Stat Comput
DOI 10.1007/s11222-006-9009-4

Acceleration of the Multiple-Try Metropolis algorithm using antithetic and stratified sampling

Radu V. Craiu · Christiane Lemieux

Received: July 2005 / Accepted: November 2006

Abstract The Multiple-Try Metropolis is a recent extension of the Metropolis algorithm in which the next state of the chain is selected among a set of proposals. We propose a modification of the Multiple-Try Metropolis algorithm which allows for the use of correlated proposals, particularly antithetic and stratified proposals. The method is particularly useful for random walk Metropolis in high dimensional spaces and can be used easily when the proposal distribution is Gaussian. We explore the use of quasi Monte Carlo (QMC) methods to generate highly stratified samples. A series of examples is presented to evaluate the potential of the method.

Keywords Antithetic variates · Markov Chain Monte Carlo · Extreme antithesis · Korobov rule · Latin Hypercube sampling · Quasi Monte Carlo · Sobol' sequence · Multiple-Try Metropolis · Random-Ray Monte Carlo

1 Introduction

Is is well recognized that the Markov Chain Monte Carlo (MCMC) methods provide huge support for realistic statistical modelling. In recent years, as the statistical models have increased in complexity and size, there is a greater demand for fast MCMC algorithms and more reliable convergence diagnostics. A promising direction is represented by the so-called local search samplers and adaptive algorithms. For examples, we refer to the papers by Atchade and Perron (2005), Atchade and Rosenthal (2005), Liu et al. (2000), Gilks et al. (1994), Chen and Schmeiser (1993). In this paper we discuss possible accelerations of the Multiple-Try Metropolis (MTM) of Liu et al. (2000, henceforth denoted LLW) via correlated proposals. LLW introduce the MTM as a generalization of the classical Metropolis algorithm which allows one to select, at each update, among multiple proposals. The main advantage of the MTM is that it explores a larger portion of the sample space resulting in better mixing and shorter running times. In addition, LLW propose the use of MTM with the Adaptive Direction Sampling of Gilks et al. (1994) as well as the hit-and-run algorithm (Chen and Schmeiser, 1993) and the griddy Gibbs sampler (Ritter and Tanner, 1992). The modifications we propose here for the MTM can be used directly in all of the above.

Recent approaches to the acceleration of MCMC algorithms have included the use of antithetic variates (Frigessi et al., 2000; Craiu and Meng, 2005), and quasi Monte Carlo (QMC) methods (Owen and Tribble, 2004; Lemieux and Sidorsky, 2006), in which *highly-uniform point sets*—such as the one shown on Fig. 2—are used to produce structured sampling schemes meant to improve upon random draws. The combination of MCMC and QMC is still at an early stage, but it seems like a promising approach since QMC methods have been shown to be very successful in the context of multivariate integration, especially in the area of finance (see, for example, Paskov and Traub, 1995; Caflisch et al., 1997).

Antithetic coupling has proven to be particularly effective in MCMC algorithms with monotone kernels, such as Gibbs and slice samplers, or in perfect sampling processes in

R. V. Craiu (✉)
Department of Statistics, University of Toronto, 100 St. George Street, Toronto, ON, M5S 3G3, Canada
e-mail: craiu@utstat.toronto.edu

C. Lemieux
Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada
e-mail: clemieux@math.uwaterloo.ca

which the updating function is monotone in all arguments. The Metropolis-Hastings algorithm, one of the most frequently used MCMC algorithms in practice, does not have the monotone properties required for an efficient antithetic coupling. We propose the use of antithetic and stratified variates for Metropolis algorithms via the MTM. The new algorithm, *Multiple Correlated-Try Metropolis (MCTM)* selects among correlated proposals instead of independent ones. Section 2 contains the general construction of MCTM. Correlation among the proposals can be introduced in various ways but in this paper we study the antithetic and stratified approaches. Both are detailed in Section 3 in the context of a random-walk Metropolis with either (a) univariate proposal distribution for which the inverse cumulative distribution function is available, or (b) multivariate Gaussian proposals. Section 4 contains three examples for which we compare the performances of MCTM and MTM. We end with discussion in Section 5.

2 Multiple correlated-try metropolis

Suppose we want to draw samples from a target distribution characterized by the density function $\pi(x)$. Metropolis algorithms do that by first choosing a proposal transition rule $T(x; y)$, which gives the density function for the future state y given the current state x. In the MTM algorithm proposed by LLW, a weighting function $w(x, y)$ defined by

$$w(x, y) = \pi(x)T(x; y)\lambda(x, y),$$

is also specified, where $\lambda(x, y)$ is a nonnegative symmetric function (in x and y) that can be chosen by the user. (To simplify the arguments, unless otherwise stated we take $\lambda(x, y) = 1$ throughout this paper.) To update the current state x, the MTM performs the following steps.

1. Draw k trial proposals $y_1, \ldots, y_k$ from $T(x; y)$. Compute $w(y_j, x)$ for each j;
2. Select y among the k proposals with probability proportional to $w(y_j, x)$, $j = 1, \ldots, k$.
3. Draw $x_1^*, \ldots, x_{k-1}^*$ variates from the distribution $T(y; x)$ and let $x_k^* = x$;
4. Accept y with generalized acceptance probability

$$r_g = \min\left\{1, \frac{w(y_1, x) + \cdots + w(y_k, x)}{w(x_1^*, y) + \cdots + w(x_k^*, y)}\right\}.$$

While it is obvious that the proposals do not need to be independently generated, some care is required in implementing the MTM with correlated proposals, especially if we want to maintain the reversibility of the Markov chain. Consider the MTM algorithm in which the proposals are generated jointly from $\tilde{T}(x; y_1, \ldots, y_k)$ and are exchangeable. In other words, the transition rule now specifies how to generate a (correlated) sample $y_1, \ldots, y_k$ of future states given the current state x, and $\tilde{T}(x; y_1, \ldots, y_k)$ is the conditional joint density function of that sample given x. Furthermore, we assume that the marginal transition kernel is equal to the original existing kernel $T(x; y)$ that is used to generate independent trials, i.e.

$$\int \tilde{T}(x; y_1, \ldots, y_k) dy_1 \ldots dy_{i-1} dy_{i+1} \ldots dy_k = T(x; y_i), \quad \forall\, 1 \le i \le k.$$

We call this approach the *Multiple Correlated-Try Metropolis* (MCTM) algorithm. The motivation here is that by carefully choosing the joint distribution of the correlated sample $y_1, \ldots, y_k$, we hope to sample the state space in a more structured way than by using a set of independent proposals, as done in the MTM. The new MCTM algorithm is as follows:

1. Draw k trial proposals $y_1, \ldots, y_k$ from $\tilde{T}(x; y_1, \ldots, y_k)$. Compute $w(y_j, x) = \pi(y_j)T(y_j; x)\lambda(y_j, x)$, for each j;
2. Select y among the k proposals with probability proportional to $w(y_j, x)$, $j = 1, \ldots, k$.
3. Draw $(x_1^*, \ldots, x_{k-1}^*)$ variates from the conditional transition kernel $\tilde{T}(y; x_1, \ldots, x_{k-1} \,|\, x_k = x)$ and let $x_k^* = x$;
4. Accept y with generalized acceptance probability

$$r_g = \min\left\{1, \frac{w(y_1, x) + \cdots + w(y_k, x)}{w(x_1^*, y) + \cdots + w(x_k^*, y)}\right\}.$$

Proposition 2.1. *The Markov chain defined with the algorithm above has stationary distribution π and satisfies the detailed balance condition.*

