Acceleration of the Multiple-Try Metropolis using Antithetic and Stratified Sampling

Radu V. Craiu[1] Christiane Lemieux[2]

October 4, 2005

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 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.

Key words and phrases: Antithetic variates, Markov chain Monte Carlo, Extreme antithesis, Korobov rule, Latin Hypercube sampling, Quasi-Monte Carlo, 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 (2003), Atchade and Rosenthal (2003), Liu, Liang and Wong (2000),

Draper and Liu (2003), Gilks, Roberts and George (1994), Chen and Schmeiser (1993). In this paper we discuss possible accelerations of the Multiple-Try Metropolis (MTM) of Liu, Liang and Wong (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, Gåsemyr and Rue, 2000; Craiu and Meng, 2005), and quasi-Monte Carlo (QMC) methods (Owen and Tribble, 2004; Lemieux and Sidorsky, 2004), which can be thought of as a way of producing stratified samples based on a highly-uniform point set. 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, Morokoff and Owen, 1997).

The 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 which the updating function is monotone in all arguments. However, one of the most used MCMC algorithms in practice, the Metropolis-Hastings, 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 discussions in Section 5.

2 Multiple correlated-try Metropolis

Consider $T(x; y)$ a Metropolis proposal transition rule so that one can define

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

where $\lambda(x, y)$ is a nonnegative symmetric function that can be chosen by the user. The original MTM proposed by LLW has the following steps.

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

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

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 $\tilde{T}(x; \cdot)$ and are exchangeable. To simplify the arguments, unless otherwise stated we take $\lambda(x, y) = 1$ throughout this paper. 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.

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

1. Draw k trial proposals $y_1, \ldots, y_k$ from $\tilde{T}(x; \cdot)$. Compute $w(y_j, x) = \pi(y_j)T(y_j; x)$, for each j;

2. Select $Y = y$ among the k trials with probability proportional to $w(y_j, x)$.

3. Draw $(x_1^*, \ldots, x_{k-1}^*)$ variates from the conditional transition kernel $\tilde{T}(y; \cdot | x_k = x)$ and let $x_k^* = x$;

4. Accept y with probability

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

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

$$\begin{aligned}
\pi(x)A(x,y) &= \pi(x)P[\cup_{j=1}^k \{Y_j = y\} \cap \{I = j\}|x] = k\pi(x)P[\{Y_1 = y\} \cap \{I = 1\}|x] \\
&= k\pi(x) \int \tilde{T}(x; y, y_2, \ldots, y_k) \frac{w(y,x)}{w(y,x) + \sum_{i=2}^k w(y_i, x)} \times \\
&\quad \min \left\{ 1, \frac{w(y,x) + \sum_{i=2}^k w(y_i, x)}{w(x,y) + \sum_{i=2}^k w(x_i^*, y)} \right\} \tilde{T}(y; x_2^*, \ldots, x_k^* | x) dy_2 \ldots dy_k dx_2^* \ldots dx_k^* \\
&= k \frac{w(y,x)w(x,y)}{\lambda(y,x)} \int \min \left\{ \frac{1}{w(y,x) + \sum_{i=2}^k w(y_i, x)}, \frac{1}{w(x,y) + \sum_{i=2}^k w(x_i^*, y)} \right\} \times \\
&\quad \tilde{T}(y; x_2^*, \ldots, x_k^* | x) \tilde{T}(x; y_2, \ldots, y_k | y) dy_2 \ldots dy_k dx_2^* \ldots dx_k^* = \pi(y)A(y,x).
\end{aligned}$$

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

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 further 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 MTM. 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 $N_2\left((x, y)^T, \Sigma\right)$, 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 $x_i' \perp y_i'$. 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

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

is maximized when ρ_1 is equal to its smallest possible value. In general it is more efficient to take $\rho_2 = 0$. 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, Beckman and Conover (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 $v_1, \dots v_k \sim \text{Uniform}(0, 1)$.

Step II Select a random permutation τ of $\{0, \dots, k-1\}$

Step III Construct $u_i = (v_i + \tau(i))/k$, for all $1 \leq i \leq k$.

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 Figure 1.

1. Draw k proposals using the uniform deviates $u_1 \dots, u_k$ 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 $0 \leq j \leq k-1$, $j \neq j_0$ construct $u_j^* = (\tau(j) + w_j)/k$ where the $w_j \sim \text{Uniform}(0, 1)$ are independent.

3. For each $j \neq j_0$, $x_j^* = F_y^{-1}(u_j^*)$ and $x_{j_0}^* = x$.

Figure 1: MCTM with LHS proposals

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 $G : [0,1)^r \to \mathbb{R}^r$ 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.

