = 0.01$, $\beta_0 = (1.5, 0, \ldots, 0)'$, and $A_0 = 0.0001 I_{23}$. They also considered alternative choices of A_0, $A_0 = 0.01 I_{23}$ and

I_{23}, for checking the sensitivity of the posterior distributions to the specification of the prior distributions.

Table 1. Design and lifetime data for the router bit experiment.

	Design													Data	
				D			E							Censoring Interval	
Run	A	B	C	D1	D2	D3	E1	E2	E3	F	G	H	I	Left	Right
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	4
2	-1	-1	-1	-1	1	1	-1	1	1	1	1	-1	-1	0	1
3	-1	-1	-1	1	-1	1	1	1	-1	-1	1	1	-1	0	1
4	-1	-1	-1	1	1	-1	1	-1	1	1	-1	1	-1	17	∞
5	-1	1	1	1	-1	1	-1	-1	-1	1	1	-1	-1	0	1
6	-1	1	1	1	1	-1	-1	1	1	-1	-1	-1	-1	2	3
7	-1	1	1	-1	-1	-1	1	1	-1	1	-1	1	-1	0	1
8	-1	1	1	-1	1	1	1	-1	1	-1	1	1	-1	0	1
9	1	-1	1	1	1	-1	-1	-1	-1	-1	1	1	-1	17	∞
10	1	-1	1	1	-1	1	-1	1	1	1	-1	1	-1	2	3
11	1	-1	1	-1	1	1	1	1	-1	-1	-1	-1	-1	0	1
12	1	-1	1	-1	-1	-1	1	-1	1	1	1	-1	-1	3	4
13	1	1	-1	-1	1	1	-1	-1	-1	1	-1	1	-1	0	1
14	1	1	-1	-1	-1	-1	-1	1	1	-1	1	1	-1	2	3
15	1	1	-1	1	1	-1	1	1	-1	1	1	-1	-1	0	1
16	1	1	-1	1	-1	1	1	-1	1	-1	-1	-1	-1	3	4
17	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	17	∞
18	-1	-1	-1	-1	1	1	-1	1	1	1	1	-1	1	0	1
19	-1	-1	-1	1	-1	1	1	1	-1	-1	1	1	1	0	1
20	-1	-1	-1	1	1	-1	1	-1	1	1	-1	1	1	17	∞
21	-1	1	1	1	-1	1	-1	-1	-1	1	1	-1	1	0	1
22	-1	1	1	1	1	-1	-1	1	1	-1	-1	-1	1	17	∞
23	-1	1	1	-1	-1	-1	1	1	-1	1	-1	1	1	14	15
24	-1	1	1	-1	1	1	1	-1	1	-1	1	1	1	0	1
25	1	-1	1	1	1	-1	-1	-1	-1	-1	1	1	1	17	∞
26	1	-1	1	1	-1	1	-1	1	1	1	-1	1	1	3	4
27	1	-1	1	-1	1	1	1	1	-1	-1	-1	-1	1	17	∞
28	1	-1	1	-1	-1	-1	1	-1	1	1	1	-1	1	3	4
29	1	1	-1	-1	1	1	-1	-1	-1	1	-1	1	1	0	1
30	1	1	-1	-1	-1	-1	-1	1	1	-1	1	1	1	3	4
31	1	1	-1	1	1	-1	1	1	-1	1	1	-1	1	0	1
32	1	1	-1	1	-1	1	1	-1	1	-1	-1	-1	1	17	∞

4.2. The DA algorithm and Markov-normal analysis

We consider fitting this model with the Data Augmentation (DA) algorithm (Tanner and Wong (1987)). Each iteration of DA consists of two steps: an

I-step that imputes the complete-data lifetime of the router bit for each run from its predictive distribution given the current draw of the parameter vector (β, σ^2), and a *P-step* that takes a draw of the parameter vector (β, σ^2) from its posterior distribution given the currently imputed complete data. Detailed implementation is a special case of the DA algorithm for the analysis of censored data using covariance adjustments discussed by Liu and Sun (2000). Because the DA algorithm has two steps, rather than three or more as with a general Gibbs sampler, the simulated sequence of the parameter vector $(\beta^{(t)}, \sigma^{(t)})$ is a Markov chain, a condition theoretically needed for our Markov-normal analysis.

