Communications in
Applied
Mathematics and
Computational
Science

ENSEMBLE SAMPLERS WITH AFFINE INVARIANCE

JONATHAN GOODMAN AND JONATHAN WEARE

vol. 5 no. 1 2010

mathematical sciences publishers

COMM. APP. MATH. AND COMP. SCI.
Vol. 5, No. 1, 2010

ENSEMBLE SAMPLERS WITH AFFINE INVARIANCE

JONATHAN GOODMAN AND JONATHAN WEARE

We propose a family of Markov chain Monte Carlo methods whose performance is unaffected by affine tranformations of space. These algorithms are easy to construct and require little or no additional computational overhead. They should be particularly useful for sampling badly scaled distributions. Computational tests show that the affine invariant methods can be significantly faster than standard MCMC methods on highly skewed distributions.

1. Introduction

Markov chain Monte Carlo (MCMC) sampling methods typically have parameters that need to be adjusted for a specific problem of interest [9; 10; 1]. For example, a trial step-size that works well for a probability density $\pi(x)$, with $x \in \mathbb{R}^n$, may work poorly for the scaled density

$$\pi_\lambda(x) \;=\; \lambda^{-n}\,\pi\,(\lambda x)\,, \tag{1}$$

if λ is very large or very small. Christen [2] has recently suggested a simple method whose performance sampling the density π_λ is independent of the value of λ. Inspired by this idea we suggest a family of many particle (ensemble) MCMC samplers with the more general *affine invariance* property. Affine invariance implies that the performance of our method is independent of the aspect ratio in highly anisotropic distributions such as the one depicted in Figure 1.

An affine transformation is an invertible mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$ of the form $y = Ax + b$. If X has probability density $\pi(x)$, then $Y = AX + b$ has density

$$\pi_{A,b}(y) \;=\; \pi_{A,b}(Ax+b) \;\propto\; \pi(x). \tag{2}$$

Consider, for example, the skewed probability density on $\mathbb{R}^2$ pictured in Figure 1:

$$\pi(x) \;\propto\; \exp\Big(\frac{-(x_1-x_2)^2}{2\epsilon} \;-\; \frac{(x_1+x_2)^2}{2}\Big). \tag{3}$$

MSC2000: 65C05.

Keywords: Markov chain Monte Carlo, affine invariance, ensemble samplers.

Both authors were supported in part by the Applied Mathematical Sciences Program of the U.S. Department of Energy under contract DEFG0200ER25053.

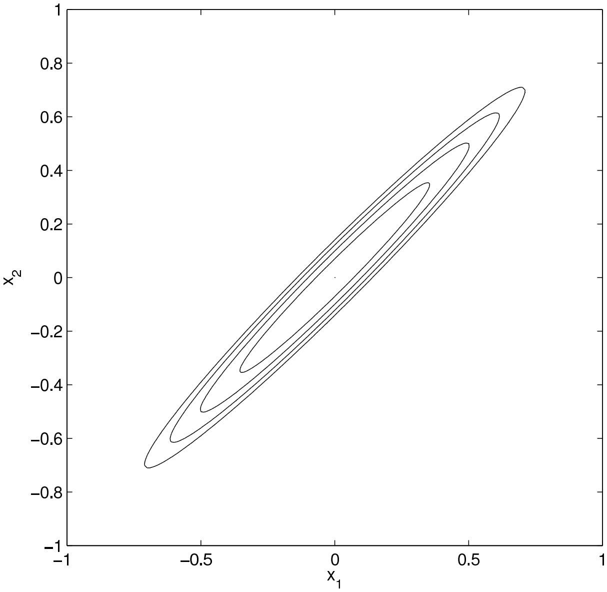


Figure 1. Contours of the Gaussian density defined in expression (3).

Single-variable MCMC strategies such as Metropolis or heat bath (Gibbs sampler) [13; 10] would be forced to make perturbations of order $\sqrt{\epsilon}$ and would have slow equilibration. A better MCMC sampler would use perturbations of order $\sqrt{\epsilon}$ in the $(1, -1)$ direction and perturbations of order one in the $(1, 1)$ direction.

On the other hand, the affine transformation

$$y_1 = \frac{x_1 - x_2}{\sqrt{\epsilon}}, \qquad y_2 = x_1 + x_2,$$

turns the challenging sampling problem (3) into the easier problem

$$\pi_A(y) \;\propto\; e^{-(y_1^2 + y_2^2)/2}. \tag{4}$$

Sampling the well scaled transformed density (4) does not require detailed customization. An affine invariant sampler views the two densities as equally difficult. In particular, the performance of an affine invariant scheme on the skewed density (3) is independent of ϵ. More generally, if an affine invariant sampler is applied to a nondegenerate multivariate normal $\pi(x) \propto e^{-x^t Hx/2}$, the performance is independent of H.

We consider general MCMC samplers of the form $X(t+1) = R(X(t), \xi(t), \pi)$, where $X(t)$ is the sample after t iterations, $\xi(t)$ is a sequence of iid (independent identically distributed) random variables[1], and π is a probability density. General

[1] The probability space for ξ is not important. A Monte Carlo code would typically take $\xi(t)$ to be an infinite sequence of independent uniform [0, 1] random variables.

purpose samplers such as Gibbs samplers have this form. We call such an MCMC algorithm *affine invariant* if, for any affine transformation $Ax + b$,

$$R(Ax+b, \xi(t), \pi_{A,b}) \;=\; AR(x(t), \xi(t), \pi) \;+\; b,$$

for every x and almost all $\xi(t)$.