The sample $y_1, \ldots, y_k$ thus obtained can then be used to estimate quantities of the form $\mu = \mathrm{E}(f(Y))$, where f is some real-valued function, by the unbiased estimator

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

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 3.2 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 Figure 3,

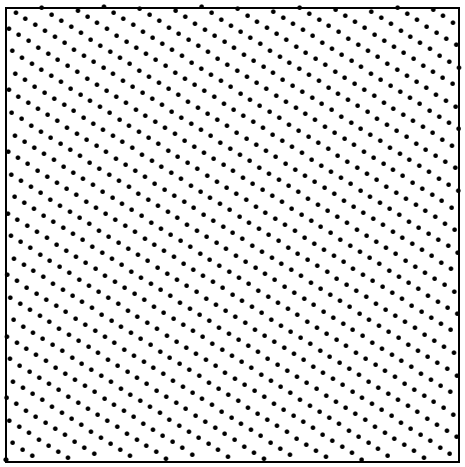

Figure 2: *Two-dimensional Korobov rule with* $k = 1024$ *and* $a = 139$.

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 $T(y; x)$.

1. Draw k trial proposals $y_1, \ldots, y_k$ using a randomly-shifted point set $\tilde{P}_k$. More precisely, let $y_i = G_x(\tilde{\mathbf{u}}_i)$, for $i = 1, \ldots, k$. Compute $w(y_j, x) = \pi(y_j)T(y_j; x)$.

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 $G_y(\mathbf{w}) = x$.

4. Set $x_j^* = G_y((\mathbf{u}_j + \mathbf{w}) \bmod 1)$ for $j > 1$.

Figure 3: MCTM with QMC proposals

In other words, once we have our set of trial proposals $y_1, \ldots, y_k$ and one of them, y_j, 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.

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

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

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 unrandomized 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, which essentially amounts to perform importance sampling. 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 $g : [0,1] \to [0,1]$ 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 Figure 4.

1. Draw k trial proposals $y_1, \ldots, y_k$ using the randomized (and transformed) point set Q_k. More precisely, let $y_i = G_x(g(\tilde{\mathbf{u}}_i))$, for $i = 1, \ldots, k$, where $\tilde{P}_k = \{\tilde{\mathbf{u}}_i, i = 1, \ldots, k\}$ 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 $P_k = \{\mathbf{u}_i, i = 1, \ldots, k\}$ is the unrandomized QMC point set.

Figure 4: MCTM with QMC proposals to which a transformation has been applied

In the above 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 $\lambda(x, y) = 1$.

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 Figure 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 $u \in (0, 1)$.

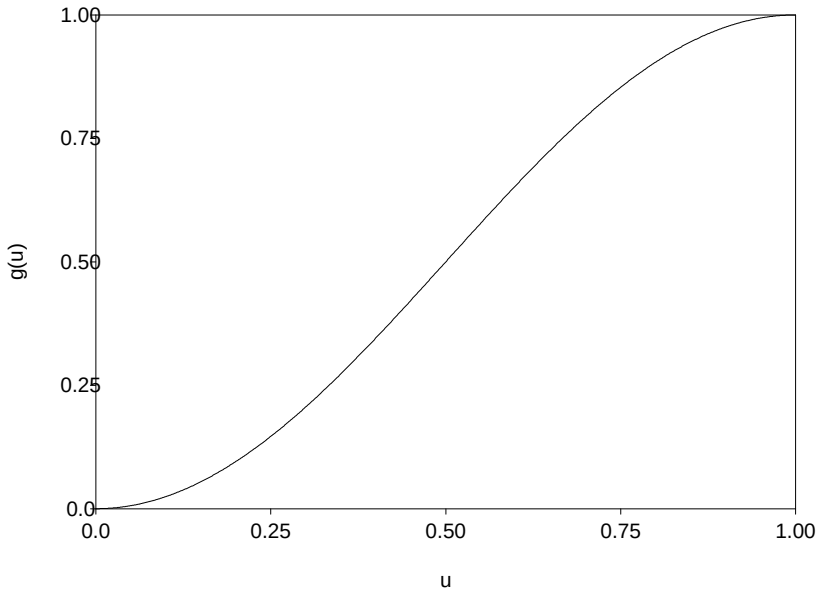

Figure 5: *Transformation $g(u)$ used in Section 4.2.*

These properties imply that, if $U \sim \text{Uniform}(0,1)$ 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 $Z_j = \Phi^{-1}(g(U_j))$, $j = 1, \ldots, r$, where the U_j's are i.i.d. Uniform$(0,1)$, then $Z_1, \ldots, Z_r$ has joint density

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

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

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

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

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 $U \sim \text{Uniform}(0,1)$, then

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

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 Figure 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 $\alpha = 1$.