Proof: The proof follows closely from the one given by LLW for the original MTM. If $A(x, y)$ is the actual transition probability and $I(\cdot)$ is the indicator function that shows which y_j has been selected at step 2, then

$$\begin{aligned}\pi(x)A(x, y) &= \pi(x)P\left[\cup_{j=1}^k \{Y_j = y\} \cap \{I = j\} \,|\, x\right]\\ &= k\pi(x)P[\{Y_1 = y\} \cap \{I = 1\} \,|\, x]\\ &= k\pi(x) \int \tilde{T}(x; y, y_2, \ldots, y_k)\\ &\quad \times \frac{w(y, x)}{w(y, x) + \sum_{i=2}^k w(y_i, x)}\\ &\quad \times \min\left\{1, \frac{w(y, x) + \sum_{i=2}^k w(y_i, x)}{w(x, y) + \sum_{i=2}^k w(x_i^*, y)}\right\}\end{aligned}$$

$$
\begin{aligned}
&\times \tilde{T}(y; x_2^*, \ldots, x_k^* \mid x) dy_2 \ldots dy_k dx_2^* \ldots dx_k^* \\
&= k \frac{w(y,x)w(x,y)}{\lambda(y,x)} \\
&\quad \times \int \min \left\{ \frac{1}{w(y,x) + \sum_{i=2}^k w(y_i, x)}, \right. \\
&\quad \left. \frac{1}{w(x,y) + \sum_{i=2}^k w(x_i^*, y)} \right\} \\
&\quad \times \tilde{T}(y; x_2^*, \ldots, x_k^* \mid x) \tilde{T}(x; y_2, \ldots, y_k \mid y) \\
&\quad dy_2 \times \ldots dy_k dx_2^* \ldots dx_k^* = \pi(y) A(y,x).
\end{aligned}
$$

In the above derivation we have used $\tilde{T}(x; y, y_2, \ldots, y_k) = T(x; y)\tilde{T}(x; y_2, \ldots, y_k \mid y)$. □

3 Correlated proposals

An open question that we try to answer here is what type of correlation between proposals will result in improvement in efficiency over the original MTM. To simplify the exposition, consider first the situation in which the proposals are exchangeable univariate random variables, say $Y_1, \ldots, Y_k$ with distribution function F. Without loss of generality we can assume that $E[Y_i] = 0$. Intuitively, we would like the proposals to be "well distributed" in the sample space. There is not a single comprehensive mathematical definition of what we mean by "well distributed" but two possible approaches can be outlined. First, one could consider proposals that are, on average, as far away from one another as possible with respect to a particular distance. If we consider the Euclidean distance $d(Y_i, Y_j) = \sqrt{(Y_i - Y_j)^2}$, then we need to consider the pairwise correlation between proposals since $E[d^2(Y_i, Y_j)] = 2\sigma^2(1-\rho)$ where $\sigma^2 = \text{Var}(Y_i)$ for all i and $\rho = \text{corr}(Y_i, Y_j)$ for all $i \neq j$. Therefore, the largest distance is achieved on average by proposals that are *extremely antithetic (EA)* (Craiu and Meng, 2005), i.e. they achieve the smallest possible pairwise correlation $\rho = \text{corr}(Y_i, Y_j)$, subject to the constraint that the random variables $Y_1, \ldots, Y_k$ are exchangeable and marginally distributed with distribution function F. However, having a larger distance between proposals is not always the most efficient implementation of the MCTM. An alternative is to stratify the sample of proposals. In this case of interest is the stratification of the proposals in the sample space. In recent years the literature on quasi-Monte Carlo algorithms has explored a wide variety of methods for producing stratified samples that are equidistributed in the unit hypercube (e.g. L'Ecuyer and Lemieux, 2002) and we investigate some of these techniques in the context of MCTM.

3.1 Extremely antithetic proposals

We limit our discussion of the antithetic approach to MCTM to the situation in which the proposals are multivariate normals and the MTM is implemented within a Random-Walk Metropolis algorithm. This is one of the most common instances in which the Metropolis-Hastings is used when the stationary distribution of interest is multivariate. Proposition 2.1 is particularly attractive in the case of Gaussian proposals since the conditional kernel is easy to compute and to sample from.

More precisely, consider an r-dimensional sample space for the Markov chain X_t constructed via MTM with multivariate Gaussian proposals. More specifically, the original MTM algorithm generates k proposals from $N_r(\tilde{x}, \Sigma)$ whenever the current state is $\tilde{x}$. A general version of the original MTM uses at each step k proposals which are jointly normal from $N_{kr}(\tilde{x}_k, \Sigma_{kr})$. To simplify the notation we assume that $r = 2$ but the discussion is true in general.

If the independent proposals are sampled from $N_2((x,y)^T, \Sigma)$, then a pair of correlated proposals is

$$
(x_1, y_1, x_2, y_2)^T \sim N_4\left((x, y, x, y)^T, \begin{pmatrix} \Sigma & \Psi \\ \Psi & \Sigma \end{pmatrix}\right),
$$

where $\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$ and $\Psi = \begin{pmatrix} \rho_1\sigma_1^2 & \rho_2\sigma_1\sigma_2 \\ \rho_2\sigma_1\sigma_2 & \rho_1\sigma_2^2 \end{pmatrix}$. We seek a correlation structure, as determined by (ρ_1, ρ_2), so that the average Euclidean distance between proposals is maximized. It can be assumed without loss of generality that Σ is diagonal, say $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2)$. Otherwise one can apply an orthogonal transformation $(x_1', y_1')^T = C(x_1, y_1)^T$ and $(x_2', y_2')^T = C(x_2, y_2)^T$ so that, if $d((x,y),(x',y')) = \sqrt{(x-x')^2 + (y-y')^2}$, then $d((x_1', y_1'), (x_2', y_2')) = d((x_1, y_1), (x_2, y_2))$ and x_i' is independent of y_i'. The marginal distribution of

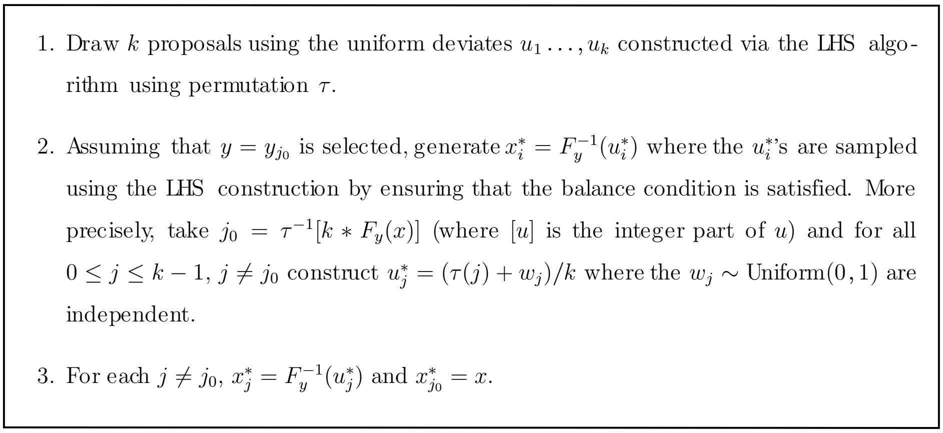
$$
\begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \end{pmatrix} \sim N\left((0,0)^T, \begin{pmatrix} 2\sigma_1^2 - 2\rho_1\sigma_1^2 & -2\rho_2\sigma_1\sigma_2 \\ -2\rho_2\sigma_1\sigma_2 & 2\sigma_2^2 - 2\rho_1\sigma_2^2 \end{pmatrix}\right).
$$