Less formally, suppose we make two Monte Carlo runs using the same random number generator and seed so that the $\xi(t)$ will be identical for both runs. Suppose one of the runs uses probability density π and starting point $X(0)$. Suppose the other uses $\pi_{A,b}$ and initial point $Y(0) = AX(0) + b$. If the algorithm is affine invariant, the sequences will satisfy $Y(t) = AX(t) + b$. We are not aware of a practical sampler that has this affine invariance property for any general class of densities.

In this paper we propose a family of affine invariant *ensemble* samplers. An ensemble, $\vec{X}$, consists of L *walkers*[2] $X_k \in \mathbb{R}^n$. Since each walker is in $\mathbb{R}^n$, we may think of the ensemble $\vec{X} = (X_1, \dots, X_L)$ as being in $\mathbb{R}^{nL}$. The target probability density for the ensemble is the one in which the walkers are independent and drawn from π, that is,

$$\Pi(\vec{x}) \;=\; \Pi(x_1, \dots, x_L) \;=\; \pi(x_1)\,\pi(x_2)\cdots\pi(x_L). \tag{5}$$

An ensemble MCMC algorithm is a Markov chain on the state space of ensembles. Starting with $\vec{X}(1)$, it produces a sequence $\vec{X}(t)$. The ensemble Markov chain can preserve the product density (5) without the individual walker sequences $X_k(t)$ (as functions of t) being independent, or even being Markov. This is because the distribution of $X_k(t+1)$ can depend on $X_j(t)$ for $j \neq k$.

We apply an affine transformation to an ensemble by applying it separately to each walker:

$$\vec{X} \;=\; (X_1, \dots, X_L) \;\xrightarrow{A,b}\; (AX_1 + b, \dots, AX_L + b) \;=\; (Y_1, \dots, Y_L) \;=\; \vec{Y}. \tag{6}$$

Suppose that $\vec{X}(1) \xrightarrow{A,b} \vec{Y}(1)$ and that $\vec{Y}(t)$ is the sequence produced using $\pi_{A,b}$ in place of π in (5) and the same initial random number generator seed. The ensemble MCMC method is affine invariant if $\vec{X}(t) \xrightarrow{A,b} \vec{Y}(t)$. We will describe the details of the algorithms in Section 2.

Our ensemble methods are motivated in part by the Nelder–Mead [11] simplex algorithm for solving deterministic optimization problems. Many in the optimization community attribute its robust convergence to the fact that it is affine invariant. Applying the Nelder–Mead algorithm to the ill conditioned optimization problem for

[2] Here x_k is walker k in an ensemble of L walkers. This is inconsistent with (3) and (4), where x_1 was the first component of $x \in \mathbb{R}^2$.

the function (3) in Figure 1 is exactly equivalent to applying it to the easier problem of optimizing the well scaled function (4). This is not the case for noninvariant methods such as gradient descent [6].

The Nelder–Mead simplex optimization scheme evolves many copies of the system toward a local minimum (in our terminology: many walkers in an ensemble). A new position for any one copy is suggested by an affine invariant transformation which is constructed using the current positions of the other copies of the system. Similarly, our Monte Carlo method moves one walker using a proposal generated with the help of other walkers in the ensemble. The details of the construction of our ensemble MCMC schemes are given in the next section.

An additional illustration of the power of affine invariance was pointed out to us by our colleague Jeff Cheeger. Suppose we wish to sample X uniformly in a convex body, K (a bounded convex set with nonempty interior). A theorem of Fritz John [8] states that there is a number r depending only on the dimension such that for any convex body K there is an affine transformation $Ax + B$ that makes $\widetilde{K} = AK + b$ well conditioned in the sense that $B_1 \subseteq \widetilde{K}$ and $\widetilde{K} \subseteq B_r$, where B_r is the ball of radius r centered at the origin. An affine invariant sampling method should, therefore, be uniformly effective over all the convex bodies of a given dimension regardless of their shape.

After a discussion of the integrated autocorrelation time as a means of comparing our ensemble methods with single-particle methods in Section 3 we present the results of several numerical tests in Section 4. The first of our test distributions is a difficult two-dimensional problem that illustrates the advantages and disadvantages of our scheme. In the second example we use our schemes to sample from a 101-dimensional approximation to the invariant measure of stochastic partial differential equation. In both cases the affine invariant methods significantly outperform the single site Metropolis scheme. Finally, in Section 5 we give a very brief discussion of the method used to compute the integrated autocorrelation times of the algorithms.

