Antithetic Coupling for Perfect Sampling

RADU V. CRAIU and XIAO-LI MENG[1]
THE UNIVERSITY OF CHICAGO, USA

SUMMARY

This paper reports some initial investigations of the use of antithetic variates in perfect sampling. A simple random walk example is presented to illustrate the key ingredients of antithetic coupling for perfect sampling as well as its potential benefit. A key step in implementing antithetic coupling is to generate $k \geq 2$ random variates that are *negatively associated*, a stronger condition than negative correlation as it requires that the variates remain non-positively correlated after any (component-wise) monotone transformations have been applied. For $k = 2$, this step is typically trivial (e.g., by taking U and $1 - U$, where $U \sim U(0, 1)$) and it constitutes much of the common use of antithetic variates in Monte Carlo simulation. Our emphasis is on $k > 2$ because we have observed some general gains in going beyond the commonly used pair of antithetic variates. We discuss several ways of generating negatively associated random variates for arbitrary k, and our comparison generally favors *Iterative Latin Hypercube Sampling*.

Keywords: ANTITHETIC VARIABLES, COUPLING FROM THE PAST, EXACT SAMPLING, ITERATIVE LATIN HYPERCUBE SAMPLING, NEGATIVE ASSOCIATION, RANDOM WALK.

1. PERFECT SAMPLING

Exploring a probability distribution π using MCMC methods is now a routine practice in Bayesian statistics. The main idea is to run a Markov chain whose stationary distribution is π. After an initial "burn-in" period, the frequency with which the chain moves within the state space can be used to approximate the target distribution. An important practical issue is to determine how long we need to run the chain in order to achieve acceptable accuracy in this approximation. As discussed in Wilson (2000), among all methods that are currently available, the best one, not surprisingly, is to avoid the problem in the first place. The so-called *perfect sampling* or *exact sampling*, introduced by Propp and Wilson (1996), has gained great and rapid attention because it achieves exactly that. This class of methods, by cleverly tracing coupled Markov chains from the past, obtains samples which are both independent and distributed exactly according to the limiting distribution π, hence eliminating the worry of the chain not being run long enough.

Currently, the perfect sampling algorithms rely mostly on the *coupling from the past* (CFTP) technique (Propp and Wilson, 1996). In a nutshell, CFTP makes it possible to trace the chain's sample paths from all – possibly infinitely many – states simultaneously

[1]We thank Duncan Murdoch for comments and NSF and NSA for partial research support.

from the indefinite past, i.e., $t = -\infty$. Because these paths are coupled via the use of common random variates $\{U_t, t \leq 0\}$ in the Markov chain updating, $X_{t+1} = \psi(X_t, U_{t+1})$, they will eventually coalesce and the common value at $t = 0$ will be a draw from the limiting distribution π. Currently, the actual construction of an effective coupler for a particular problem at hand can be quite challenging or even infeasible, especially for many routine Bayesian problems with unbounded state spaces and complex distribution forms. The use of perfect sampling for Bayesian computation is therefore a developing area, and some initial findings and successes are reported in Murdoch and Green (1998), Møller and Nicholls (1999), Holmes and Mallick (1999), Murdoch (2000), and Wilson (2000). An informative example, both in terms of the potential and difficulties of using perfect sampling for routine Bayesian analysis, can be found in Murdoch and Meng (2000) in this volume.

The construction is most straightforward for the class of Markov chains with a discrete state space which has a maximum state and a minimum state and where $\psi(X, U)$ is monotone (or anti-monotone) in X with respect to a partial ordering. With such a chain, we only need to trace the *top chain* starting from the maximum state and the *bottom chain* starting from the minimum state. If these two chains have coalesced by time $t = 0$, then it is clear that a chain starting from any other state has to coalesce to the same value of X_0 because it is "sandwiched" by the top and bottom chains. There are many applications where such a construction is possible, including the original Ising model application of Propp and Wilson (1996). Although the CFTP technique now covers a much wider range of applications than such discrete bounded monotone chains (see Wilson's annotated bibliography at `http://dimacs.rutgers.edu/~dbwilson/exact`), and although our antithetic coupling is not restricted to such chains either, in this paper we will focus on such cases because they are more easily dealt with in both theoretical analysis and implementation.