Therefore

$$
E[d((x_1, y_1), (x_2, y_2))] = 2\left(\sigma_1^2 + \sigma_2^2\right)(1 - \rho_1)
$$

is maximized when ρ_1 is equal to its smallest possible value. In our experience, the choice of ρ_2 does not influence the efficiency of the MCTM. In addition, any choice different than $\rho_2 = 0$ increases the complexity of the constraint on ρ_1 since the covariance matrix of *all* the proposals must be

Fig. 1 MCTM with LHS proposals



1. Draw k proposals using the uniform deviates $u_1 \ldots, u_k$ constructed via the LHS algorithm using permutation τ.
2. Assuming that $y = y_{j_0}$ is selected, generate $x_i^* = F_y^{-1}(u_i^*)$ where the u_i^*'s are sampled using the LHS construction by ensuring that the balance condition is satisfied. More precisely, take $j_0 = \tau^{-1}[k * F_y(x)]$ (where $[u]$ is the integer part of u) and for all $0 \le j \le k-1$, $j \ne j_0$ construct $u_j^* = (\tau(j) + w_j)/k$ where the $w_j \sim \text{Uniform}(0,1)$ are independent.
3. For each $j \ne j_0$, $x_j^* = F_y^{-1}(u_j^*)$ and $x_{j_0}^* = x$.

positive definite. In our applications we used $\rho_2 = 0$. Therefore, for the MCTM with Gaussian proposals, $y_i \sim N_r(\tilde{x}, \Sigma)$, one can use $(y_1^T, \ldots, y_k^T)^T \sim N((\tilde{x}^T, \tilde{x}^T, \ldots, \tilde{x}^T)^T, \Sigma_{kr})$ with

$$\Sigma_{kr} = \begin{pmatrix} \Sigma & \Psi & \ldots & \Psi \\ \Psi & \Sigma & \Psi & \Psi \\ \ldots & \ldots & \ldots & \ldots \\ \Psi & \Psi & \Psi & \Sigma \end{pmatrix}$$

where $\Psi = \text{diag}(\rho\sigma_1^2, \ldots, \rho\sigma_r^2) \in R^{r \times r}$ and $\rho = -1/(k-1)$. The lower bound $\rho = -\frac{1}{k-1}$ is obtained from the constraint that the joint correlation matrix of *all* the proposals, Σ_{kr}, is a positive semi-definite matrix.

3.2 Quasi Monte Carlo proposals

A situation in which it is straightforward to implement MCTM is one in which the proposals are univariate and can be generated using the inverse cumulative distribution function. In such a case the stratified sample of proposals can be obtained from a stratified sample in the unit interval. One of the most widely used techniques to obtain the latter is the Latin Hypercube Sampling (LHS) which has been introduced by McKay et al. (1979) and has been studied intensively in the literature ever since (see also Stein, 1987; Owen, 1992; Loh, 1996; Craiu and Meng, 2005). The generation of k uniform variates via LHS involves the following three steps.

Step I Generate independently $v_1, \ldots v_k \sim \text{Uniform}(0, 1)$.
Step II Select a random permutation τ of $\{0, \ldots, k-1\}$.
Step III Construct $u_i = (v_i + \tau(i))/k$, for all $1 \le i \le k$.

It can be noticed that the LHS adds little computational overhead when compared to the independent generation of samples. In addition, there is no requirement for a symmetric distribution of the proposals. It is easy to use LHS within MCTM as described in Fig. 1.

While the LHS method can be extended to generation of multivariate uniforms, a better way of producing a correlated set of proposals is to use a *randomized quasi Monte Carlo* (RQMC) method. These methods are often used in the context of high-dimensional numerical integration to provide more accurate estimators than the Monte Carlo method, which corresponds to using independent sampling. When not randomized, QMC methods offer a deterministic approximation that can be proved to be asymptotically better than Monte Carlo for some classes of functions (Niederreiter, 1992), but their performance on specific problems with a fixed sample size is difficult to assess because no error estimate comes with them. Their randomized versions, RQMC methods, avoid this problem by using the following idea: suppose we want to generate a sample $y_1, y_2, \ldots, y_k$ with the requirement that each y_i has a given marginal distribution. (Note that we do not specify what should be the joint distribution of the sample, which means the y_i's could be independent or correlated.) Assume the sampling space for each y_i has dimension r, and that we have a function $G\colon [0, 1)^r \to \mathbb{R}^r$ such that for a random vector $\mathbf{u}$ uniformly distributed over $[0, 1)^r$, $G(\mathbf{u})$ has the desired distribution for y_i. In other words, $G(\cdot)$ represents the transformation used to generate observations y_i's having the desired distribution. Now, let $P_k = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ be a deterministic highly-uniform point set such as those used by QMC methods, and assume $\tilde{P}_k = \{\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_k\}$ is a randomized version of P_k such that (i) each $\tilde{\mathbf{u}}_i$ is uniformly distributed over $[0, 1)^r$, and (ii) $\tilde{P}_k$ has the same highly-uniform properties as P_k (examples of such constructions are given below). Then one can generate the sample $y_1, \ldots, y_k$ by letting $y_i = G(\mathbf{u}_i)$, $i = 1, \ldots, k$ and in this way, each y_i has the desired distribution, and the structure of P_k induces correlation among the y_i's.

For a real-valued function f, the sample $y_1, \ldots, y_k$ thus obtained can be used to estimate $\mu = E(f(Y))$ via the

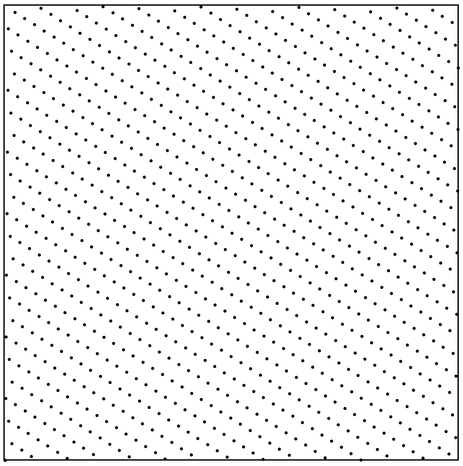


Fig. 2 Two-dimensional Korobov rule with $k = 1024$ and $a = 139$

unbiased estimator

$$\hat{\mu}_{\text{RQMC}} = \frac{1}{k}\sum_{i=1}^{k} f(y_i),$$

whose variance can be estimated by generating m independent randomizations of P_k.

Before we explain how to use RQMC methods in the context of the MCTM algorithm, let us give an example illustrating how the above procedure can be applied in practice. Suppose each y_i is multinormal. More precisely, assume our goal is to have $y_i \sim N((\mu_1, \ldots, \mu_r)^T, \Sigma)$, where Σ is an $r \times r$ covariance matrix that we assume can be written as $\Sigma = AA^T$, with A a lower-triangular matrix. Let $\Phi(\cdot)$ be the CDF of a standard normal random variable. If $\mathbf{u} = (u_1, \ldots, u_r)^T$ is uniformly distributed over $[0, 1)^r$, then $y_i = (y_{i,1}, \ldots, y_{i,r})^T$ can be obtained as follows:

$$\begin{pmatrix} y_{i,1} \\ \vdots \\ y_{i,r} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_r \end{pmatrix} + A \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_r) \end{pmatrix}. \tag{3.1}$$

In other words, the function $G(\mathbf{u})$ in this case is given by the right-hand side of (3.1).