2. Construction

As mentioned in the introduction, our ensemble Markov chain is evolved by moving one walker at time. We consider one step of the ensemble Markov chain $\vec{X}(t) \to \vec{X}(t+1)$ to consist of one cycle through all L walkers in the ensemble. This is expressed in pseudo-code as

```
for $k = 1, \dots, L$
 {
   update:   $X_k(t) \to X_k(t+1)$
 }
```

Each walker X_k is updated using the current positions of all of the other walkers in the ensemble. The other walkers (besides X_k) form the *complementary ensemble*

$$\vec{X}_{[k]}(t) = \{X_1(t+1), \dots, X_{k-1}(t+1), X_{k+1}(t), \dots, X_L(t)\}\,.$$

Let $\mu(d\tilde{x}_k, x_k \mid \vec{x}_{[k]})$ be the transition kernel for moving walker X_k. The notation means that for each $x_k \in \mathbb{R}^n$ and $\vec{x}_{[k]} \in \mathbb{R}^{(L-1)n}$, the measure $\mu(\cdot, x_k \mid \vec{x}_{[k]})$ is the probability measure for $X_k(t+1)$, if $X_k(t) = x_k$ and $\vec{X}_{[k]}(t) = \vec{x}_{[k]}$.

Our single walker moves are based on *partial resampling* [13; 10]. This states that the transformation $\vec{X}(t) \to \vec{X}(t+1)$ preserves the joint distribution Π if the single walker moves $X_k(t) \to X_k(t+1)$ preserve the conditional distribution of x_k given $X_{[k]}$. For our Π (which makes walkers independent), this is the same as saying that $\mu(\cdot, \cdot \mid \vec{x}_{[k]})$ preserves π for all $\vec{x}_{[k]}$, or (somewhat informally)

$$\pi(\tilde{x}_k)\, d\tilde{x}_k \; = \int_{\mathbb{R}^n} \mu(d\tilde{x}_k, x_k \mid \vec{x}_{[k]}) \pi(x_k)\, dx_k.$$

As usual, this condition is achieved using detailed balance. We use a trial distribution to propose a new value of X_k and then accept or reject this move using the appropriate Metropolis Hastings rule [13; 10]. Our motivation is that the distribution of the walkers in the complementary ensemble carries useful information about the density π. This gives an automatic way to adapt the trial move to the target density. Christen [2] uses an ensemble of 2 walkers to generate scale invariant trial moves using the relative positions of the walkers.

The simplest (and best on the Rosenbrock test problem in Section 4) move of this kind that we have found is the *stretch move*. In a stretch move, we move walker X_k using one complementary walker $X_j \in \vec{X}_{[k]}(t)$ (that is, $j \neq k$). The move consists of a proposal of the form (see Figure 2)

$$X_k(t) \to Y = X_j \, + \, Z(X_k(t) - X_j). \tag{7}$$

The stretch move defined in expression (7) is similar to what is referred to as the "walk move" in [2] though the stretch move is affine invariant while the walk move of [2] is not. As pointed out in [2], if the density g of the scaling variable Z satisfies the symmetry condition

$$g\left(\frac{1}{z}\right) = z\, g(z), \tag{8}$$

then the move (7) is symmetric in the sense that (in the usual informal way Metropolis is discussed)

$$\Pr\left(X_k(t) \to Y\right) = \Pr\left(Y \to X_k(t)\right).$$

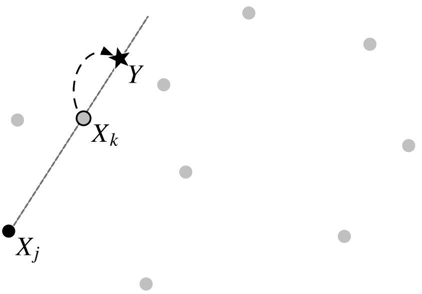


Figure 2. A stretch move. The light dots represent the walkers not participating in this move. The proposal is generated by stretching along the straight line connecting X_j to X_k.

The particular distribution we use is the one suggested in [2]:

$$g(z) \propto \begin{cases} \dfrac{1}{\sqrt{z}} & \text{if } z \in \left[\dfrac{1}{a}, a\right], \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where the parameter $a > 1$ can be adjusted to improve performance.

To find the appropriate acceptance probability for this move we again appeal to partial resampling. Notice that the proposal value Y lies on the ray

$$\left\{ y \in \mathbb{R}^n : \, y - X_j = \lambda \, (X_k(t) - X_j), \;\; \lambda > 0 \right\}.$$

The conditional density of π along this ray is proportional to

$$\| y - X_j \|^{n-1} \, \pi(y).$$

Since the proposal in (7) is symmetric, partial resampling then implies that if we accept the move $X_k(t+1) = Y$ with probability

$$\min \left\{ 1, \, \frac{\| Y - X_j \|^{n-1} \, \pi(Y)}{\| X_k(t) - X_j \|^{n-1} \, \pi(X_k(t))} \right\} = \min \left\{ 1, \, Z^{n-1} \, \frac{\pi(Y)}{\pi(X_k(t))} \right\},$$

and set $X_k(t+1) = X_k(t)$ otherwise, the resulting Markov chain satisfies detailed balance.