2. K-PROCESS ANTITHETIC COUPLING

A primary use of draws $\{X_1, \ldots, X_n\}$ from π is to estimate the expectation of $f(X)$, $E_\pi f$, for various choices of f. By far, the most frequently used estimator of $E_\pi f$ is the sample average, $\bar{f}_n = \sum_i f(X_i)/n$. In cases where $\{X_1, \ldots, X_n\}$ are pair-wise uncorrelated, it is well known that the variance $V(\bar{f}_n) = \sigma_f^2/n$, where $\sigma_f^2 = V_\pi(f(X))$. The antithetic principle (Hammersley and Morton 1956) aims to reduce this variance by inducing negative correlations among $\{f_i \equiv f(X_i), i = 1, \ldots, n\}$. Specifically, suppose the draws $\{X_1, \ldots, X_n\}$ were obtained from $k \geq 2$ parallel implementations of the same simulation process (e.g., a CFTP algorithm), each of which contributed $m = n/k$ independent and identically distributed (i.i.d.) draws. In other words, we can relabel the n draws as $\{X_{ij}, i = 1, \ldots, m; j = 1, \ldots, k\}$, and accordingly $\{f_i, i = 1, \ldots, n\} = \{f_{ij}, i = 1, \ldots, m; j = 1, \ldots, k\}$. It is then easy to check that

$$V(\bar{f}_n) = \frac{\sigma_f^2}{n}\left[1 + (k-1)\rho_k^{(f)}\right] \equiv \frac{\sigma_f^2}{n} R_k^{(f)}, \tag{1}$$

where $\rho_k^{(f)} = corr(f_{ij_1}, f_{ij_2})$, $j_1 \neq j_2$. Consequently, if we can implement in parallel the k simulation processes such that $\rho_k^{(f)} < 0$, then we can reduce the Monte Carlo variance by the factor $R_k^{(f)}$ relatively to the variance we would obtain from independent implementation

($\rho_k^{(f)} = 0$). We emphasize here that $\rho_k^{(f)}$, and thus $R_k^{(f)}$, depends on both f and k. For example, $\rho_k^{(f)} \geq -(k-1)^{-1}$, since $R_k^{(f)} \geq 0$. We also emphasize that the fact that $\rho_k^{(f)}$ becomes less and less negative with the increase of k is not a reason to prefer small k (e.g., $k = 2$), because the relevant reduction is not measured by $\rho_k^{(f)}$, but rather by $R_k^{(f)}$, which magnifies the contribution of $\rho_k^{(f)}$ by the compensating factor $(k-1)$. We have good empirical evidence that it is beneficial in general to go beyond $k = 2$, which is undoubtedly the most common choice in the literature, including the recent use of antithetic coupling in forward MCMC algorithms (e.g., Frigessi, Gåsemyr, and Rue, 1999).

Figure 1 sketches how we will proceed to produce k antithetically coupled CFTP processes. The jth process, $j = 1, \ldots, k$, is represented by a pair of top-bottom chains, and they are positively coupled because they share the same sequence of (not necessarily uniform) random variates $\{U_t^{(j)}, t \leq 0\}$. Whereas for given j, $\{U_t^{(j)}, t \leq 0\}$ need to be mutually independent in order to have a legitimate CFTP algorithm, there is no requirement for the joint distribution $\mathcal{U}_t^k \equiv \{U_t^{(1)}, \ldots, U_t^{(k)}\}$ other than that each margin has to have the same distribution specified by the underlying CFTP algorithm. It is thus legitimate, for example, to use a joint distribution such that the common pair-wise correlation of the components of $\mathcal{U}_t^k$ is negative. However, this is not enough to guarantee the output of the k processes, $\{X_0^{(j)}, j = 1, \ldots, k\}$, to be negatively correlated because the sign of a correlation is generally not preserved after a transformation has been applied to each variable, even if such transformation is monotone (see Section 3.1 for an example). For the jth CFTP process, the output $X_0^{(j)}$ is a function of $\{U_t^{(j)}, t \leq 0\}$, and thus we need a stronger notion than negative correlation for $\{U_t^{(1)}, \ldots, U_t^{(k)}\}$ in order to guarantee a negative correlation between $X_0^{(j)}$ and $X_0^{(j')}$, $j \neq j'$, as well as a negative correlation between $f(X_0^{(j)})$ and $f(X_0^{(j')})$ for any f that is monotone in each component of X (clearly this cannot be guaranteed for a non-monotone f).