For the highly-uniform point set, let us use a *Korobov rule*, which is defined as follows: choose an integer $a \in \{1, \ldots, k-1\}$, and let

$$P_k = \left\{ \frac{i-1}{k}(1, a, \ldots, a^{r-1}) \bmod 1, i = 1, \ldots, k \right\}.$$

Figure 2 gives an example of a Korobov rule with $k = 1024$ and $a = 139$. As suggested in Cranley and Patterson (1976), this type of point set can be randomized by generating a random vector $\mathbf{v}$ uniformly in $[0, 1)^r$, and adding it to each point of P_k (modulo 1). That is, let $\tilde{P}_k = \{\tilde{\mathbf{u}}_i, i = 1, \ldots, k\}$, where

$$\tilde{\mathbf{u}}_i = (\mathbf{u}_i + \mathbf{v}) \bmod 1.$$

Putting everything together, the sample $y_1, \ldots, y_k$ can be generated by a randomly-shifted Korobov rule as follows:

1. Generate a uniform vector $\mathbf{v}$ over $[0, 1)^r$.
2. For each $i = 1, \ldots, k$:
   (a) Let $\mathbf{u}_i = (i-1)(1, a, \ldots, a^{r-1})/k \bmod 1$, and $\tilde{\mathbf{u}}_i = (\mathbf{u}_i + \mathbf{v}) \bmod 1$.
   (b) Let $y_i = G(\tilde{\mathbf{u}}_i)$, where $G(\cdot)$ is defined on the right-hand side of (3.1).

In the context of the MCTM algorithm, RQMC methods can be used in a way that mimics the LHS implementation described at the beginning of this section. In Fig. 3, we describe an implementation based on a randomly-shifted highly-uniform point set (not necessarily a Korobov rule). In what follows, we assume $G_x(\cdot)$ is such that $y = G_x(\mathbf{u})$ has distribution $T(x; y)$ for a uniform $\mathbf{u}$, and $G_y(\cdot)$ is such that $x = G_y(\mathbf{u})$ has distribution $T(y; x)$.

In other words, once we have our set of trial proposals $y_1, \ldots, y_k$ and one of them, y, is chosen, we fix x_1^* to be the current state x, find the shift $\mathbf{w}$ such that the first point (corresponding to the origin) of a point set randomized by that shift $\mathbf{w}$ would have generated x, and then we generate the other x^*'s using the remaining points of that point set. By doing so, the joint distribution of $x_1^*, \ldots, x_k^*$ given y is the same as the joint distribution of $y_1, \ldots, y_k$ given x, which is required for the balance condition to hold. Note that technically, for this method to produce exchangeable proposals, one should first randomly permute the order of the points in $\tilde{P}_k$. However, as explained in the next proposition, in practice this is not necessary since the order in which the proposals y_i are produced is not important.

Proposition 3.1. *Let MCTM1 be a version of the MCTM algorithm based on an unpermuted randomly-shifted point set, and MCTM2 be based on a randomly permuted version of the same randomly-shifted point set. Then for a given input x, MCTM1 and MCTM2 can be implemented so that they produce the same output y.*

Proof: First, note that under MCTM2, the proposals are exchangeable. Let $y_i = G_x(\tilde{\mathbf{u}}_i)$ for $i = 1, \ldots, k$ and let $z_i = G_x(\tilde{\mathbf{u}}_{\pi(i)})$, $i = 1, \ldots, k$, where π is a random permutation of $[1, \ldots, k]$. So the y_i are the proposals used in MCTM1 and the z_i are the ones used in MCTM2. Note that the samples $\{y_1, \ldots, y_k\}$ and $\{z_1, \ldots, z_k\}$ are the same.

Fig. 3 MCTM with QMC proposals

```
1. Draw $k$ trial proposals $y_1$ ,..., $y_k$ using a randomly-shifted point set $P_k$. More precisely,
   let $y_i = G_x(\mathrm{u}_i)$, for $i = 1$ ,..., $k$. Compute $w(y_j, x) = \pi(y_j)T(y_j; x)$.
2. Select $y = y_j$ among the $k$ tri als with probability proportbnal to $w(y_j, x)$.
3. Let $x_1^* = x$ and find w such that $G_y(\mathrm{w}) = x$.
4. Set $x_j^* = G_y((\mathrm{u}_j + \mathrm{w}) \bmod 1)$ for $j > 1$.
```

Let us introduce some notation: for $i = 1, \ldots, k$, let

$$w_i = \frac{w(y_i, x)}{\sum_{i=1}^{k} w(y_i, x)}, \quad W_i = \sum_{j=1}^{i} w_j, \quad \text{and} \quad W_0 = 0.$$

Note that $w_i = w(z_{\pi^{-1}(i)}, x)$.

Now suppose that in MCTM1, we choose y_{j_0} as our proposal using the following procedure: generate $U \sim U(0, 1)$, let j_0 be such that $W_{j_0 - 1} < U \leq W_{j_0}$. For MCTM2, assume that z_{i_0} is chosen as follows: generate $U \sim U(0, 1)$, let l_0 be such that $W_{l_0 - 1} < U \leq W_{l_0}$, and then let $i_0 = \pi^{-1}(l_0)$. In other words, in MCTM2, the bins used to choose the index i_0 are ordered according to the unpermuted sample $\{y_1, \ldots, y_k\}$. It is clear that in MCTM1, the probability of choosing index I is proportional to $w_I = w(y_I, x)$, and for MCTM2, this probability is proportional to $w_{\pi(I)} = w(y_{\pi(I)}, x) = w(z_I, x)$, as desired. We can also see that for a given value u for U, $l_0 = j_0$ above and so if MCTM1 chooses y_{j_0}, then MCTM2 chooses the sample point $z_{i_0} = z_{\pi^{-1}(j_0)} = y_{j_0}$, i.e., both implementations choose the same y.

For the rest of the MCTM step, if we assume that the first point $\tilde{\mathbf{u}}_1$ in the randomly-shifted point set corresponds to the non-randomized point $\mathbf{u}_1 = \mathbf{0}$, then the only difference between MCTM1 and MCTM2 is that in the latter, we would set $x^*_{\pi^{-1}(1)} = x$ and let $x_j^* = G_y((\mathbf{u}_{\pi(j)} + \mathbf{w}) \bmod 1)$ for $j \neq \pi^{-1}(1)$, instead of having $x_1^* = x$ and $x_j^* = G_y((\mathbf{u}_j + \mathbf{w}) \bmod 1)$. Hence for a given x and y (which are the same for MCTM1 and MCTM2), the sample $\{x_1^*, \ldots, x_k^*\}$ will be the same under MCTM1 and MCTM2, which means the probability r_g of acceptance is the same under both approaches. Hence if we use the same uniform $V \sim U(0, 1)$ in both MCTM1 and MCTM2 to decide whether we keep y or not (based on whether $V \leq r_g$ or not), then the decision of keeping y or not will be the same under both MCTM1 and MCTM2.

Note that similar arguments can be used to show that, in practice, it is not necessary to randomly permute the order of the points in the one-dimensional LHS approach outlined at the beginning of this section. □

3.2.1 *Transformations over the unit hypercube*

As we will see in Section 4.2, for some problems it might be helpful to transform the points of the randomized QMC point set to be used in the MCTM algorithm before generating the QMC proposals. The intuition here is that in some cases, we may be interested in having more points in some regions of the unit hypercube. For instance, if the proposals are multivariate gaussian variables, then perhaps we would like to generate more proposals in the tails of the gaussian distribution, which means we would like to have more points in the "corners" $(0, \ldots, 0)$ and $(1, \ldots, 1)$ of the unit hypercube. Here, we explain how this can be achieved in the context of the MCTM algorithm.