The stretch move, and the walk and replacement moves below, define irreducible Markov chains on the space of *general* ensembles. An ensemble is general if there is no lower-dimensional hyperplane (dim $< n$) that contains all the walkers in the ensemble. The space of general ensembles is $\mathcal{G} \subset \mathbb{R}^{nL}$. For $L \geq n+1$, a condition we always assume, almost every ensemble (with respect to Π) is general. Therefore, sampling Π restricted to $\mathcal{G}$ is (almost) the same as sampling Π on all of $\mathbb{R}^{nL}$. It is clear that if $\vec{X}(1) \in \mathcal{G}$, then almost surely $\vec{X}(t) \in \mathcal{G}$ for $t = 2, 3, \ldots$. We assume that $\vec{X}(1)$ is general. It is clear that any general ensemble can be transformed to any other general ensemble by a finite sequence of stretch moves.

The operation $\vec{X}(t) \to \vec{X}(t+1)$ using one stretch move per walker is given by

```
for $k = 1, \dots, L$
 {
   choose $X_j \in \vec{X}_{[k]}(t)$ at random
   generate $Y = X_j + Z(X_k(t) - X_j)$, all $Z$ choices independent
   accept, set $X_k(t+1) = Y$, with probability (7)
   otherwise reject, set $X_k(t+1) = X_k(t)$
 }
```

We offer two alternative affine invariant methods. The first, which we call the *walk* move, is illustrated in Figure 3. A walk move begins by choosing a subset S of the walkers in $\vec{X}_{[k]}(t)$. It is necessary that $|S| \geq 2$, and that the choice of S is independent of $X_k(t)$. The walk move offers a proposal $X_k \to X_k + W$, where W is normal with mean zero and the same covariance as the walkers $X_j \in S$.

More formally, let

$$\pi_S(x) = (1/\,|S|) \sum_{X_j \in S} \delta(x - X_j)$$

be the empirical distribution of the walkers in S. Given S, the mean of a random variable $X \sim \pi_S$ is

$$\overline{X}_S = \frac{1}{|S|} \sum_{X_j \in S} X_j .$$

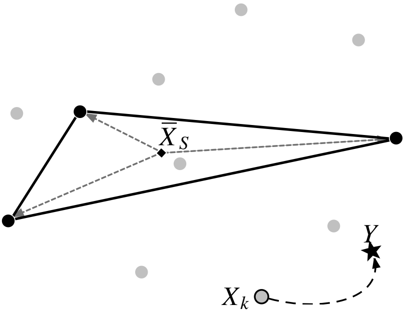


Figure 3. A walk move. The dots represent the ensemble of particles. The dark ones represent the walkers in $\vec{X}_S$. The diamond inside the triangle represents the sample mean $\overline{X}_S$. The proposed perturbation has covariance equal to the sample covariance of the three dark dots. The perturbation is generated by summing random multiples of the arrows from $\overline{X}_S$ to the vertices of the triangle.

The covariance is

$$C_S = \frac{1}{|S|} \sum_{X_j \in S} (X_j - \overline{X}_S)(X_j - \overline{X}_S)^t. \tag{10}$$

It is easy to check that if the Z_j are univariate standard normals then, conditioned on S,

$$W = \sum_{X_j \in S} Z_j (X_j - \overline{X}_S) \tag{11}$$

is normal with mean zero and covariance (10). The proposed trial move is

$$X_k(t) \to X_k(t) + W.$$

The random variable (11) is symmetric in that

$$\Pr(X \to X + W = Y) = \Pr(Y \to Y - W = X).$$

Therefore, we insure detailed balance by accepting the move $X_k(t) \to X_k(t) + W$ with the Metropolis acceptance probability

$$\min\left\{ 1,\ \frac{\pi\ ((X_k(t) + W)}{\pi\ ((X_k(t))} \right\}.$$

The walk move ensemble Monte Carlo method just described clearly is affine invariant in the sense discussed above. In the invariant density $\Pi(\vec{x})$ given by (5), the covariance matrix for W satisfies (an easy check)

$$\mathrm{cov}\,[W] \ \propto\ \mathrm{cov}_\pi\,[X].$$

The constant of proportionality depends on σ^2 and $|S|$. If π is highly skewed in the fashion of Figure 1, then the distribution of the proposed moves will have the same skewness.

Finally, we propose a variant of the walk move called the *replacement move*. Suppose $\pi_S(x \mid S)$ is an estimate of $\pi(x)$ using the subensemble $S \subset X_{[k]}(t)$. A replacement move seeks to replace $X_k(t)$ with an independent sample from $\pi_S(x \mid S)$. The probability of an $x \to y$ proposal is $\pi(x)\pi_S(y \mid S)$, and the probability of a $y \to x$ proposal is $\pi(y)\pi_S(x \mid S)$. It is crucial here, as always, that S is the same in both expressions. If $P_{x\to y}$ is the probability of accepting an $x \to y$ proposal, detailed balance is the formula

$$\pi(x)\pi_S(y \mid S) P_{x\to y} = \pi(y)\pi_S(x \mid S) P_{y\to x}.$$

The usual reasoning suggests that we accept an $x \to y$ proposal with probability

$$P_{x\to y} = \min\left\{ 1,\ \frac{\pi(y)}{\pi_S(y \mid S)} \cdot \frac{\pi_S(x \mid S)}{\pi(x)} \right\}. \tag{12}$$

In the case of a Gaussian π, one can easily modify the proposal used in the walk move to define a density $\pi_S(x \mid S)$ that is an accurate approximation to π if L and $|S|$ are large. This is harder if π is not Gaussian. We have not done computational tests of this method yet.