We started the DA algorithm from its P-step with a simply imputed complete dataset: the missing lifetime is imputed with its left-censoring value, the averages of the two censoring values, and its right-censoring value for each of the left-, interval- and right-censored data, respectively. Although better starting values can be obtained using the EM algorithm (Dempster, Laird and Rubin (1977); Liu and Sun (2000)), we used these simple starting values to illustrate that our method does not require accurate starting values. A single chain was run up to 20,000 iterations. The last half-sequences of the parameters obtained from the single chain are displayed in Figure 1, which indicate a slow rate of convergence of the algorithm and the uselessness of the sequence of length 20,000 iterations for computing posterior distributions accurate enough for reliable statistical inference.

To investigate the DA scheme in terms of the normal-based transition distribution, we ran $m = 20$ parallel MCMC sequences of common length, with common starting values but different seeds for the underlying random number generator. We replicated the process three times, once with $n_i = 1000$, $n_i = 5000$, and $n_i = 10,000$, respectively. The components considered consist of the 23 regression coefficients and $\ln(\sigma^2)$, the logarithm of the variance of the residuals. The nine smallest eigenvalues are less than 10^{-3}, with the smallest one below 10^{-4}. Although they do not appear to have converged, the estimated eigenvalues appear to be consistent in the sense that they indicate the convergence rate of the DA scheme is extremely slow.

Because the results with $n_i = 10,000$ are the most precise, we focus on them (although results from longer runs did show that the smallest eigenvalue is much smaller than 0.0001). The corresponding eigenvectors indicated that (i) $\ln(\sigma^2)$ is the fastest converging component, with a fully satisfactory speed of convergence close to 0.86; and (ii) the rate of convergence in the remaining 23 dimensional space is not fully satisfactory, except for a one-dimensional subspace that corresponds to the second largest eigenvalue 0.58. According to Corollary 2, the simulation requires at least another 1,000,000,000 iterations if an equivalent effectively-independent sample size of $10,000$ is desired! Applying the method of Gelman and Rubin ((1992), henceforth GR) to assess the convergence of 20 parallel restarted sequences (not reported) also indicates that the DA algorithm converges too slowly to be useful. Thus, in order to obtain a reliable estimate of the posterior distribution via iterative simulation, a faster converging version of the DA algorithm is needed: it is hopeless to run the current version and simply wait, at least using current computing resources.