Suppose $g : [0, 1] \rightarrow [0, 1]$ is a bijection, and let $g(\mathbf{u}) = (g(u_1), \ldots, g(u_r))$. Now let $Q_k = g(\tilde{P}_k) = \{g(\tilde{\mathbf{u}}_1), \ldots, g(\tilde{\mathbf{u}}_k)\}$. That is, Q_k is the point set obtained after applying g to each point of a randomized point set $\tilde{P}_k$. Let $\hat{T}(x; y)$ be the probability density function of $G_x(y)$, where $y = g(\mathbf{u})$. Similarly, $\hat{T}(y; x)$ is the density function of $G_y(x)$. Note that since we assumed that $g(\cdot)$ was a bijection, $\hat{T}(\cdot; \cdot)$ is indeed a probability density function.

If we assume that the ratio

$$\ell(x; y) = \frac{T(x; y)}{\hat{T}(x; y)}$$

is symmetric in x and y, then we can set $\lambda(x, y) = \ell(x; y)$ in the MCTM algorithm based on the point set Q_k, which is then performed as shown in Fig. 4.

In this algorithm, we choose the shift $\hat{\mathbf{w}}$ so that if we had used the first point (corresponding to the origin) of P_k to generate x_1^*, then after the shift *and* the transformation, we would have obtained x, i.e., $G_y(g(\hat{\mathbf{w}})) = x$. Also, the reason why we used $\lambda(x, y) = \ell(x; y)$ is two-fold: first, doing so prevents us from having to evaluate $\hat{T}(y; x)$, which typically is harder to compute than $T(y; x)$; second, numerical experiments suggested that when using this type of transformation, better results were obtained by choosing this $\lambda(x, y)$ instead of just taking $\lambda(x, y) = 1$.

Fig. 4 MCTM with QMC proposals to which a transformation has been applied

1. Draw k trial proposals $y_1, \dots, y_k$ using the randomized (and transformed) point set Q_k. More precisely, let $y_i = G_x(g(\tilde{\mathbf{u}}_i))$, for $i = 1, \dots, k$, where $\tilde{P}_k = \{\tilde{\mathbf{u}}_i, i = 1, \dots, k\}$ is a randomly-shifted QMC point set. Compute $w(y_j, x) = \pi(y_j)\hat{T}(y_j; x)\lambda(y_j, x) = \pi(y_j)T(y_j; x)$.
2. Select $y = y_j$ among the k trials with probability proportional to $w(y_j, x)$.
3. Let $x_1^* = x$ and find $\mathbf{w}$ such that $G_y(\mathbf{w}) = x$. Let $\hat{\mathbf{w}} = g^{-1}(\mathbf{w})$.
4. Set $x_j^* = G_y(g((\mathbf{u}_j + \mathbf{w}) \bmod 1))$ for $j > 1$, where $P_k = \{\mathbf{u}_i, i = 1, \dots, k\}$ is the unrandomized QMC point set.

Example 3.1. In Section 4.2, we will be using the transformation

$$g(u) = \frac{\sin((u - 0.5)\pi) + 1}{2} \tag{3.2}$$

illustrated on Fig. 5.

One can easily verify that $g(\cdot)$ is a bijection with the following properties: $g(0) = 0$, $g(1) = 1$, $g(1/2) = 1/2$, and $g(u) + g(1 - u) = 1$ for any $u \in (0, 1)$.

These properties imply that, if $U \sim$ Uniform(0, 1) then the density function of $\Phi^{-1}(g(U))$ is symmetric around 0—just like that of a standard $N(0, 1)$ given by $\Phi^{-1}(U)$—but with fatter tails than the $N(0, 1)$ distribution.

The inverse of g is given by

$$g^{-1}(z) = \frac{\arcsin(2z - 1)}{\pi} + 0.5,$$

and we can show that if $Z_j = \Phi^{-1}(g(U_j))$, $j = 1, \dots, r$, where the U_j's are i.i.d. Uniform(0, 1), then $Z_1, \dots, Z_r$ has joint density

$$f_{Z_1,\dots,Z_r}(z_1, \dots, z_r) = \prod_{j=1}^{r} \frac{e^{-z_j^2/2}}{\sqrt{2\pi}} \frac{1}{\pi\sqrt{\Phi(z_j)(1 - \Phi(z_j))}}.$$

Thus if $T(x; y)$ is a multivariate gaussian with no correlation (as in Section 4.2) given by

$$T(x; y) = \prod_{j=1}^{r} \frac{e^{-(y_j - x_j)^2/2\sigma_j^2}}{\sqrt{2\pi\sigma_j^2}}$$

then

$$\hat{T}(x; y) = \prod_{j=1}^{r} \frac{e^{-(y_j - x_j)^2/2\sigma_j^2}}{\sqrt{2\pi\sigma_j^2}} \frac{1}{\pi\sqrt{\Phi(\frac{y_j - x_j}{\sigma_j})(1 - \Phi(\frac{y_j - x_j}{\sigma_j}))}}$$

and thus

$$\ell(x; y) = \frac{T(x; y)}{\hat{T}(x; y)} = \prod_{j=1}^{r} \frac{1}{\pi\sqrt{\Phi(\frac{y_j - x_j}{\sigma_j})(1 - \Phi(\frac{y_j - x_j}{\sigma_j}))}}$$

$$= \prod_{j=1}^{r} \frac{1}{\pi\sqrt{(1 - \Phi(\frac{x_j - y_j}{\sigma_j}))\Phi(\frac{x_j - y_j}{\sigma_j})}},$$

which means $\ell(x; y)$ is symmetric, as required.

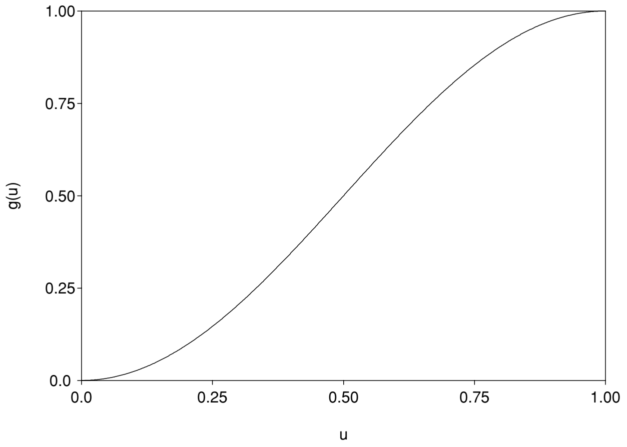


Fig. 5 Transformation $g(u)$ used in Section 4.2

One may wonder whether the transformation defined by (3.2) is optimal in its class. More precisely, suppose we are interested in studying a general transformation of the form

$$f_\alpha(u) = \frac{\sin[(u^\alpha - 1/2)\pi] + 1}{2}.$$

If $Z_\alpha = \Phi^{-1}(f_\alpha(U))$, with $U \sim$ Uniform(0, 1), then

$$P(Z_\alpha \le t) = \left[\frac{\arcsin(2\Phi(t) - 1)}{\pi} + \frac{1}{2}\right]^{1/\alpha}$$