3. Evaluating ensemble sampling methods

We need criteria that will allow us to compare the ensemble methods above to standard *single-particle* methods. Most Monte Carlo is done for the purpose of estimating the expected value of something:

$$A = E_\pi\,[f(X)] = \int_{\mathbb{R}^n} f(x)\pi(x)\,dx, \tag{13}$$

where π is the target density and f is some function of interest.[3] Suppose $X(t)$, for $t = 1, 2, \ldots, T_s$, are the successive states of a single-particle MCMC sampler for π. The standard single-particle MCMC estimator for A is

$$\widehat{A}_s = \frac{1}{T_s}\sum_{t=1}^{T_s} f(X(t)). \tag{14}$$

An ensemble method generates a random path of the ensemble Markov chain $\vec{X}(t) = (X_1(t), \ldots, X_L(t))$ with invariant distribution Π given by (5). Let T_e be the length of the ensemble chain. The natural ensemble estimator for A is

$$\widehat{A}_e = \frac{1}{T_e}\sum_{t=1}^{T_e}\left(\frac{1}{L}\sum_{k=1}^{L} f(X_k(t))\right). \tag{15}$$

When $T_s = LT_e$, the two methods do about the same amount of work, depending on the complexity of the individual samplers.

For practical Monte Carlo, the accuracy of an estimator is given by the asymptotic behavior of its variance in the limit of long chains [13; 10]. For large T_s we have

$$\mathrm{var}\left[\widehat{A}_s\right] \approx \frac{\mathrm{var}_\pi\,[f(X)]}{T_s/\tau_s}, \tag{16}$$

where τ_s is the *integrated autocorrelation time* given by

$$\tau_s = \sum_{t=-\infty}^{\infty} \frac{C_s(t)}{C_s(0)}, \tag{17}$$

[3] Kalos and Whitlock [9] make a persuasive case for making this the definition: *Monte Carlo* means using random numbers to estimate some number that itself is not random. Generating random samples for their own sakes is *simulation*.

and the lag t *autocovariance* function is

$$C_s(t) = \lim_{t'\to\infty} \operatorname{cov}\big[f(X(t'+t))\,,\, f(X(t'))\big]. \tag{18}$$

We estimate τ_s from the time series $f(X(t))$ using a shareware procedure called `Acor` that uses a variant (described below) of the self consistent window method of [7].

Define the ensemble average as $F(\vec{x}) = \frac{1}{L}\sum_{k=1}^{L} f(x_k)$. Then (15) is

$$\widehat{A}_e = \frac{1}{T_e}\sum_{t=1}^{T_e} F(\vec{X}(t)).$$

The analogous definitions of the autocovariance and integrated autocorrelation time for the ensemble MCMC method are

$$\tau_e = \sum_{t=-\infty}^{\infty} \frac{C_e(t)}{C_e(0)},$$

with

$$C_e(t) = \lim_{t'\to\infty} \operatorname{cov}\big[F(\vec{X}(t'+t))\,,\, F(\vec{X}(t'))\big].$$

Given the obvious relation (Π in (5) makes the walkers in the ensemble independent)

$$\operatorname{var}_\Pi[F(\vec{X})] = \frac{1}{L}\operatorname{var}_\pi[f(X)],$$

the ensemble analogue of (16) is

$$\operatorname{var}[\widehat{A}_e] \approx \frac{\operatorname{var}_\pi[f(X)]}{LT_e/\tau_e}\,.$$

The conclusion of this discussion is that, in our view, a sensible way to compare single-particle and ensemble Monte Carlo is to compare τ_s to τ_e. This compares the variance of two estimators that use a similar amount of work. Comparing variances is preferred to other possibilities such as comparing the mixing times of the two chains [4] for two reasons. First, the autocorrelation time may be estimated directly from Monte Carlo data. It seems to be a serious challenge to measure other mixing rates from Monte Carlo data (see, however, [5] for estimating the spectral gap). Second, the autocorrelation time, not the mixing rate, determines the large time error of the Monte Carlo estimator. Practical Monte Carlo calculations that are not in this large time regime have no accuracy.

Of course, we could take as our ensemble method one in which each $X_k(t)$ is an independent copy of a single Markov chain sampling π. The reader can easily convince herself or himself that in this case $\tau_e = \tau_s$ exactly. Thus such an ensemble method with $T_e = LT_s$ would have exactly the same large time variance as the single-particle method. Furthermore with $T_e = LT_s$ the two chains would require exactly the same computation effort to generate. The two methods would therefore be indistinguishable in the long time limit.