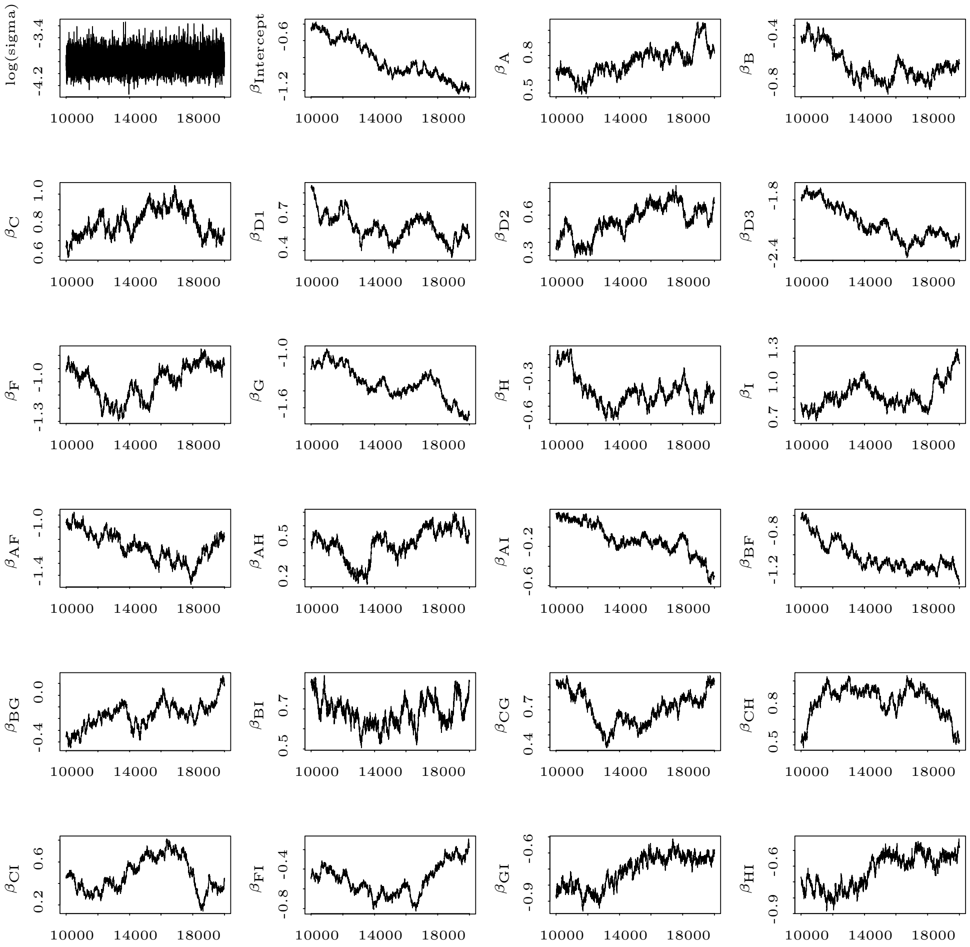

Figure 1. The last half sequences of the 24 parameters in the model for the router lifetime experiment. The sequences, 20,000 iteration long, are simulated using the standard DA algorithm.

4.3. Speeding up the DA algorithm

The Markov-normal analysis provides useful guidance for speeding up the DA algorithm: the speeding-up needs to take place in the space of the regression coefficients. The reason for the slow rate of convergence is due to the highly censored sufficient statistics for the regression coefficients. To accelerate the DA algorithm, we make use of the idea of covariance adjustment for speeding up MCMC algorithms (Liu (2001)), which builds on the idea of covariance adjustment (CA) for EM (PX-EM: Liu, Rubin and Wu (1998)).

In general the full, and hence ideal, CA step is difficult to implement, because it is not easy to take a joint draw of the parameters and sufficient statistics. However, it is easy to take a joint draw of the parameters and sufficient statistics when the corresponding design matrix is for a saturated design. Suppose that the set of parameters for the actual design matrix can be obtained as the union of the sets of parameters for K saturated designs, that is, the union of the K sets of parameters for the saturated designs spans the same space as spanned by the parameters of the original design. Then cycling through the K CA steps for the K saturated designs (i.e., cycling through the K joint drawings of parameters and sufficient statistics), implements a partial CA adjustment. This idea of contructing saturated design matrices to span the same space as the original design matrix is closely related to the "space-filling" parameterizations used in iterative maximization rountines (ECM: Meng and Rubin (1993); ECME: Liu and Rubin (1994)), and implicit even in Iterative Proportional Fitting (e.g., Bishop, Fienberg and Holland (1975)).

For our example, we used the following collection of space-filling design matrices, constructed in such a way that the 23 factorial effects correspond to the parameters of the over-lapping saturated models for the subsets of the original factors.

- {Intercept, A, B, C, D1, D2, D3, F, G, H, AF, AH, BF, BG, CG, CH};
- {I, AI, BI, CI, D1, CH, H, AH};
- {I, AI, BI, CI, D3, G, CG, BG};
- {I, AI, BI, CI, D2, BF, AF, F};
- {I, FI, GI, HI, A, AF, D3, AH};
- {I, FI, GI, HI, D2, C, CH, CG};
- {I, FI, GI, HI, D1, BG, BF, B}.

Details of the implementation are omitted, they appear in Liu (2001).