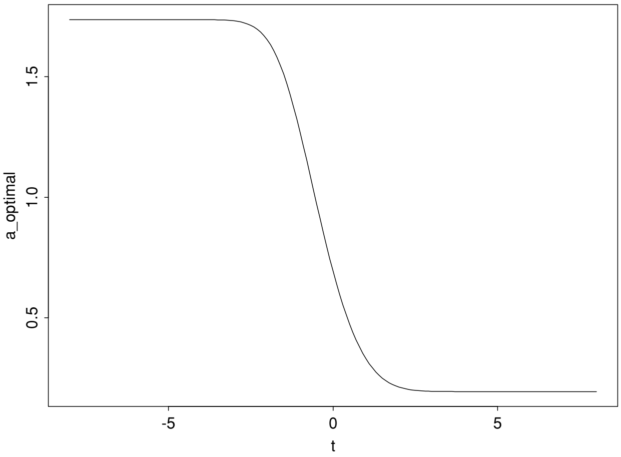


Fig. 6 Plot of $\alpha_{opt}(t)$

has density $g_\alpha(z) = \frac{d\,P(Z_\alpha \leq z)}{d\,z}$. Unless $\alpha = 1$ the density g_α is not symmetric. One can study the tail probability of g_α in comparison to the tail probability of g_1 by looking at the ratio $g_\alpha(z)/g_1(z)$ for large values of z. It turns out that

$$\frac{g_\alpha(t)}{g_1(t)} = \frac{1}{\alpha}\left[\frac{\arcsin(2\Phi(t)-1)}{\pi} + \frac{1}{2}\right]^{\frac{1-\alpha}{\alpha}}.$$

Maximizing the previous ratio with respect to α yields:

$$\alpha_{opt}(t) = -\log\left[\arcsin\frac{2\Phi(t)-1}{\pi} + 1/2\right].$$

The function $\alpha_{opt}(t)$ is plotted in Fig. 6 for various values of t. It is seen that for $|t|$ moderately large the value of α_{opt} seem to stabilize around $\lim_{t\to\infty}\alpha_{opt}(t) \approx 0.1937$ and $\lim_{t\to-\infty}\alpha_{opt}(t) \approx 1.737$. In our applications we will use the symmetric g_α given by $\alpha = 1$.

4 Examples

We examine three instances for which the performance of the MCTM is compared to MTM. In all examples shown the MTM is doing better than the classical Metropolis-Hastings algorithm with only one proposal.

4.1 MCTM for local search MCMC

We begin with a simple example in which the MCTM algorithm is used with univariate proposals in combination with a random-ray Monte Carlo algorithm. LLW have shown that MTM can be used within the random-ray Monte Carlo, the hit-and-run algorithm (Chen and Schmeiser, 1993) or the Adaptive Direction Sampling algorithm (Gilks et al., 1994). The random-ray Monte Carlo is a modified form of the hit-and-run algorithm and is especially effective when the distribution of interest is multimodal and the modes are aligned on a direction which is not parallel to any of the coordinate axes. We consider here one target density from a bimodal family of bivariate distributions constructed by Gelman and Meng (1991). More precisely, the density

$$f(x_1,x_2) \propto \exp\left\{-\left(9x_1^2x_2^2 + x_1^2 + x_2^2 - 8x_1 - 8x_2\right)/2\right\} \tag{4.1}$$

has the property that the two conditional densities $f(x_1|x_2)$ and $f(x_2|x_1)$ are normal but the joint density is not normal. A three-dimensional plot of the density $f(x_1,x_2)$ is shown in Fig. 7.

The construction of the random-ray Monte Carlo via MTM has been detailed by LLW and is followed here. Specifically, at each iteration t of the algorithm, a random direction, say $\mathbf{e}$, is generated and then, along direction $\mathbf{e}$, the proposals $y_1,\ldots,y_k$ are generated from the distribution $T_{\mathbf{e}}(x;y)$ where x is the state of the chain at time t. The proposals are generated using $y_i = x + r_i\mathbf{e}$ where $r_1,\ldots,r_k$ are sampled from Uniform$[-\sigma,\sigma]$. In our implementation of MCTM, we use k antithetic variates r_i. The parameters chosen here are $\sigma \in \{3,4,5\}$ and $k \in \{3,4,5,6\}$. Due to the stratification induced by the hypercube sampling the MCTM has a higher acceptance rate and thus mixes better than the original MTM.

Table 1 offers support to the previous observations. We report the Monte Carlo MSE reduction factor, R, for different choices of σ and k. In each case we perform 1000 updates with each algorithm and we replicate the analysis 500 times. The starting points are the same for the two algorithms. The numbers reported in each cell represent the estimates of R. The true marginal mean of X can be computed via numerical integration and is approximately equal to 1.83. In this example, the acceptance rates are different for the MTM and the MCTM so are also reported in Table 2.

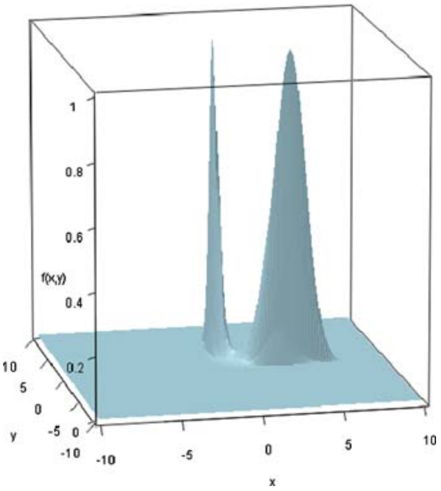


Fig. 7 Random-ray Monte Carlo. The bivariate density $f(x_1,x_2)$

Table 1 Values of the MSE reduction factor $R = \frac{\text{MSE}_{\text{anti}}}{\text{MSE}_{\text{ind}}}$

$\sigma \backslash k$	3	4	5	6
3	0.35	0.53	0.64	0.81
4	0.31	0.42	0.58	0.76
5	0.29	0.40	0.49	0.62

Table 2 Probability of acceptance for MTM/MCTM

$\sigma \backslash k$	3	4	5	6
3	26.5/46.1	31.2/47.8	35.2/50.3	38.7/49.7
4	24.5/40.9	26.6/41.8	29.8/44.5	32.3/46.2
5	18.8/35.4	22.7/37.6	26.2/40.3	29.4/42.4

4.2 Lupus data

The random-walk Metropolis is a useful method in multivariate settings in which the information about the stationary distribution is not concrete enough to help us build a good proposal distribution. We apply our method to data from van Dyk and Meng (2001) consisting of measurements on 55 patients of which 18 have been diagnosed with latent membranous lupus. Table 3 shows the data with two clinical covariates, IgA and IgG, that measure the levels of immunoglobulin of type A and of type G, respectively. Of interest is the prediction of disease occurrence using the two covariates IgG3-IgG4 and IgA. We consider a logit regression model in which

$$logit\ P(Y_i = 1) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i},$$

where $X_i^T = (1, X_{i1}, X_{2i})$ is the vector of covariates for the i-th individual. We follow Tan (2006) and consider that the prior for $\beta = (\beta_0, \beta_1, \beta_2)^T$ is trivariate normal with zero mean and variance diag$(100^2, 100^2, 100^2)$. The posterior density is then proportional to

$$\pi(\beta|x, y) \propto \prod_{j=0}^{2} \frac{e^{-0.5\beta_j/100^2}}{100\sqrt{2\pi}}$$
$$\times \prod_{i=1}^{55} \left[\frac{\exp(X_i^T \beta)}{1 + \exp(X_i^T \beta)}\right]^{y_i} \left[\frac{1}{1 + \exp(X_i^T \beta)}\right]^{1-y_i}.$$

Table 3 The number of latent membranous lupus nephritis cases, the numerator, and the total number of cases, the denominator, for each combination of the values of the two covariates

	IgA				
IgG3-IgG4	0	0.5	1	1.5	2
−3.0	0/ 1	–	–	–	–
−2.5	0/ 3	–	–	–	–
−2.0	0/ 7	–	–	–	0/ 1
−1.5	0/ 6	0/ 1	–	–	–
−1.0	0/ 6	0/ 1	0/ 1	–	0/ 1
−0.5	0/ 4	–	–	1/ 1	–
0	0/ 3	–	0/ 1	1/ 1	–
0.5	3/ 4	–	1/ 1	1/ 1	1/ 1
1.0	1/ 1	–	1/ 1	1/ 1	4/ 4
1.5	1/ 1	–	–	2/ 2	–

The random walk Metropolis is used with multiple proposals, antithetic, QMC, and independent. The proposal $T(\cdot|\beta)$ is trivariate normal with mean β and variance $\Sigma = \text{diag}(\sigma^2, \sigma^2, \sigma^2)$. All chains are started from $\beta = 0$.