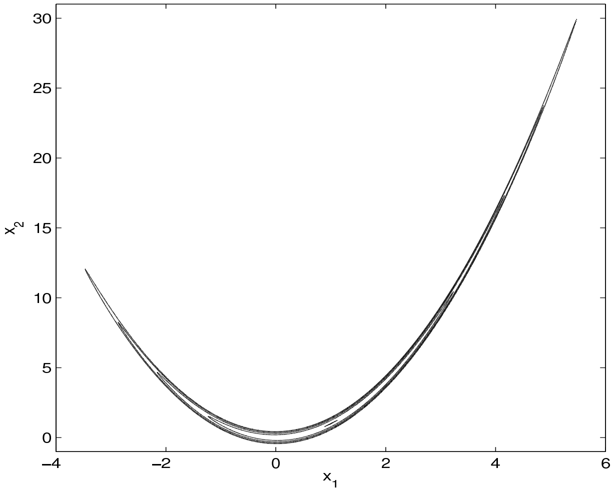
4. Computational tests

In this section we present and discuss the results of computational experiments to determine the effectiveness of our ensemble methods relative to a standard single-particle Markov chain Monte Carlo method. The MCMC method that we choose for comparison is the single site Metropolis scheme in which one cycles through the coordinates of $X(t)$ perturbing a single coordinate at a time and accepting or rejecting that perturbation with the appropriate Metropolis acceptance probability before moving on to the next coordinate. For the perturbations in the Metropolis scheme we choose Gaussian random variables. All user defined parameters are chosen (by trial and error) to optimize performance (in terms of the integrated autocorrelation times). In all cases this results in an acceptance rate close to 30%. For the purpose of discussion, we first present results from tests on a difficult two-dimensional example. The second example is a 101-dimensional, badly scaled distribution that highlights the advantages of our scheme.

4.1. ***The Rosenbrock density.*** In this subsection we present numerical tests on the Rosenbrock density, which is given by[4]

$$\pi(x_1, x_2) \propto \exp\Big(-\frac{100(x_2 - {x_1}^2)^2 + (1-x_1)^2}{20}\Big). \tag{19}$$

Here are some contours of the Rosenbrock density:



[4] To avoid confusion with earlier notation, in the rest of this section (x_1, x_2) represents an arbitrary point in $\mathbb{R}^2$.

method↓ ensemble size →	$f(x_1, x_2) = x_1$				$f(x_1, x_2) = x_2$			
	1	10	100	∞	1	10	100	∞
Metropolis	163	–	–	–	322	–	–	–
stretch moves	–	19.4	8.06	8.71	–	67.0	18.4	23.5
walk moves, $\lvert S\rvert = 3$	–	46.4	19.8	18.6	–	68.0	44.2	47.1

Table 1. Autocorrelation times (multiplied by 10^{-3}) with the functionals $f(x_1, x_2) = x_1$ and $f(x_1, x_2) = x_2$ for single-particle isotropic Metropolis and the chains generated by the two ensemble methods. The ensemble methods with ensemble size $L = \infty$ generate complementary walkers by exact sampling of the Rosenbrock density. The per-step cost of the methods are roughly equivalent on this problem.

Though only two-dimensional, this is a difficult density to sample efficiently as it exhibits the scaling and degeneracy issues that we have discussed throughout the paper. Further the Rosenbrock density has the feature that there is not a single affine transformation that can remove these problems. Thus in some sense this density is designed to cause difficulties for our affine invariant estimators. Of course its degeneracy will cause problems for the single-particle estimator and we will see that the affine invariant schemes are still superior.

Table 1 presents results for the functionals $f(x_1, x_2) = x_1$ and $f(x_1, x_2) = x_2$. The values given should be multiplied by 1000 because we subsampled every Markov chain by 1000. In both cases, the best ensemble sampler has an autocorrelation time about ten times smaller than that of isotropic Metropolis. The walk move method with $|S| = 3$ has autocorrelation times a little more than twice as long as the stretch move method. All estimates come from runs of length $T_s = 10^{11}$ and $T_e = T_s/L$. In all cases we estimate the autocorrelation time using the `Acor` procedure.

To simulate the effect of $L = \infty$ (infinite ensemble size), we generate the complementary X_j used to move X_k by independent sampling of the Rosenbrock density (19). For a single step, this is exactly the same as the finite L ensemble method. The difference comes in possible correlations between steps. With finite L, it is possible that at time $t = 1$ we take $j = 4$ for $k = 5$ (that is, use $X_4(1)$ to help move $X_5(1)$), and then use $j = 4$ for $k = 5$ again at the next time $t = 2$. Presumably, possibilities like this become unimportant as $L \to \infty$. We sample the Rosenbrock density using the fact that the marginal of X is Gaussian, and the conditional density of Y given X also is Gaussian.

Finally, we offer a tentative explanation of the fact that stretch moves are better than walk moves for the Rosenbrock function. The walk step, W, is chosen using three points as in Figure 3; see (11). If the three points are close to X_k, the covariance

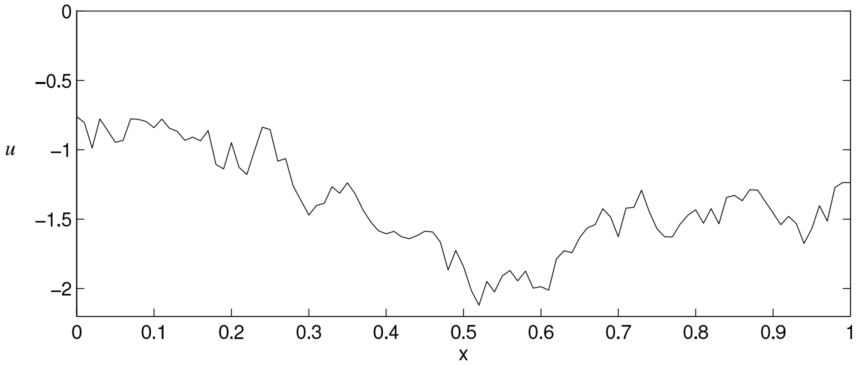


Figure 4. Sample path generated according to π in (22).

of W will be skewed in the same direction of the probability density near X_k. If one or more of the X_m are far from X_k, the simplex formed by the X_m will have the wrong shape. In contrast, the stretch move only requires that we choose one point X_j in the same region as X_k. This suggests that it might be desirable to use proposals which depend on clusters of near by particles. We have been unable to find such a method that is at the same time reasonably quick and has the Markov property, and is even approximately affine invariant. The replacement move may have promise in this regard.