With such a partial covariance adjustment, dramatically improved computational efficiency is obtained, as can be seen by comparing Figure 2 to Figure 1 (note the ranges over which the corresponding sequences wandered during the same number of iterations in both Figure 1 and Figure 2). The three smallest eigenvalues of the normal-based estimate of the improved transition distribution, 0.5944, 0.0048 and 0.0023, confirm the improved efficiency, but also indicate that there is a two-dimensional subspace corresponding to the two smallest eigenvalues that might need further consideration. In the following section, we use this version of the covariance-adjusted DA (CA-DA) algorithm to show how the Markov-normal analysis can be used to create an over-dispersed starting distribution for running multiple chains, a process that leads to a reliable estimate of the posterior distribution for inference.

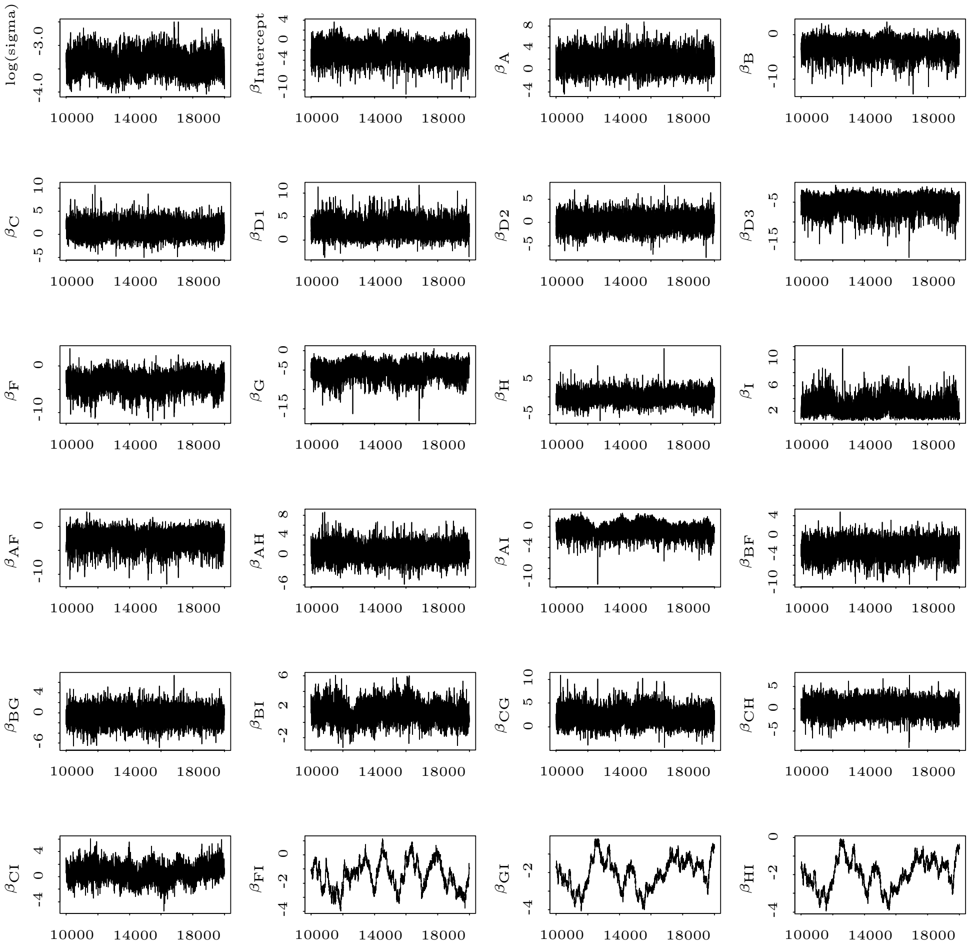

Figure 2. The last half sequences of the 24 parameters in the model for the router lifetime experiment. The sequences, 20,000 iteration long, are simulated using the covariance-adjusted DA algorithm.

4.4. Computing the posterior distribution

Using the normal-based maximum likelihood estimates of the parameters from the 20 parallel runs of CA-DA for 10,000 iterations, we obtained a multivariate normal approximation to the target distribution of the 24 parameters. We then created a 24-dimensional multivariate t-distribution on 5 degrees of freedom, with center and scatter matrix given by the mean vector and the covariance matrix (inflated by a factor of 1.5) of the normal approximation, respectively. We made 20 parallel runs of CA-DA using this t-distribution as the presumably overdispersed starting distribution. The plots of all the sequences in the first 1000 iterations for the 24 parameters, not included, support the view that this distribution is overdispersed, in the sense that it has longer tails than the target distribution. We then ran $m = 20$ parallel runs of length of $20,000$ each.