Note that transformations $g(\cdot)$ that are not necessarily bijections but that preserve uniformity – that is, such that if $\mathbf{u}$ is uniformly distributed over $[0,1)^r$, then $g(\mathbf{u})$ is also uniform – can also be applied. When doing so, only the last step of the MCTM must be modified, in a similar way to what is described in Figure 4. More precisely, when the shift $\mathbf{w}$ is found

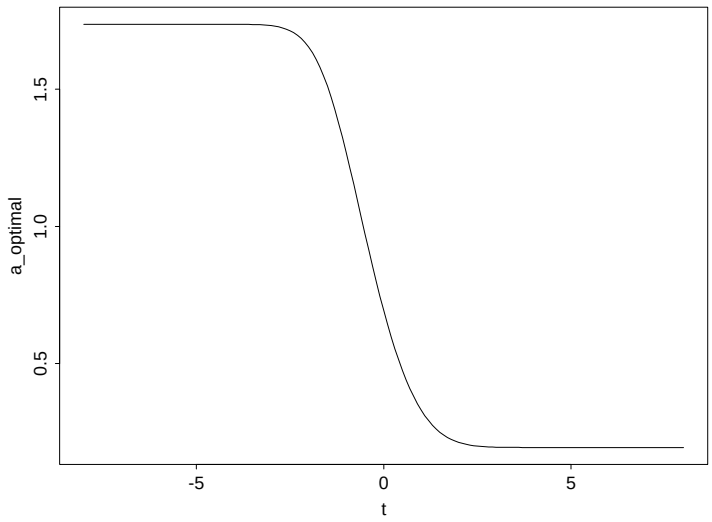

Figure 6: *Plot of* $\alpha_{opt}(t)$.

in step 3, then in step 4 we must replace $\mathbf{w}$ by another shift $\hat{\mathbf{w}}$ such that $G_y(g(\mathbf{w})) = x$. For instance, if we use the *baker transformation* (Hickernell, 2002), which is done by applying

$$g(u) = \begin{cases} 2u & \text{if } u < 0.5 \\ 2(1-u) & \text{otherwise,} \end{cases}$$

to each coordinate, then we can define $\hat{\mathbf{w}}$ as follows: for each $j = 1, \ldots, r$, choose one of the two subintervals $V = [0, 1/2)$ and $W = [1/2, 1)$ with probability $1/2$; if V is chosen, then let $\hat{w}_j = w_j/2$, otherwise let $\hat{w}_j = 1 - w_j/2$.

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.

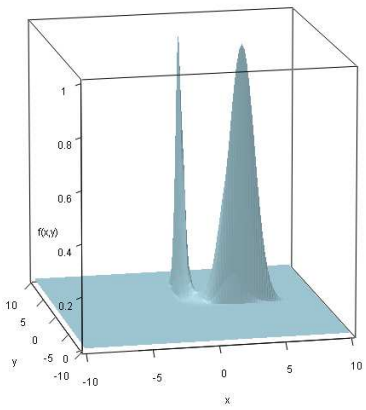

Figure 7: *Random-ray Monte Carlo. The bivariate density* $f(x_1, x_2)$.

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, Roberts and George, 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

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

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 Figure 7.

The construction of the random-ray Monte Carlo via MTM has been detailed by LLW and is followed here. More precisely, 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 $T_{\mathbf{e}}(x, \cdot)$ 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 look at 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 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.

Table 1: *Values of the MSE reduction factor* $R = \dfrac{MSE_{anti}}{MSE_{ind}}$.

$\sigma \backslash k$	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 the data are taken from van Dyk and Meng (2001) and consist 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 (2003) and consider that the prior for $\beta = (\beta_0, \beta_1, \beta_2)^T$ is trivariate normal with zero mean and variance $\text{diag}(100^2, 100^2, 100^2)$. The posterior density is then proportional to

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

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 j^{th} sample point drawn in the i^{th} 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 $i = 1, \ldots, M$ the MSE is defined as

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

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	-

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, 2004).

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, 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.

4.3 Multivariate t-distribution

In the last example we investigate the benefits of implementing MCTM in situations in which the distribution of interest has a larger number of dimensions. More specifically, suppose that we are interested in simulating from a r-variate non-central t distribution with density

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

given by

$$t(x|\mu,\sigma^2\Sigma,\nu) = \frac{\Gamma[(\nu+r)/2]}{(\pi\nu)^{r/2}|\sigma^2\Sigma|^{1/2}\Gamma(\nu/2)}\left(1+\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{\sigma^2\nu}\right)^{-(\nu+r)/2}, \qquad (4.2)$$

where $\Gamma(y)$ is the single parameter Gamma function. It is known that the mean of the distribution (4.2) is μ and the variance-covariance matrix is $\sigma^2\Sigma\nu/(\nu-2)$.

Multivariate t distributions are used, among others, in robust linear regression models (Pinheiro et al., 2001) and mixture modelling (Peel and McLachlan, 2000). Efficient simulation methods that perform well under any variance-covariance matrix Σ are still under investigation (Bretz and Genz, 2002).

