Probability in the Engineering and Informational Sciences, **17**, 2003, 277–303. Printed in the U.S.A.

SHIFT AND SCALE COUPLING METHODS FOR PERFECT SIMULATION

J. N. Corcoran and U. Schneider

Department of Applied Mathematics
University of Colorado
Boulder, CO 80309-0526
E-mail: corcoran@colorado.edu,
Uli.Schneider@colorado.edu

We describe and develop a variation on a layered multishift coupler due to Wilson that uses a slice sampling procedure to allow one to obtain potentially common draws from two different distributions. Our main application is coupling sample paths of Markov chains for use in perfect sampling algorithms. The coupler is based on slicing density functions and we describe a "folding" mechanism as an attractive alternative to the accept/reject step commonly used in slice sampling algorithms. Applications of the coupler are given to storage models and to auto-gamma distribution sampling.

1. INTRODUCTION

In recent years, adaptations of perfect simulation or "coupling-from-the-past" algorithms have greatly expanded the applicability of such schemes. These methods often rely on the investigator's ability to couple sample paths of a Markov chain. This is often a nontrivial task, and, in the case of a continuous state space, it may depend on the development of tedious minorization conditions as in Corcoran and Tweedie [3] and Murdoch and Green [11]. In this article, as an alternative, we describe and develop a variation on a layered multishift coupler due to Wilson [15] that allows one to obtain potentially common draws from two different continuous distributions.

Simulation of stochastically ordered variables having potentially common realizations is certainly not a new idea; however, as each method is subject to imple-

© 2003 Cambridge University Press 0269-9648/03 $16.00

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

mentational limitations, our aim here is to broaden the reach of a class of techniques. For example, Lindvall [9] gives a method for drawing stochastically ordered variables $X \leq Y$ that maximizes the probability $P(X = Y)$, but this technique requires that we know proper densities f_X and f_Y (as opposed to likelihoods), that we are able to compute $\int (f_X \wedge f_Y)\, dx$, and that we are able to work with the generalized inverse of the distribution function $H(A) = \int_A (f_X \wedge f_Y)\, dx$. These are requirements that we cannot meet in, for example, Section 4.2.1.

In Section 2, we give a "folding coupler" method for drawing a potentially common value from two uniform distributions where the support of one is contained within the support of the other using a *single* random number. We then extend to other distributions using standard slice sampling ideas. The final step in slice sampling, described in detail in Section 2.2, requires that one make uniform draws on subsequently narrower and narrower slices, and this is typically achieved with an accept/reject algorithm that uses an a priori unknown number of values from a random number generator. In a perfect simulation setting, described briefly in Section 2.3, it is essential that one organize and reuse random numbers. It has been our experience that newcomers to perfect simulation have benefited from the streamlined storage and retrieval offered by the folding coupler.

In Section 3, we describe Wilson's layered multishift coupler, which allows one to draw potentially common values from two distributions with the same shape but with different locations (such as two Normal distributions with a common variance but different means) and extend the procedure to noninvertible distributions. In Section 4, we combine both folding and shifting methods to allow potentially common draws from distributions that differ both in shape and location and we apply our results to a simple storage model.

2. THE FOLDING COUPLER

Out goal is to draw a potentially common value from two differently shaped distributions for the purpose of coupling sample paths of a Markov chain. The "folding coupler" requires that only a single random number be generated and is, hence, a simple alternative to a more standard accept/reject algorithm that requires an a priori unknown number of random numbers. This is especially useful in a perfect simulation setting where one stores and reuses random numbers.

2.1. The Uniform Distribution

We begin by defining and illustrating the folding coupler for simple uniform random variables. Suppose that we want to generate a random variable X that is uniformly distributed on the interval $[a, b]$ and another random variable Y that is uniformly distributed on the interval $[c, d]$, where $a \leq c < d \leq b$. In Section 3, we remove the requirement that $[c, d]$ be contained in $[a, b]$ by combining the folding coupler with a shift coupler from Wilson [15].