To assess the convergence of the sequence numerically in terms of the posterior mean of the parameters, we computed the GR scale reduction coefficients for the 24 parameters. The results from the $m = 20$ parallel runs of length of $20,000$ each (not reported) showed that the GR scale reduction coefficients for the effects of FI, GI and HI on the router bit lifetime indicate that a faster converging MC algorithm than the current version CA-DA might still be helpful. This is consistent with the results from the Markov-normal analysis given in the previous section. Comparing the histograms for the parameters in the model with $A_0 = 0.0001I$ with a restarting distribution (not reported) also show that the Markov-normal analysis provides a restarting distribution that is slightly more overdispersed than the target distribution. Table 2 gives the GR estimates of the quantiles of the posterior distributions of the 24 parameters.

4.5. Investigating sensitivity of the posterior distribution to the specification of the prior distribution

To check the sensitivity of the posterior distribution to the specification of the prior distribution, Hamada and Wu (1995) also considered $A_0 = 0.01I_{23}$ and $A_0 = I_{23}$. For these two cases, we repeated the above procedure and obtained the posterior distributions of the parameters. The GR results for the parameters in the case of $A_0 = I_{23}$ are displayed in Table 2. From our computations, it appears that the posterior distribution is quite sensitive to the specification of the prior distribution, a somewhat different conclusion from that of Hamada and Wu (1995), who essentially ran a single chain MCMC with 14 iterations in analogy with the method of using DA described in Wei and Tanner (1990). For the cases of $A_0 = 0.01I_{23}$ and $A_0 = I_{23}$, the DA algorithm does not appear to be as problematic as it was for the case of $A_0 = 0.0001I_{23}$.

Table 2. The GR estimates of of the quatiles of the posterior distributions of the 24 parameters in the model with $A_0 = 0.0001I$ and ln(lifetime)-1.5 for the router bit lifetime example based on 20 parallel runs of length of 20,000 each. The GR results are obtained using a S-plus function provided by Prof. Andrew Gelman.

	$A_0 = 0.0001I$					$A_0 = I$				
Parameter	2.5%	25.0%	50.0%	75.0%	97.5%	2.5%	25.0%	50.0%	75.0%	97.5%
$\beta_{\text{Intercept}}$	-5.7	-3.2	-2.2	-1.2	0.7	-1.5	-0.9	-0.7	-0.5	-0.1
β_{A}	-1.4	0.4	1.3	2.2	4.6	-0.2	0.1	0.3	0.5	1.0
β_{B}	-6.4	-3.8	-2.7	-1.8	-0.1	-1.6	-1.0	-0.8	-0.6	-0.2
β_{C}	-1.6	0.2	1.1	2.0	4.3	-0.3	0.1	0.3	0.5	1.0
β_{D1}	-0.2	1.5	2.4	3.4	6.1	0.1	0.5	0.6	0.9	1.4
β_{D2}	-2.9	-0.9	0.1	0.9	2.9	-0.6	-0.2	0.0	0.2	0.6
β_{D3}	-10.4	-6.8	-5.4	-4.3	-2.7	-2.5	-1.7	-1.4	-1.2	-0.9
β_{F}	-7.4	-4.3	-3.1	-2.1	-0.4	-1.7	-1.1	-0.8	-0.6	-0.3
β_{G}	-9.7	-6.0	-4.7	-3.6	-2.0	-2.3	-1.5	-1.3	-1.0	-0.7
β_{H}	-3.4	-1.0	-0.1	0.9	3.0	-0.6	-0.2	-0.0	0.2	0.6
β_{I}	1.0	1.8	2.5	3.4	6.0	0.4	0.6	0.8	1.0	1.5
β_{AF}	-6.1	-3.5	-2.4	-1.5	0.1	-1.5	-0.9	-0.7	-0.5	-0.2
β_{AH}	-2.1	-0.2	0.8	1.7	4.0	-0.4	-0.1	0.1	0.3	0.8
β_{AI}	-3.2	-1.5	-0.8	-0.2	1.2	-0.7	-0.3	-0.2	0.0	0.4
β_{BF}	-6.1	-3.4	-2.3	-1.4	0.3	-1.4	-0.8	-0.6	-0.4	-0.0
β_{BG}	-3.6	-1.5	-0.6	0.3	2.3	-0.8	-0.4	-0.2	0.0	0.4
β_{BI}	-1.1	0.3	0.9	1.6	3.4	-0.3	0.1	0.3	0.4	0.9
β_{CG}	-0.7	0.9	1.9	2.9	5.5	0.0	0.4	0.5	0.8	1.3
β_{CH}	-3.0	-0.9	0.0	1.0	3.0	-0.6	-0.1	0.1	0.2	0.7
β_{CI}	-1.2	0.3	1.0	1.8	3.5	-0.3	0.1	0.3	0.5	0.9
β_{FI}	-4.8	-2.4	-1.7	-1.2	0.0	-1.1	-0.6	-0.5	-0.3	0.0
β_{GI}	-4.5	-2.2	-1.5	-1.1	-0.4	-1.3	-0.8	-0.6	-0.5	-0.2
β_{HI}	-4.5	-2.1	-1.4	-1.1	-0.3	-1.0	-0.6	-0.4	-0.2	0.1
$\ln\sigma^2$	-7.6	-7.1	-6.8	-6.5	-5.8	-1.7	-1.1	-0.8	-0.4	0.4