In this example we assume that the number of dimensions, $r = 25$, and that the matrix Σ is block diagonal with each block of size $r_1 = 5$.

We try to consider the least favourable situation in which the user does not have information about the shape of the distribution so we use a random walk Metropolis with proposals following a multivariate ($r = 25$) Gaussian distribution with diagonal variance-covariance matrix. The standard deviation of the proposals is chosen to be either $\tau = 1.25$ or $\tau = 1.5$. In our example $\sigma = 2$ and $\nu = 3$. The values of Σ and μ were chosen to be different from zero and are available from the authors. The QMC proposals were implemented with the baker transformation mentioned in Section 3.2.1.

As in previous examples in Table 5 we report the ratio R of mean squared errors for four different marginal components of $x = (x_1, \ldots, x_{25})$. One can notice that the use of MCTM(antithetic) starts to be beneficial once the number of proposals is larger, with higher reduction factors obtained for $k = 16$. Unlike previous examples the use of MCTM is slowing down the performance for smaller values of $k = 3, 6$, and in one case for $k = 16$ with QMC proposals.

Table 5: *Reduction factor R for the multivariate t distribution: antithetic.*

	Antithetic				QMC	
	$\tau = 1.25$					
$X_i \setminus k$	3	6	8	16	8	16
X_1	> 2	$\in (1, 2)$	0.52	0.40	0.65	1.30
X_5	> 2	$\in (1, 2)$	0.55	0.34	0.55	0.90
X_{13}	> 2	$\in (1, 2)$	0.63	0.37	0.49	0.78
X_{23}	> 2	$\in (1, 2)$	0.76	0.43	0.72	0.82
	$\tau = 1.5$					
X_1	> 2	$\in (1, 2)$	0.76	0.18	0.64	0.15
X_5	> 2	$\in (1, 2)$	0.66	0.13	0.70	0.07
X_{13}	> 2	$\in (1, 2)$	0.79	0.14	0.85	0.04
X_{23}	> 2	$\in (1, 2)$	0.73	0.12	0.81	0.03

5 Conclusions

The MCTM algorithm requires small modifications once a 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.

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.

References

Atchade, Y. F. and Perron, F. (2002). Improving on the independent Metropolis-Hastings algorithm. *Technical Report*, Department of Mathematics and Statistics, University of Montreal.

Atchade, Y. F. and Rosenthal, J. (2003). On adaptive Markov Chain Monte Carlo algorithms. *Technical Report*, Department of Statistics, University of Toronto.

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, P. C. 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, D. and Liu, S. (2003). Strategies for MCMC acceleration. *Technical Report*, Department of Applied Mathematics and Statistics, University of California, Santa Cruz.

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*, 125–6.

Genz A. and Bretz F. (2002) Comparison of methods for the computation of multivariate t probabilities. *J. Computat. Graph. Statist.*, **11**, 950–971.

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

Hammersley, D. C. and Morton, K. V. (1956). A new Monte Carlo technique: Antithetic Variates. *Math. Proc. Cambridge Philos. Soc.*, **52**, 449–475.

Hickernell, F. J. (2002). Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules, Monte Carlo and Quasi-Monte Carlo Methods 2000, K.-T. Fang, F.J. Hickernell, and H. Niederreiter, eds., Springer-Verlag, Berlin, pp. 274–289.

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

Lemieux, C. and Sidorsky, P. (2004) Exact sampling with highly-uniform point sets. Technical Report 2004-762-27, Department of Computer Science, University of Calgary.

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. *Information Theory and Statistical Decision Functions Random Processes: Transactions of the Third Prague Conference*, J. Koseznik Ed., Czechoslovak Academy of Sciences, Prague, pages 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, volume 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. Manuscript.

Paskov, S. and Traub, J. (1995). Faster valuation of financial derivatives. *Journal of Portfolio Management*, **22**, 113–120.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, **10**, 339 – 348.

Pinheiro J. C., Liu C. and Wu Y.N. Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution. *J. Computat. Graph. Statist.*, **10**, 249–276.

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.

Roberts, G. O., Gelman, A. and Gilks, W. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithm. *Ann. Appl. Probab.*, **7**, 110–120.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling of various Metropolis-Hastings algorithms. *Statist. Sci.*, **16**, 351–367.

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

Tan, Z. (2003). Monte Carlo integration with acceptance-rejection. *Dept. of Biostatistics Working Papers*, Johns Hopkins University,

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

Affiliation of authors

1: Department of Statistics, University of Toronto, 100 St. George Street, Toronto, ON, M5S 3G3, Canada, craiu@utstat.toronto.edu

2: Department of Mathematics and Statistics, University of Calgary, 2500 University Drive N.W., Calgary, AB, T2N 1N4, Canada, lemieux@math.ucalgary.ca

List of Figures