The folding coupler, illustrated in Figure 1, proceeds as follows:

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

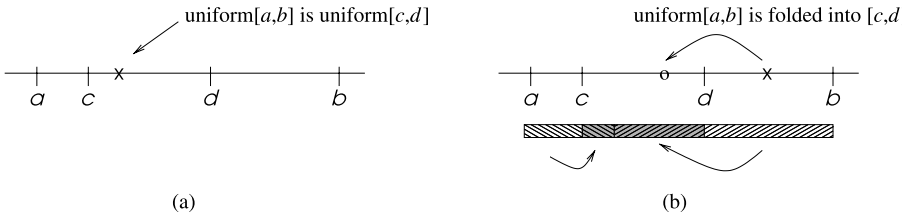


FIGURE 1. (a) Step 2 of the folding coupler; (b) step 3 of the folding coupler.

1. Draw a value $X \sim \text{Uniform}[a,b]$.[1]
2. If $X \in [c,d]$, accept this also as a uniform draw from $[c,d]$ and set $Y = X$.
3. If $X \notin [c,d]$, map X onto the interval $[c,d]$ by "folding it in" according to the function

$$f(x) = f(x;a,b,c,d) = \begin{cases} \dfrac{x-a}{(c-a)+(b-d)}(d-c)+c & \text{if } x \in [a,c) \\ d - \dfrac{b-x}{(c-a)+(b-d)}(d-c) & \text{if } x \in (d,b] \end{cases} \tag{1}$$

and set $Y = f(X)$.

It is routine to show that the resulting Y value will be uniformly distributed on the interval $[c,d]$ and easy to see that $Y = X$ whenever $X \in [c,d]$.

We also have two particular cases holding that will be of interest when we use this technique to generate random transitions for Markov chains:

4. If $c = a$ and $d \le b$ (the interval for Y is contained in and left aligned with the interval for X), then this algorithm will result in draws x and y, where $y \le x$.
5. If $m := (c+d)/2 = (a+b)/2$ (the interval for Y is contained in and centered in the interval for X), then this algorithm will result in a draws x and y, where $|y-m| \le |x-m|$.

2.2. Other Distributions

We now extend the procedure of Section 2.1 to enable us to draw a common value from other nonuniform distributions. One way to sample from a distribution is to sample uniformly from the region under the plot of its density function. To formalize a method for achieving this, we describe the basis for the *slice sampler* approach.

[1] Due to the large number of similar statements in this article, we do not distinguish, except when necessary, between a random variable X and a realization x. For example, this particular statement should be read as "Draw a value $X = x$, where $X \sim \text{Uniform}[a,b]$."

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

Suppose that we can draw a value x for a random variable X with density function $\pi(x) = ch(x)$ where the constant of proportionality c is possibly unknown. We then draw a value $Y = y$ given $X = x$ uniformly over the interval $(0, h(x))$ and, finally, draw a value of X' uniformly from $H(y)$, where $H(y)$ is defined to be the "horizontal slice":

$$H(y) = \{x : h(x) > y\}.$$

Then X' has density $\pi(x)$ since Y has density

$$\begin{aligned} f_Y(y) &= \int_{-\infty}^{\infty} f_{Y|X}(y|x)\pi(x)\,dx \\ &= \int_{-\infty}^{\infty} \frac{1}{h(x)}\,\mathbb{1}_{(0,h(x))}(y)\pi(x)\,dx \\ &= c\int_{H(y)} dx \\ &= c|H(y)| \end{aligned}$$

and so X' has density

$$\begin{aligned} f_{X'}(x) &= \int_{-\infty}^{\infty} f_{X'|Y}(x|y) f_Y(y)\,dy \\ &= \int_{-\infty}^{\infty} \frac{1}{|H(y)|}\,\mathbb{1}_{H(y)}(x)c|H(y)|\,dy \\ &= c\int_0^{h(x)} dy \\ &= ch(x) \\ &= \pi(x). \end{aligned}$$

In the above, $\mathbb{1}_A(\cdot)$ is the indicator function.

The final step of the slice sampler requires that one draws uniformly from a horizontal slice. Using the folding coupler at this step allows us to potentially draw a common value for two differently shaped nonuniform distributions. We illustrate this approach with the Normal distribution.