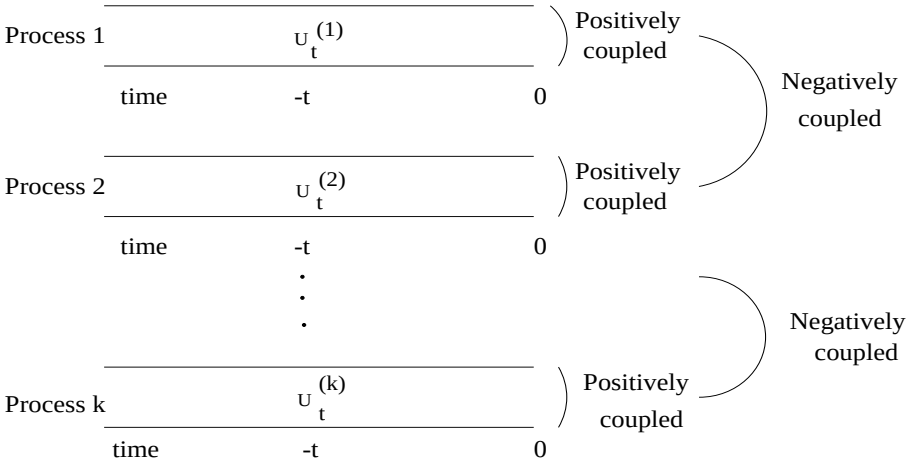

Figure 1. *k-process Antithetic Coupling for CFTP*

The stronger notion we adopt here is the *negative association* defined by Joag-Dev and Proschan (1983). A collection of variables $\{X_1, \ldots, X_n\}$ is called *negatively associated*

(NA) if for every pair of disjoint subsets A_1 and A_2 of $\{1, \ldots, n\}$,

$$corr\ (f_1(X_i, i \in A_1), f_2(X_j, j \in A_2)) \leq 0$$

whenever both f_1 and f_2 are nondecreasing (or nonincreasing) in each of their arguments. For our application, we need to extend this notion to an infinite sequence of random variables, which can be done easily, as in Craiu and Meng (2000). A very nice property of NA is that it is closed under independent unions (Joag-Dev and Proschan, 1983). Therefore, if we can make $\mathcal{U}_t^k = \{U_t^{(1)}, \ldots, U_t^{(k)}\}$ negatively associated for each t, then so will be the whole collection $\{U_t^{(1)}, \ldots, U_t^{(k)}, t \leq 0\}$ because the vectors $\{\mathcal{U}_t^k, t \leq 0\}$ are mutually independent. Consequently,

$$corr\left(\Phi(U_t^{(j)}, t \leq 0), \Phi(U_t^{(j')}, t \leq 0)\right) \leq 0, \quad j \neq j'$$

for any Φ that is increasing/decreasing in each of its arguments. Taking Φ to be the infinite composition of the chain mapping $X_{t+1} = \phi(X_t, U_{t+1})$ yields the desired result when $\phi(X, U)$ is monotonically increasing or decreasing in both X and U, as in our illustrative example of Section 4. The monotonicity with respect to X is automatic under our monotone chain assumption, and the corresponding monotonicity with respect to U often can be made true since we have some freedom in choosing the form of U. For example, suppose $\phi(X, U)$ is monotone increasing in X but monotone decreasing in U, where $U \sim U(0, 1)$. We can then use $\phi(X, 1 - U)$, which will then satisfy the requirement.

The NA concept refers to the qualitative aspect of negative dependence. For a quantitative measure of the latter, that is, for a measure of how antithetic the variables are, we need the notion of *extreme antithesis* (EA). We say that $\{X_1, X_2, ..., X_k\}$ achieve EA with respect to a distribution F if they are exchangeable and

$$corr(X_i, X_j) = min\left\{corr(Y_i, Y_j) : Y_1, ..., Y_k \text{ exchangeable } \& \ \forall i, Y_i \sim F\right\}.$$

For $k = 2$, it is trivial to achieve both NA and EA by letting $X_1 = F^{-1}(r)$ and $X_2 = F^{-1}(1 - r)$, where $r \sim U(0, 1)$. In fact this construction achieves EA for any monotone function g in the sense that it also minimizes $corr(g(X_1), g(X_2))$ with $X_1, X_2 \sim F$ (see, for example, Craiu and Meng, 2000, for discussion and references). For $k > 2$, however, the matter is much more complicated, and there is no universal way of accomplishing both NA and EA for arbitrary F. In Section 3 we discuss three methods, which will then be applied and compared in Section 4 in our illustrative example.