In Table 4 we report, for β_1 and $p_{25} = 1_{\{\beta_1 > 25\}}$, the ratios $R = \frac{\text{MSE}_{\text{anti}}}{\text{MSE}_{\text{ind}}}$ and $R = \frac{\text{MSE}_{\text{qmc}}}{\text{MSE}_{\text{ind}}}$, where MSE represents the Monte Carlo mean squared error, and the index refers to the method of generating the proposals, i.e., independently, antithetically or with QMC sampling. To be more specific, we replicated $M = 5000$ samples, each of $N = 1000$ draws. If we denote by b_{ij} the jth sample point drawn in the ith replicate from the posterior distribution of β_1 then, using $\bar{b}_{\cdot\cdot} = \frac{\sum_{ij} b_{ij}}{MN}$ and $\bar{b}_{i\cdot} = \frac{\sum_j b_{ij}}{N}$ for all $i = 1, \dots, M$ the MSE is defined as

$$\text{MSE} = (\bar{b}_{\cdot\cdot} - E[\beta_1|\text{data}])^2 + \frac{\sum_i (\bar{b}_{i\cdot} - \bar{b}_{\cdot\cdot})^2}{(M - 1)}.$$

Similar calculations can be done for p_{25}. Numerical integration yields $E[\beta_1|\text{data}] \approx 13.57$ and $E[p_{25}|\text{data}] \approx 0.073$ (see Tan, 2006).

The QMC sampling is based on a randomly-shifted Korobov point set to which the transformation described in Example 3.1 has been applied. Note that while inversion of the normal CDF is used to generate the QMC proposals, both independent and antithetic proposals use instead Marsaglia's polar method (Marsaglia, 1962) to generate normal variates.

It is seen that the use of antithetic proposals is more effective when the number of streams is average ($k = 8$). But the larger savings are obtained with the QMC stratified samples. When the number of proposals is very large ($k = 16$) the benefit of antithetic or stratified proposals diminishes as the independent MTM has already very good properties. However, in practical applications one may not have the computational power to generate a large number of proposals so the improvement brought in by the MCTM can be important.

On the root scale, the reduction in RMSE obtained with QMC correspond to savings between 20–25%. In none of the situations explored has the use of antithetic proposals been inflating the MSE. For the QMC proposals, we only give results for $k = 8$ and $k = 16$ since smaller values of k make it difficult for the high-uniformity of the QMC point set to be significant. However, we see that for those values,

Table 4 Values of R for β_1/p_{25} in the logit example

	Antithetic			QMC		
$k\backslash\sigma$	2	3	4	2	3	4
3	0.92/0.92	0.90/0.86	0.99/0.95	–	–	–
4	0.94/0.87	0.88/0.88	0.91/0.89	–	–	–
5	0.98/0.96	0.81/0.81	0.89/0.86	–	–	–
6	0.91/0.86	0.86/0.78	0.95/0.92	–	–	–
8	0.81/0.70	0.75/0.69	0.83/0.80	0.69/0.72	0.61/0.60	0.59/0.56
16	0.87/0.81	0.97/0.94	0.91/0.88	0.81/0.81	0.82/0.84	0.76/0.75

the MSE reductions are quite good, with values below 0.6 in some cases, and never much more than 0.8. We should also point out that the transformation we used for the QMC proposals has the effect of making the acceptance rate smaller in this case than with independent proposals. When using QMC *without* the transformation, we get larger acceptance rates than for independent sampling, but the MSE reduction factors are not as good as when the transformation is used. Antithetic proposals give acceptance rates that are about the same as for independent proposals.

In terms of computation time, the MCTM and MTM are about the same: for instance, when $k = 8$, to run 100 replications of 1000 draws on a 2.0 GHz Pentium 4-M laptop computer, the MCTM based on the Korobov point set needs about 32 seconds, while the corresponding MTM version requires 34 seconds. Results are reversed for the antithetic implementation, which needs about two seconds more than the MTM.

4.3 Orange tree data

We consider data on the growth of orange trees over time which was originally discussed by Draper and Smith (1981) and later on by Lindstrom and Bates (1990). The data shown in Fig. 8 consists of circumference measurements Y_{ij} made for tree i at time x_j, $i = 1, \dots, 7$ and $j = 1, \dots, 5$. We consider the logistic growth model in which $Y_{ij} \sim N(\mu_{ij}, \sigma_c^2)$ where

$$\mu_{ij} = \frac{\exp(\theta_{i1})}{1 + (\exp(\theta_{i2}) - 1)\exp(-\exp(\theta_{i3})x_j)},$$

for $i = 1, \dots, 5$ and $j = 1, \dots, 7$.

A priori the parameters $(\theta_{11}, \dots, \theta_{53})$ are independent and identically distributed as Gaussian with mean zero and standard deviation $\sigma = 10$. For these experiments, we used an inverse gamma prior with parameters (0.001,0.001) for σ_c^2.

We implement a random walk Metropolis in which the proposals for θ are drawn from a multivariate normal centered at the current state and the proposals for σ_c are drawn independently from the proposal distribution which, in turn, does not depend on the current state. This corresponds to a realistic scenario in which one cannot generate correlated proposals for all the components of the Markov chain. For the stratified (QMC) sampling, we used a 15-dimensional randomly-shifted Sobol' point set (Sobol', 1967) to construct proposals for $\{\theta_{i1}, \theta_{i2}, \theta_{i3}, i = 1, \dots, 5\}$. To illustrate

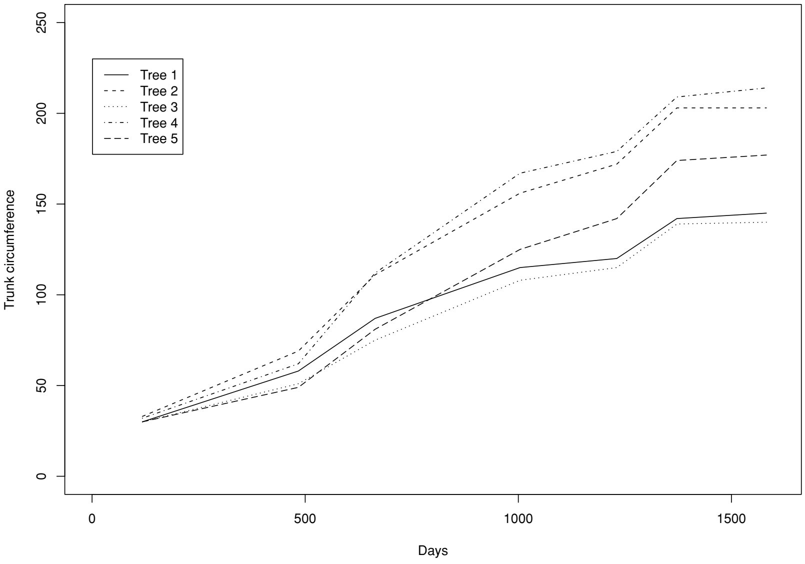


Fig. 8 Trunk circumference (in millimeters) of five orange trees data over up to 1600 days