Assume that our goal is to draw from two Normal distributions $N(\mu, \sigma_1^2)$ and $N(\mu, \sigma_2^2)$, where $\sigma_1 > \sigma_2$. Let $h_1(x)$ and $h_2(x)$ be the likelihood (or density) functions for the $N(\mu, \sigma_1^2)$ and $N(\mu, \sigma_2^2)$ distributions, respectively.

1. Draw $X_1 \sim N(\mu, \sigma_1^2)$ and, in tandem (using the same random numbers), draw the $X_2 \sim N(\mu, \sigma_2^2)$ distribution. In this example, we can simply rescale and let $X_2 = (\sigma_2/\sigma_1)X_1$. See Figure 2a.
2. Draw $U \sim$ Uniform(0,1) and let $Y_1 = Uh_1(X_1)$ and $Y_2 = Uh_2(X_2)$.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

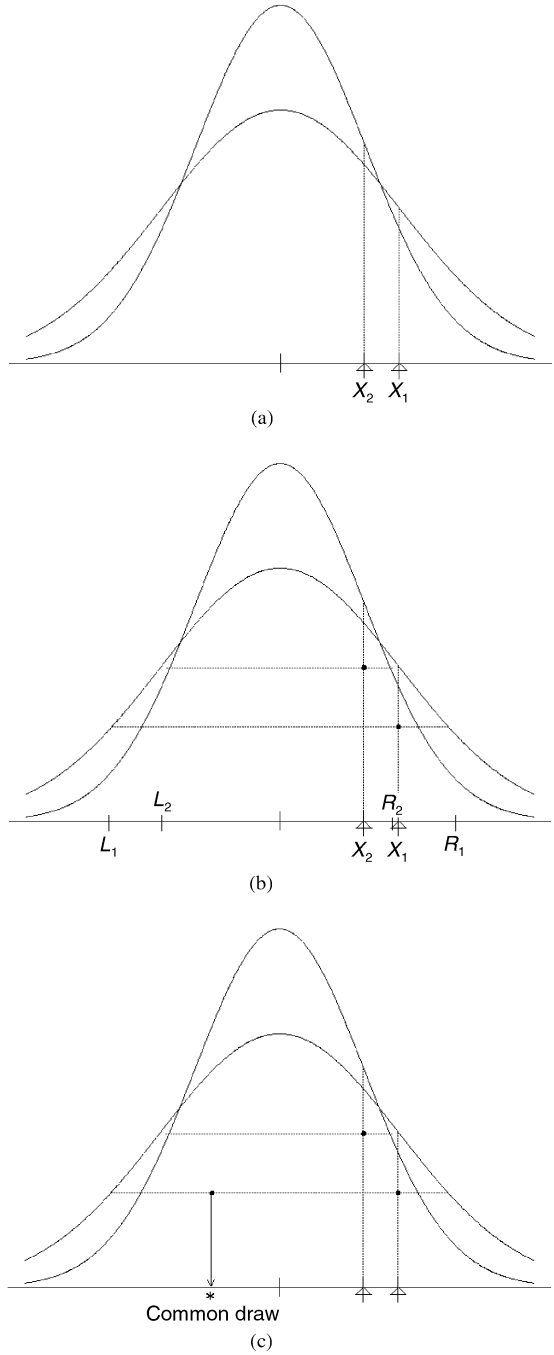


FIGURE 2. A common draw from $N(\mu, \sigma_1^2)$ and $N(\mu, \sigma_2^2)$.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

3. Compute the left and right end points for each horizontal slice:

$$L_1 = -\sigma_1\sqrt{-2\log(\sigma_1 Y_1)} + \mu, \qquad L_2 = -\sigma_2\sqrt{-2\log(\sigma_2 Y_2)} + \mu,$$
$$R_1 = \sigma_1\sqrt{-2\log(\sigma_1 Y_1)} + \mu, \qquad R_2 = \sigma_2\sqrt{-2\log(\sigma_2 Y_2)} + \mu.$$

Note that $L_1 < L_2 < R_2 < R_1$ and that the intervals (L_1, R_1) and (L_2, R_2) are both centered at μ. See Figure 2b.