3. GENERATING K ANTITHETIC VARIATES

Section 2 indicates that a main task in implementing k-process antithetic coupling is to generate $\{U^{(1)}, \ldots, U^{(k)}\}$ that are NA and with the desired margin, often $U(0, 1)$. To achieve as much a variance reduction as possible, we also aim at EA. To fix our target, suppose our goal is to generate k exchangeable random variables $\{u_1, ..., u_k\}$ such that:

(I) Marginally, each component u_i is $U(0, 1)$ distributed;

(II) $\{u_1, ..., u_k\}$ are NA;

(III) $\{u_1, ..., u_k\}$ achieve EA, i.e., $corr(u_j, u_{j'}) = -(k-1)^{-1}$, for any $j \neq j'$.

Before we proceed to discuss our methods, we mention in passing that the generation of $\{u_1, ..., u_k\}$ that satisfy criteria (I) and (III) was often treated as a "fun-math" problem in the literature, and there are many ways of doing that (e.g., see Bondesson (1983) and the subsequent solutions). However, the criterion (II), which is at the core of our applications, has received much less attention and it is more difficult to satisfy or to prove it is satisfied. Indeed, for the method given in Arvidsen and Johnsson (1982), which is a key building block of our first method, the resulting k-tuple is not NA for any $k \geq 3$. (We give assertions without proof in this paper. A full development is given in Craiu and Meng (2000).)

3.1. *Permuted Displacement Method*

The first method we have is a permuted version of the one given in Arvidsen and Johnsson (1982) and has two steps:

Step 1 Let $G(x) = x - [x]$ (fractional part of x) and $r_1 \sim U(0,1)$. Compute

$$r_i = G(2^{i-2} r_1 + \frac{1}{2}), \quad i = 2, ..., k-1, \quad \text{and} \quad r_k = 1 - G(2^{k-2} r_1).$$

Step 2 Let S_k be the set of all permutations of $\{1, \ldots, k\}$. Pick at random a $\sigma = (\sigma(1), \ldots, \sigma(k)) \in S_k$, and set $u_i = r_{\sigma(i)}$.

Step 1 is the original method given in Arvidsen and Johnsson (1982), and we call it the *displacement* method because it corresponds to displacing the digits in the binary expansion of r_1 such that $\sum_{i=1}^{k} r_i = k/2$; see Craiu and Meng (2000). It follows immediately that the permuted version $\{u_1, \ldots, u_k\}$ has the same property (i.e. $\sum_{i=1}^{k} u_i = k/2$) and thus it achieves EA because $\{u_1, \ldots, u_k\}$ are exchangeable. Note that the permutation not only renders the exchangeability, but more importantly makes it possible to achieve NA. (However, currently we do not have a proof that $\{u_1, \ldots, u_k\}$ are NA, though we conjecture this is true.) The $\{r_1, \ldots, r_k\}$ from Step 1 are not NA, as the following example illustrates. Let $k = 5$ and $f(x) = -1$ if $0 \leq x \leq 1/5$ and $f(x) = 0$ if $x > 1/5$. Then while $corr(r_2, r_4) = -\frac{1}{8} < 0$, $corr(f(r_2), f(r_4)) = \frac{1}{20} - \frac{1}{25} > 0$.

3.2. *Multivariate Normal Method*

The second method we present uses the fact that negatively correlated multivariate normal variables are NA (Joag-Dev and Proschan, 1983). The method again has two steps:

Step 1 Generate

$$\begin{bmatrix} Z_1 \\ \cdot \\ \cdot \\ Z_{k-1} \end{bmatrix} \sim N_k \left(\begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -\frac{1}{k-1} & ... & -\frac{1}{k-1} \\ -\frac{1}{k-1} & 1 & ... & -\frac{1}{k-1} \\ ... & ... & ... & ... \\ -\frac{1}{k-1} & -\frac{1}{k-1} & ... & 1 \end{bmatrix} \right),$$

and let $Z_k = -(Z_1 + Z_2 + ... + Z_{k-1})$.

Step 2 Compute $u_i = \Phi(Z_i), i = 1, \ldots, k$, where $\Phi(z)$ is the CDF of $N(0,1)$.