For comparison with the case of $A_0 = 0.0001I_{23}$, we compared the histograms for the parameters in the model with $A_0 = I_{23}$ to the corresponding density functions of the starting distribution obtained from the Markov-normal analysis with the initial 20 parallel runs of length 1,000 each with simple starting values. Again, these starting distributions appear to be appropriately overdispersed.

5. Discussion

We expect that simulation will continue its growth as a critical tool in modern inferential statistics. We also expect, however, a parallel growth of the application of more traditional mathematical statistical methods of analysis to improve the design of such simulations. We have discussed and illustrated the use of Markov-normal analysis of iterative simulations before their convergence, a method proposed originally in Liu and Rubin (1996). The potential benefits of the method reported in this paper can be summarized as: (i) to create over-dispersed restarting distributions for running multiple chain iterative simulations (as in Gelman and Rubin (1992)); (ii) to estimate the efficiency of MCMC algorithms in terms of their rate of convergence and thereby be warned of problems of convergence early in the simulation; and (iii) to provide information useful for modifying the underlying iterative simulation scheme by identifying components that appear to generate very slowly converging sequences with potentially unreliable results. Liu and Rubin (1996) discuss other relevant aspects of the method, such as its efficiency in massively parallel or distributed computing environments, where many sequences can be run for the same effective cost as one sequence.

It is, however, difficult to apply this Markov-Normal analysis when the number of variables is large. In the context of Bayesian estimation, it is useful to consider the model *parameters* because (i) the dimensionality of the parameters can be substantially smaller than the dimensionality of the *missing* data; and (ii) the normality assumption can be practically appropriate, at least when the sample size of the observed data used for estimating the parameters is large. Although it is simple to apply the method to marginals of a MCMC, care must be taken because the Markov property generally does not hold for the marginals although it does for DA (a two-step Gibbs sampler). In particular, Liu and Rubin (1996) noticed that the method can be misleading when applied to the non-Markovian marginals of a MCMC (e.g., a Gibbs sampler with more than two steps). Further investigation of the use of preliminary Markov-normal analysis for models with many variables and parameters should be pursued, especially in the context of massively parallel or distributed computing environments. Thus, we view work presented here as only a simple illustration of the kinds of techniques that should be developed in the coming years to help scientists and statisticians design and analyze inferential simulations.

Acknowledgement

The authors thank the Editor, an associate editor, and a referee for insightful and constructive comments.

References

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

Bishop, Y. M. M., Fienberg, S. E. and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. The MIT Press.