4.2. ***The invariant measure of an SPDE.*** In our second example we attempt to generate samples of the infinite-dimensional measure on continuous functions of [0, 1] defined formally by

$$\exp\Big(-\int_0^1 \tfrac{1}{2}u_x(x)^2 + V(u(x))\,dx\Big), \tag{20}$$

where V represents the double well potential

$$V(u) = (1-u^2)^2.$$

This measure is the invariant distribution of the stochastic Allen–Cahn equation

$$u_t = u_{xx} - V'(u) + \sqrt{2}\,\eta, \tag{21}$$

with free boundary condition at $x = 0$ and $x = 1$ [3; 12]. In these equations η is a space time white noise. Samples of this measure tend to resemble rough horizontal lines found either near 1 or near -1 (see Figure 4).

In order to sample from this distribution (or approximately sample from it) one must first discretize the integral in (20). The finite-dimensional distribution can

method	time
Metropolis	80
stretch moves	5.2
walk moves, $\lvert S\rvert = 3$	1.4

Table 2. Autocorrelation times (multiplied by 10^{-3} with f given in (23) for single-particle Metropolis and the chains generated by the two ensemble methods. The ensemble size is 102. Note that in terms of CPU time in our implementation, the Metropolis scheme is about five times more costly per step than the other two methods. We have not adjusted these autocorrelation times to incorporate the extra computational requirements of the Metropolis scheme.

then be sampled by Markov chain Monte Carlo. We use the discretization

$$\pi(u(0), u(h), u(2h)\dots, u(1)) =$$
$$\exp\biggl(-\sum_{i=0}^{N-1} \frac{1}{2h}(u((i+1)h) - u(ih))^2 + \frac{h}{2}\bigl(V(u((i+1)h) + u(ih))\bigr)\biggr), \qquad (22)$$

where N is a large integer and $h = 1/N$. This distribution can be seen to converge to (20) in an appropriate sense as $N \to \infty$. In our experiments we choose $N = 100$. Note that the first term in (22) strongly couples neighboring values of u in the discretization while the entire path roughly samples from the double well represented by the second term in (22).

For this problem we compare the auto correlation time for the function

$$f(u(0), u(h), \dots, u(1)) = \sum_{i=0}^{N-1} \frac{h}{2}\bigl(u((i+1)h) + u(ih)\bigr), \qquad (23)$$

which is the trapezoidal rule approximation of the integral $\int_0^1 u(x)\,dx$. As before we use $\lvert S\rvert = 3$ for the walk step and $T_e = T_s/L$ where $T_s = 10^{11}$ and $L = 102$. As with most MCMC schemes that employ global moves (moves of many or all components at a time), we expect the performance to decrease somewhat as one considers larger and larger problems. However, as the integrated auto correlation times reported in Table 2 indicate, the walk move outperforms single site Metropolis by more than a factor of 50 on this relatively high-dimensional problem. Note that in terms of CPU time in our implementation, the Metropolis scheme is about 5 times more costly per step than the other two methods tested. We have not adjusted the autocorrelation times in Table 2 to incorporate the extra computational requirements of the Metropolis scheme.

5. Software

Most of the software used here is available on the web (for example, `Acor`). We have taken care to supply documentation and test programs, and to create easy general user interfaces. The user needs only to supply procedures in `C` or `C++` that evaluate $\pi(x)$ and $f(x)$, and one that supplies the starting ensemble $\vec{X}(1)$. We appreciate feedback on user experiences.

The `Acor` program for estimating τ uses a self consistent window strategy related to that of [7] to estimate (18) and (17). Suppose the problem is to estimate the autocorrelation time for a time series, $f^{(0)}(t)$, and to get an error bar for its mean, $\overline{f}$. The old self consistent window estimate of τ (see (17) and [13]) is

$$\widehat{\tau}^{(0)} = \min\left\{ s \;\middle|\; 1 + 2 \sum_{1 \le t \le Ms} \frac{\widehat{C}^{(0)}(t)}{\widehat{C}^{(0)}(0)} = s \right\}, \tag{24}$$

where $\widehat{C}(t)$ is the estimated autocovariance function

$$\widehat{C}^{(0)}(t) = \frac{1}{T-t} \sum_{t'=1}^{T-t} (f^{(0)}(t') - \overline{f})(f^{(0)}(t+t') - \overline{f}). \tag{25}$$

The window size is taken to be $M = 10$ in computations reported here. An efficient implementation would use an FFT to compute the estimated autocovariance function. The overall running time would be $O(T \ln(T))$.