Since $\Phi(z)$ is monotone, $\{u_1, \ldots, u_k\}$ are exchangeable and NA because $\{Z_1, \ldots, Z_k\}$ are. However, $\{u_1, \ldots, u_k\}$ do not achieve EA when $k \geq 3$ because $corr(u_1, u_2) > -(k-1)^{-1}$ when $k \geq 3$. Another disadvantage of this method is that the cumulative cost of evaluating $\Phi(z)$ repeatedly can be nontrivial when the number of repetitions is high.

3.3 *Iterative Latin Hypercube Sampling*

Our last method is the most satisfactory as it achieves all three criteria and it is easy to implement. It is an iterative version of a scheme for stratified sampling devised by McKay, Beckman and Conover (1979). Their *Latin hypercube sampling* makes it possible to stratify on all the input dimensions simultaneously. For properties of Latin hypercube sampling we refer the reader to some previous articles (e.g., Stein, 1987; Owen, 1992; Loh, 1996).

Our iterative procedure can be described by the following steps:

Step 1 Generate $u^{(0)} = (u_1^{(0)}, ..., u_k^{(0)}) \overset{\text{i.i.d.}}{\sim} U(0,1)$.

Step 2 For $l = 1, 2, \ldots$, iteratively compute

$$u^{(l+1)} = \frac{1}{k}(K^{(l)} + u^{(l)}), \tag{2}$$

where the vectors $\{K^{(l)}, l = 1, 2, \ldots\}$ are i.i.d. permutations of $\{0, 1, \ldots, k-1\}$.

It is not hard to show that for any $l > 0$, $u^{(l)} = (u_1^{(l)}, ..., u_k^{(l)})$ satisfies both criteria (I) and (II); the latter follows from the fact that a permutation distribution is NA (Joag-Dev and Proschan, 1983). It is also easy to verify that for any $l > 0$,

$$corr(u_1^{(l)}, u_2^{(l)}) = -\frac{1}{k-1}\left(1 - \frac{1}{k^{2l}}\right) \overset{l\to\infty}{\longrightarrow} -\frac{1}{k-1}, \tag{3}$$

and thus $u^{(l)}$ achieves EA very rapidly as $l \to \infty$, especially for large k. Thus, we typically need to iterate only a few times in order to practically achieve EA.

There is an alternative way of implementing this method. Let $\{K^{(l)}, l = 1, 2, \ldots\}$ be the same i.i.d. permutations as above, and write $K^{(l)} = (a_1^{(l)}, \ldots, a_k^{(l)})$, $l = 1, 2, \ldots$. For any $L > 0$, we compute

$$u_j^{(L)} = \sum_{l=1}^{L} \frac{a_j^{(l)}}{k^l}, \qquad j = 1, \ldots, k.$$

As $L \to \infty$, given the same $K^{(l)}$'s, $u^{(L)} = (u_1^{(L)}, \ldots, u_k^{(L)})$ will have the same limit as $u^{(l)}$ of (**??**) as $l \to \infty$. For any finite L, $\{u_1^{(L)}, \ldots, u_k^{(L)}\}$ are NA because a permutation distribution is NA and NA is preserved under independent unions. Furthermore, one can easily check that for any $\epsilon > 0$, as long as $L \geq -\log \epsilon / \log k$, $\max_j |u_j^{(L)} - u_j^{(\infty)}| \leq \epsilon$. Thus, it is easy to control the precision of $u^{(L)}$ as approximation to $u^{(\infty)}$. It is also easy to see that for any $L > 0$,

$$\sum_{j=1}^{k} u_j^{(L)} = \frac{k}{2}\left(1 - \frac{1}{k^L}\right), \tag{4}$$

and thus $\{u_1^{(L)}, \ldots, u_k^{(L)}\}$ achieves EA for any L. Comparing (**??**) with (**??**), we see that a key difference between the first and the second method is that when we stop at a finite iteration l (for the first method) or use a finite L (for the second method), as we have to in the actual implementation, the first method gives up on achieving the exact EA, while the second method gives up on the exact marginal uniformity. Although both losses can be easily controlled to be as small as we wish, we prefer the first method not only because it is easier to implement but also because meeting criterion (I) exactly is more important than meeting criterion (III) exactly (especially because even when we meet (III) exactly, it does not imply that $\{X_0^{(1)}, \ldots, X_0^{(k)}\}$ achieves EA; we just hope that by getting closer to (III), we can obtain more variance reduction).