Table 5 Values of the MSE and variance (in parentheses) reduction factor $R = \frac{\mathrm{MSE}_{\mathrm{MCTM}}}{\mathrm{MSE}_{\mathrm{MTM}}}$

$\theta \backslash k$	4	6	8
θ_{31}	0.99 (0.79)	0.99 (1.00)	0.98 (0.69)
θ_{32}	0.97 (1.01)	1.08 (1.06)	0.96 (0.90)
θ_{33}	0.98 (0.86)	0.83 (0.80)	1.01 (1.04)
θ_{41}	0.95 (0.86)	0.89 (0.84)	0.71 (0.57)
θ_{42}	0.83 (0.80)	0.80 (0.87)	0.88 (0.87)
θ_{43}	1.04 (1.05)	0.95 (1.03)	1.17 (1.08)

our findings, we present the MSE reduction factors for the three parameters of Tree 3 and Tree 4 in Table 5, for $k = 4$, 6 and 8 proposals. They were obtained using 500 replications of 10 000 draws each. The acceptance rate was between 12% and 18%, depending on the number of proposals.

We also give the variance reduction factors (in parentheses) besides each MSE ratio. As we can see from these results, the QMC sampler never does much worse than the independent one, and in some cases reduces the MSE by factors of about 30%. The variance reduction factors are in general smaller than the MSE ones, mostly because in some cases the bias is quite large for both samplers. This suggests that the QMC sampler cannot solve completely the slow mixing of the original chain. The results obtained in Section 4.2 seem to suggest as well that the performance of the sampler with correlated proposals builds upon the performance of the one with independent proposals. The antithetic implementation does not produce additional savings with the exception brought by extreme situations in which the acceptance ratio for the original MTM is low (≈ 1%).

5 Conclusions

The MCTM algorithm requires small modifications once an MTM is designed. Provided the acceptance rates of the two are comparable, the MCTM is more efficient in either the antithetic or the stratified implementation, especially if the number of proposals is increased. Further research is necessary to understand possible relations between the acceptance rate of MTM and the increase in efficiency brought by MCTM.

As for which version of the MCTM is better, what we could say is that for users who want to work with a very small number of proposals, choosing the antithetic implementation is probably best, since the stratified version requires about 7 or 8 proposals to start improving upon the MTM. Beyond that point, the stratified implementation is probably a slightly better choice than the antithetic one.

While the discussion of the present paper has focused on Multiple-Try Metropolis, we believe that the idea of stratification and antithetic sampling can be further implemented in other local-search algorithms designed for Monte Carlo sampling.

Acknowledgment The authors would like to thank the Editor, the Associate Editor and one referee for a set of helpful suggestions. Financial support for both authors from the Natural Sciences and Engineering Council of Canada is gratefully acknowledged.

References

Atchade Y.F. and Perron F. 2005. Improving on the independent Metropolis-Hastings algorithm. Statistica Sinica 5: 3–18.

Atchade Y.F. and Rosenthal J. 2005. On adaptive Markov Chain Monte Carlo algorithms. Bernoulli 11: 815–828.

Bliss C.I. 1935. The calculation of the dosage-mortality curve. Ann. Appl. Biology 22: 134–167.

Caflisch R.E., Morokoff W., and Owen A.B. 1997. Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. The Journal of Computational Finance 1: 27–46.

Carlin B.P. and Louis T.A. 2000. Bayes and Emprirical Bayes Methods for Data Analysis, Chapman and Hall, London.

Chen M.-H. and Schmeiser B.W. 1993. Performances of the Gibbs, hit-and-run, and Metropolis samplers. J. Computat. Graph. Statist. 2: 251–272.

Craiu R.V. and Meng X.L. 2005. Multi-process parallel antithetic coupling for backward and forward MCMC. Ann. Statist. 33: 661–697.

Cranley R. and Patterson T. 1976. Randomization of number theoretic methods for multiple integration. SIAM J. Numer. Anal. 13: 904–914. .

Draper N.R. and Smith H. 1981. Applied Regression Analysis and Its Applications, Wiley, New York.

Frigessi A., Gåsemyr J., and Rue H. 2000. Antithetic coupling of two Gibbs sampler chains. Ann. Statist. 28: 1128–1149.

Gelman A. and Meng X.L. 1991. A note on bivariate distributions that are conditionally normal. Am. Statistician 125–126.

Gilks W.R., Roberts R.O., and George E.I. 1994. Adaptive direction sampling. Statistician 43: 179–189.

L'Ecuyer P. and Lemieux C. 2002. Recent advances in randomized quasi-Monte Carlo methods. In: Dror M., L'Ecuyer P., and F. Szidarovszki (Eds.), Modelling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications, Kluwer Academic Publishers, Boston, pp. 419–474.

Lemieux C. and Sidorsky P. 2006. Exact sampling with highly-uniform point sets. Mathematical and Computer Modelling 43: 339–349.

Lindstrom M.J. and Bates D.M. 1990. Nonlinear mixed effects models for repeated measures data. Biometrics 46: 673–687.

Liu J.S., Liang F., and Wong W.H. 2000. The use of Multiple-Try method and local optimization in Metropolis sampling. J. Amer. Statist. Assoc. 95: 121–134.

Loh W.L. 1996. On Latin hypercube sampling. Ann. Statist. 24: 2058–2080.

Marsaglia G. 1962. Random variables and computers. In: Koseznik J. (Ed.), Information Theory and Statistical Decision Functions Random Processes: Transactions of the Third Prague Conference, Czechoslovak Academy of Sciences, Prague, pp. 499–510.

McKay M.D., Beckman R.J., and Conover W.J. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21: 239– 245.

Niederreiter H. 1992. Random Number Generation and Quasi-Monte Carlo Methods. SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 63, SIAM, Philadelphia.

Owen A.B. 1992. A central limit theorem for Latin hypercube sampling. J. Roy. Statist. Soc. Ser. B 54: 541–551.

Owen A.B. 2003. Quasi Monte Carlo. Presented at the First Cape Cod Workshop on Monte Carlo Methods.

Owen A.B. and Tribble S.D. 2004. A quasi-Monte Carlo Metropolis algorithm. Technical Report 2004-35, Department of Statistics, Stanford University.

Paskov S. and Traub J. 1995. Faster valuation of financial derivatives. Journal of Portfolio Management 22: 113–120.

Ritter C. and Tanner M.A. 1992. Facilitating the Gibbs sampler: the Gibbs stopper and the griddy-Gibbs sampler. J. Amer. Statist. Assoc. 87: 861–868.

Stein M. 1987. Large sample properties of simulations using Latin hypercube sampling. Technometrics 29: 143–151, (Correction: 32, 367).

Sobol' I.M. 1967. The distribution of points in a cube and the approximate evaluation of integrals. U.S.S.R. Comput. Math. and Math. Phys. 7: 86–112.

Tan Z. 2006. Monte Carlo integration with acceptance-rejection. J. Comput. Graphical Statist., to appear.

van Dyk D. and Meng X.L. 2001. The art of data augmentation (with discussion). J. Comput. Graphical Statist. 10: 1–111.