Carlin, B. P. and Louis, T. A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis*, 2nd Edition. Chapman and Hall, New York.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Statist. Soc. Ser. B* **39**, 1-38.

Efron, B. and R. Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy (with discussion). *Statist. Sci.* **1**, 54-96.

Erkanli, A., Mazzuchi, T. A. and Soyer, R. (1998). Bayesian computations for a class of reliability growth models. *Technometrics* **40**, 14-23.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.* **85**, 398-409.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995). *Bayesian Data Analysis.* Chapman and Hall, New York.

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

Goodman J. and Sokal, A. D. (1988). Multigrid Monte Carlo methods: Conceptual foundations. *Phys. Rev. D* **40**, 2035-2071.

Hamada, M. and Wu, C. F. J. (1995). Analysis of censored data from fractionated experiment: a Bayesian approach. *J. Amer. Statist. Assoc.* **90**, 467-477.

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

Liu, C. (2001). Covariance adjustment for Markov chain Monte Carlo sampling – a general framework and the covariance-adjusted data augmentation algorithm. Under revison for *J. Amer. Statist. Assoc.*

Liu, C. and Rubin, D. B. (1994). The ECME algorithm: An simple extension of EM and ECM with faster monotone convergence. *Biometrika* **81**, 633-48.

Liu, C. and Rubin, D. B. (1996). Markov-normal analysis of iterative simulations before their convergence. *J. Econometrics* **75**, 69-78.

Liu, C., Rubin, D. B. and Wu, Y. (1998). Parameter expansion for EM acceleration – the PX-EM algorithm. *Biometrika*, **85**, 755-770.

Liu, C. and Sun, D. X. (2000). Analysis of interval censored data from fractionated experiments using covariance adjustments. *Technometrics* **42**, 353-365.

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *J. Amer. Statist. Assoc.* **89**, 958-966.

Liu, J. S., Wong, W. H. and Kong, A. (1994). Covariance structure of the Gibbs sampler with applications. *Biometrika* **81**, 27-40.

Liu, J. S. and Wu, Y. (1999). Parameter expansion scheme for data augmentation. *J. Amer. Statist. Assoc.* **94**, 1264-1274.

Meng, X. L. and van Dyk, D. (1999). Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika* **86**, 301-320.

Meng, X. L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika* **80**, 267-78.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. and Teller, A. H. (1953). Equations of state calculations by fast computing machines. *J. Chemical Physica* **21**, 1087-1091.

Metropolis, N., and Ulam, S. (1949). The Monte Carlo methods. *J. Amer. Statist. Assoc.* **44**, 335-341.

Phadke, M. S. (1986). Design optimization case studies. *AT&T Tech. J.* **65**, 51-68.

Roberts, G. O. and Sahu, S. K. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. Roy. Statist. Soc. Ser. B* **59**, 291-317.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Ann. Statist.* **12**, 1151-1172.

Rubin, D. B. (1987a). A noniterative sampling-importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. Comment on "The calculation of posterior distributions by data augmentation" by M. A. Tanner and W. H. Wong. *J. Amer. Statist. Assoc.* **82**, 543-546.

Rubin, D. B. (1987b). *Multiple Imputation for Nonresponse in Surveys.* Wiley, New York.

Swendsen, R. H. and Wang, J. S. (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.* **58**, 86-88.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation (with discussion). *J. Amer. Statist. Assoc.* **82**, 528-550.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Ann. Statist.* **22**, 1701-1762.

Tukey, J. W. (1993). Major challenges for multiple-response (and multiple-adjustment) analysis. In *Multivariate Analysis: Future Directions* (Edited by C. R. Rao), 401-421. North-Holland, New York.

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. *J. Amer. Statist. Assoc.* **85**, 699-704.

Statistics Research Department, Lucent Technologies, Bell Laboratories, 600 Mountain Avenue, Room 2C-262, Marray Hill, NJ07974, U.S.A.

E-mail: liu@research.bell-labs.com

Dempartment of Statistics, Harvard University, Cambridge, MA02138, U.S.A.

E-mail: rubin@stat.harvard.edu

(Received July 2000; accepted January 2002)