4. AN ILLUSTRATION – RANDOM WALK ON THE LINE

We consider a random walk on $S = \{1, 2, ..., N\}$ with semi-absorbent barriers, which has the updating function

$$X_{t+1} = \psi(X_t, U_{t+1}) = \begin{cases} min\{X_t + 1, N\}, & \text{if } U_{t+1} < p \\ max\{X_t - 1, 1\}, & \text{if } U_{t+1} > p \end{cases}, \text{ where } U_{t+1} \sim U(0,1).$$

Its stationary distribution is given by $\pi(x) \propto \left(\frac{p}{1-p}\right)^x$, $x \in S$, against which we validated (not shown here) the antithetically coupled CFTP algorithm described below. Note here that $\phi(X, U)$ is monotone increasing in X and monotone decreasing in U, but this does not cause any problem for applying our NA theory because, as we discussed in Section 2, we can always replace U by $1 - U$ (in theory, not in implementation).

Using the "binary back off" strategy of Propp and Wilson (1996), we implemented the k-process antithetically coupled CFTP algorithm as follows. Given a method discussed in Section 3 and a $T_0 > 0$, we carried out the following steps:

Step 0 Set $C_j = 0, j = 1, \ldots, k$; also $T_{old} = 0, T_{new} = T_0$.

Step 1 Using the chosen method to generate i.i.d. $\mathcal{U}_t^k$ for $-T_{new} < t \leq -T_{old}$, where $\mathcal{U}_t^k = \{U_t^{(1)}, \ldots, U_t^{(k)}\}$.

Step 2 For $j = 1, \ldots, k$, if $C_j = 1$, go to next j or **Step 3** if $j = k$. If $C_j = 0$, then run the top chain starting from $X_{-T_{new}} = N$ and the bottom chain starting from $X_{-T_{new}} = 1$ and couple them using the common $\{U_t^{(j)}, -T_{new} < t \leq 0\}$. If the two chains coalesce by time $t = 0$, record the common value $X_0^{(j)}$ and set $C_j = 1$. Go to next j or **Step 3** if $j = k$.

Step 3 Check if $\sum_{j=1}^k C_j = k$. If it is true, output the k (NA) draws $\{X_0^{(j)}, j = 1, \ldots, k\}$ and return to **Step 0** for the next k draws. If it is not, set $T_{old} := T_{new}$, $T_{new} := 2T_{new}$, and go back to **Step 1**.

Repeating the above process for m times will produce $n = km$ negatively associated draws from our target density. We emphasize that in **Step 2** all previously generated U_t's, i.e., for $t > -T_{old}$, were reused, as required by the CFTP construction. The use of C_j helps to keep track of jth process coalescence status at $t = 0$, because once it has coalesced there

is no need to go further back. However, unless coalescence took place for all processes, we still need to perform **Step 1** with full k even if $U_t^{(j)}$'s will not be used because $C_j = 1$ for some j's. It is legitimate to only generate antithetic variates using a reduced k, say, k_1, if there are only k_1 processes that have not coalesced. However, one must be aware that this will in general alter the joint distribution of $\{X_0^{(j)}, j = 1, \ldots, k\}$ because generating $\{U_t^{(1)}, \ldots, U_t^{(k_1)}\}$ is not the same as generating $\{U_t^{(1)}, \ldots, U_t^{(k)}\}$ and then just taking the first k_1 components (which, for example, no longer achieve EA). Currently, it is not clear whether such a "dynamic k" scheme, which is less straightforward to program, is beneficial.

We run the above algorithm for four cases given by $N = 16$ together with $p = 1/10, 1/3, 9/20, 1/2$, respectively, each under the three methods given in Section 3. Figure 2 plots the reduction factor $R_k^{(f)} = 1 + (k-1)\rho_k^{(f)}$ as a function of the number of processes k, and the choice of f. The $\rho_k^{(f)}$ was estimated by taking the average of all $\binom{k}{2}$ pair-wise sample correlations between $f(X_0^{(j)})$ and $f(X_0^{(j')})$, $j \neq j'$, where the within-process sample size m was fixed at $15,000$ (and hence $n = km$ increases with k).