The new `Acor` program uses a trick that avoids the FFT and has an $O(T)$ running time. It computes the quantities $\widehat{C}^{(0)}(t)$ for $t = 0, \dots, R$. We used $R = 10$ in the computations presented here. If (24) indicates that $M\widehat{\tau} > R$, we restart after a pairwise reduction

$$f^{(k+1)}(t) = \tfrac{1}{2}\left(f^{(k)}(2t) + f^{(k)}(2t+1) \right).$$

The new time series is half as long as the old one and its autocorrelation time is shorter. Repeating the above steps (25) and (24) successively for $k = 1, 2, \dots$ gives an overall $O(T)$ work bound. Of course, the (sample) mean of the time series $f^{(k)}(t)$ is the same $\overline{f}$ for each k. So the error bar is the same too. Eventually we should come to a k where (24) is satisfied for $s \le R$. If not, the procedure reports failure. The most likely cause is that the original time series is too short relative to its autocorrelation time.

6. Conclusions

We have presented a family of many particle ensemble Markov chain Monte Carlo schemes with an affine invariance property. Such samplers are uniformly effective on problems that can be rescaled by affine transformations to be well conditioned. All Gaussian distributions and convex bodies have this property. Numerical tests indicate that even on much more general distributions our methods can offer significant performance improvements over standard single-particle methods. The computational cost of our methods over standard single-particle schemes is negligible.

Acknowledgments

Our work grew out of discussions of a talk by Colin Fox at the 2009 SIAM meeting on Computational Science. We are grateful to our colleagues Jeff Cheeger and Esteban Tabak for useful comments and suggestions.

References

[1] C. Andrieu and J. Thoms, *A tutorial on adaptive MCMC*, Stat. Comput. **18** (2008), 343–373.

[2] J. Christen, *A general purpose scale-independent MCMC algorithm*, technical report I-07-16, CIMAT, Guanajuato, 2007.

[3] G. Da Prato and J. Zabczyk, *Ergodicity for infinite-dimensional systems*, London Math. Soc. Lecture Note Series, no. 229, Cambridge University Press, Cambridge, 1996. MR 97k:60165 Zbl 0849.60052

[4] P. Diaconis and L. Saloff-Coste, *What do we know about the Metropolis algorithm?*, J. Comput. System Sci. **57** (1998), no. 1, 20–36. MR 2000b:68094 Zbl 0920.68054

[5] K. Gade, *Numerical estimation of the second largest eigenvalue of a reversible markov transition matrix*, Ph.D. thesis, Courant Institute, New York University, 2008.

[6] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*, Academic Press, London, 1981. MR 83d:65195 Zbl 0503.90062

[7] J. Goodman and A. Sokal, *Multigrid Monte Carlo method: conceptual foundations*, Phys. Rev. D **40** (1989), no. 6, 2035–2071.

[8] F. John, *Extremum problems with inequalities as subsidiary conditions*, Studies and essays presented to R. Courant on his 60th birthday (K. O. Friedrichs, O. E. Neugebauer, and J. J. Stoker, eds.), Interscience, New York, 1948, pp. 187–204. MR 10,719b Zbl 0034.10503

[9] M. H. Kalos and P. A. Whitlock, *Monte Carlo methods*, 2nd ed., Wiley, 2008. MR 2503174 Zbl 1170.65302

[10] J. S. Liu, *Monte Carlo strategies in scientific computing*, Springer Series in Stat., no. 16, Springer, New York, 2001. MR 2002i:65006 Zbl 0991.65001

[11] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J. **7** (1965), 308–313. Zbl 0229.65053

[12] M. G. Reznikoff and E. Vanden-Eijnden, *Invariant measures of stochastic partial differential equations and conditioned diffusions*, C. R. Math. Acad. Sci. Paris **340** (2005), no. 4, 305–308. MR 2121896 Zbl 1063.60092

[13] A. Sokal, *Monte Carlo methods in statistical mechanics: foundations and new algorithms*, Functional integration, Plenum, NY, 1997, pp. 131–192. Zbl 0890.65006

Received November 6, 2009.

JONATHAN GOODMAN: goodman@cims.nyu.edu
Courant Institute, New York University, 251 Mercer St., New York, NY 10012, United States

JONATHAN WEARE: weare@cims.nyu.edu
Courant Institute, New York University, 251 Mercer St., New York, NY 10012, United States

Communications in Applied Mathematics and Computational Science

pjm.math.berkeley.edu/camcos

PRODUCTION

apde@mathscipub.org

Paulo Ney de Souza, Production Manager Sheila Newbery, Production Editor Silvio Levy, Senior Production Editor

See inside back cover or pjm.math.berkeley.edu/camcos for submission instructions.

The subscription price for 2010 is US $70/year for the electronic version, and $100/year for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840, USA.

Communications in Applied Mathematics and Computational Science, at Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer-review and production is managed by EditFLOW™ from Mathematical Sciences Publishers.

PUBLISHED BY
mathematical sciences publishers
http://www.mathscipub.org
A NON-PROFIT CORPORATION
Typeset in LaTeX

Communications in Applied Mathematics and Computational Science

vol. 5 no. 1 2010