The first row of Figure 2 corresponds to $f(x) = x$, and by design, all three methods produced $R_k^{(f)} < 1$ for all choices of k and p. However, the magnitude of reduction varies quite a bit with the methods, with iterative Latin hypercube sampling giving generally the largest reduction, mostly more than 50%. More importantly, except for the case of $p = 0.5$, for which π is perfectly symmetric and thus $k = 2$ is the best choice, there is general evidence that using some $k > 2$ can achieve larger reduction, especially with the iterative hypercube sampling method. The most striking reduction is for $p = 0.1$, where by moving from $k = 2$ to $k = 10$, we shrink the reduction factor R from about 0.9 to less than 0.2.

One worry of using antithetic coupling is that while it may help to reduce variance substantially for monotone f's, it may substantially increase variances for some non-monotone f which can also be of practical interest (e.g., a quadratic function). Whereas theoretically this is always a possibility, we have not observed such cases in all the simulations we have conducted despite the fact that we deliberately looked for such cases by choosing functions that greatly distort the monotonicity. For example, the second row and third row of Figure 2 correspond respectively to $f(x) = (x-2)(x-5)$ and $f(x) = sin(3x)$ (the frequency coefficient 3 was chosen to induce erratic behavior of the sine function on $\{1, 2, \ldots, 16\}$). Although the reductions of variance for these two functions are much smaller or even absent, especially with the first two methods, the largest $R_k^{(f)}$ is essentially statistically indistinguishable from $R = 1$ (i.e., the same as not using antithetic coupling) given the simulation error. That is, although antithetic coupling did not help in some cases, it did not hurt either (compared to the independent implementation). It is somewhat remarkable that even for an erratic function like $f(x) = sin(3x)$, the iterative Latin hypercube sampling method was able to reduce the variance by over 60% when $p = 0.1$ and $k = 10$.

We conclude by emphasizing that the two key messages we remit here, namely (1) going beyond $k = 2$ is in general beneficial and (2) iterative Latin hypercube sampling holds great promise, were also observed in other examples, including forward MCMC algorithms, as detailed in Craiu and Meng (2000). We are currently seeking simple strategies for determining an approximately optimal choice of k for a given problem, as well as other effective methods for generating k antithetic variates for arbitrary values of k. We are also investigating the use of antithetic coupling with Wilson's (2000a) read-once CFTP.

Arvidsen, N.I. and Johnsson, T. (1982). Variance reduction through negative correlation – A simulation study. *J. Statist. Comput. Simulation*, 15:119–127.

Bondesson, L. (1983). Problem 128. *Statistica Neerlandica*, 37:149.

Craiu, R.V. and Meng, X-L. (2000). Multi-process parallel antithetic coupling for forward and backward Markov chain Monte Carlo. *Technical Report*, Department of Statistics, The University of Chicago.

Joag-Dev, K. and Proschan, F. (1983). Negative association of random variables with applications. *Annals of Statistics*, 11:286–295.

Frigessi, A., Gåsemyr, J. and Rue, H. (1999). Antithetic coupling of two Gibbs sampler chains. Preprint, Norwegian University of Science and Technology, Trondheim, Norway.

Hammersley, D.C. and Morton, K.V. (1956). A new Monte Carlo technique: antithetic variates. *Proc. Camb. phil. Soc.*, 52:449–475.

Holmes, C.C. and Mallick, B.K. (1999). Perfect simulation for Bayesian curve and surface fitting. Preprint.

Loh, W-L. (1996). On Latin hypercube sampling. *Annals of Statistics* 24, no. 5, 2058–2080.

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, no. 2, 239–245.

Møller, J. and Nicholls, G. (1999). Perfect simulation for sample-based inference. Department of Mathematical Sciences, Aalborg University, Denmark. Preprint.

Murdoch, D.J. (2000). Exact sampling for Bayesian inference: unbounded state spaces. In Neil Madras, editor, *Monte Carlo Methods - Fields Institute Communications* vol. 26.

Murdoch, D.J and Green, P.J. (1998). Exact sampling from a continuous state space. *Scandinavian Journal of Statistics* 25(3): 483–502.

Murdoch, D.J. and Meng, X-L. (2000). Towards perfect sampling for Bayesian mixtures. *Paper prepared for this volume*.

Owen, A.B. (1992). A central limit theorem for Latin hypercube sampling. *J. Roy. Statist. Soc. B* **54** 541–551.

Propp, J.G. and Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223-252.

Snijders, T.A. (1984) Antithetic variates for Monte Carlo estimation of probabilities. *Statistica Neerlandica*, 38:55–73.

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

Wilson, D.B. (2000). Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In Neil Madras, editor, *Monte Carlo Methods - Fields Institute Communications* vol. 26, 141–176.

Wilson, D. B. (2000a). How to couple from the past using a read-once source of randomness. *Random Structures and Algorithms* 16: 85-113.

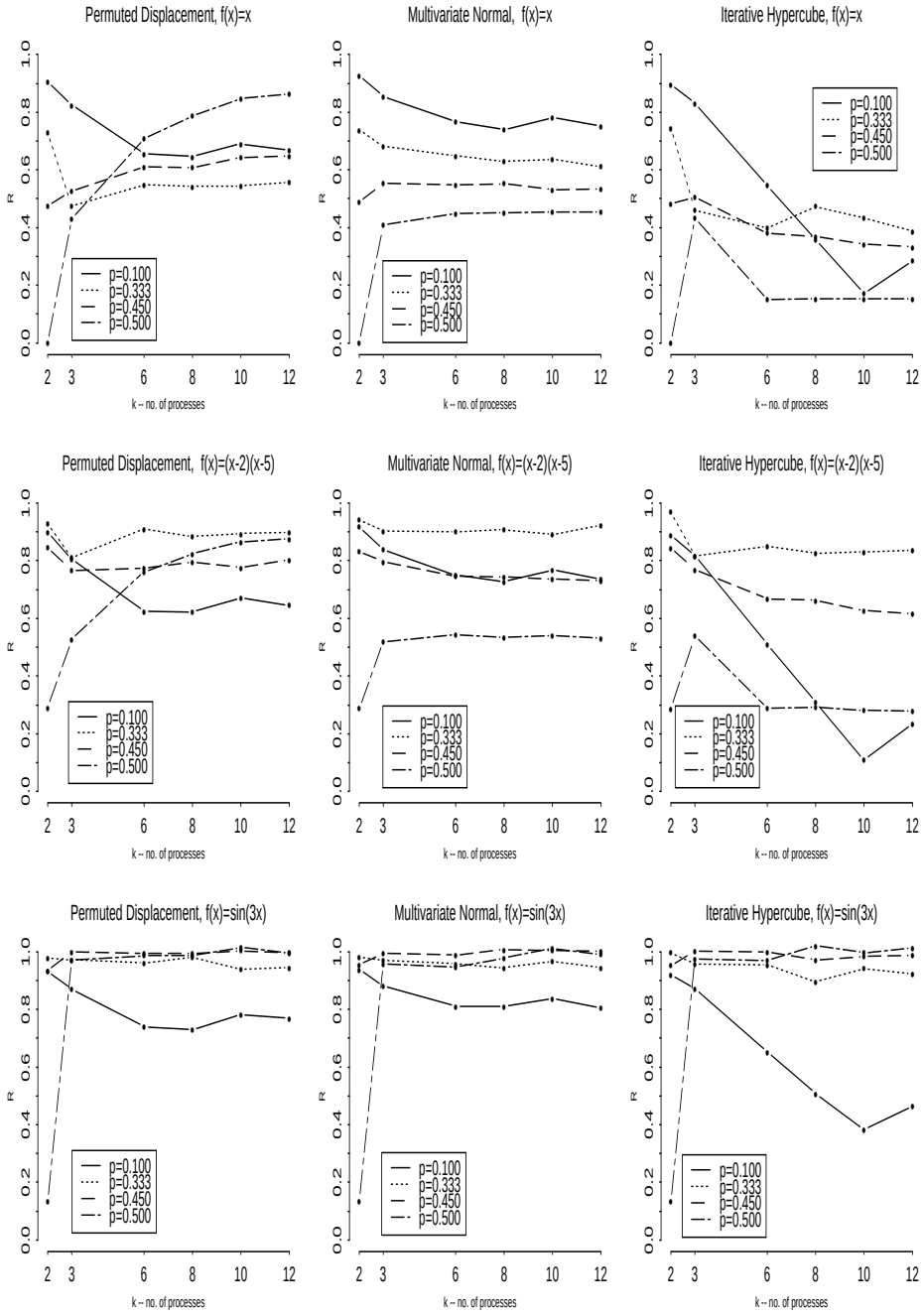

Figure 2. *Variance reduction factor $R_k^{(f)}$ plotted against k, the number of processes, for each of the three methods and for several choices of f and p.*