4. Draw a value $X \sim \text{Uniform}(L_1, R_1)$. This uniform distribution is conditional on the previous steps of this algorithm. Unconditionally though, X has the $N(0, \sigma_1^2)$ distribution. Furthermore, we have the following:

   If $X \in (L_2, R_2)$, then X is uniformly distributed over (L_2, R_2) and, hence, the unconditional distribution of X is $N(0, \sigma_2^2)$ and so we have achieved a common draw from both Normal distributions. See Figure 2c.

   Otherwise, if $X \notin (L_2, R_2)$, we will not get a common draw for both Normal distributions and we finish by drawing from the $N(\mu, \sigma_2^2)$ distribution by folding $X \sim \text{Uniform}(L_1, R_1)$ onto the interval (L_2, R_2) according to (1).

We note that when $X \notin (L_2, R_2)$ at the end of step 4, we could, instead, finish drawing from $N(\mu, \sigma_2^2)$ by independently drawing a uniform on (L_2, R_2), but our intention is to use this construction in a specific way to drive transitions of a Markov chain while preserving Markov and monotonicity properties. Specifically, for purposes of perfect simulation (see Sect. 2.3), we want to observe, at time 0, the state of any possible sample path that has traveled forward from time $-\infty$. We do not want its transitions to be conditional on the transitions of another sample path.

2.3. Examples

There has been considerable recent work on the development and application of "perfect sampling" algorithms that will enable the simulation of the invariant measure π of a Markov chain, either exactly (i.e., by drawing a random sample known to be from π) or approximately, but with a computable order of accuracy. These were sparked by the seminal article of Propp and Wilson [12], and several variations and extensions of this idea have appeared since (see [4–8,10,11]). These ideas have proven effective in areas such as statistical physics, spatial point processes, and operations research, where they provide simple and powerful alternatives to methods based on iterating transition laws, for example.

The essential idea of most of these approaches is to find a random epoch $-T$ in the past such that, if we construct sample paths from every point in the state space starting at $-T$, then all paths will have coupled successfully by time 0. If at $-T$ we were to draw from π, then, intuitively, since the paths couple regardless of the state we choose, the resulting common value at time 0 also has distribution π.

Perfect sampling algorithms can be particularly efficient if the chain is *stochastically monotone* in the sense that paths from lower starting points stay below paths from higher starting points. In this case, one need only couple sample paths from the

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

"top" and "bottom" of the space, as all other paths will be sandwiched in between. It is possible to generalize one step further to monotone chains on an unbounded state space by considering *stochastically dominating* processes to bound the journeys of sample paths.

There are several easy-to-read perfect sampling tutorials available, and we refer the reader to Casella, Lavine, and Robert [2]; a shorter description can be found in Ross [13]. In this article, we only wish to emphasize that the key idea in the search successively further and further back in time for the so-called backward coupling time T requires that one reuse random number streams; that is, if sample paths run forward to time 0 from time -1 using a random number (or random vector) U_{-1} have not coalesced by time 0, then one must go back further, say to time -2, and run paths forward for two steps using a random number U_{-2} and then the *previously used* U_{-1}.

The details of the use of the folding coupler in a backward coupling algorithm is given in the following example.

2.3.1. The attractive auto-exponential distribution. We apply the folding coupler to updates within the Gibbs sampler in order to draw from the density

$$\pi(x_1, x_2) \propto \exp\{-\beta_1 x_1 - \beta_2 x_2 - \beta_{12} x_1 x_2\}, \tag{2}$$

where $\beta_1 > 0, \beta_2 > 0, \beta_{12} < 0, 0 < x_1 < -(\beta_2/\beta_{12})$, and $0 < x_2 < -(\beta_1/\beta_{12})$. We restrict β_{12} to be negative (which then forces restrictions on the ranges of x_1 and x_2) only for ease of algorithm exposition. This restriction is unnecessary and will be removed in Section 2.3.2.

To run the standard Gibbs sampler, we alternate draws from the conditional truncated exponential distributions

$$X_1 | X_2 = x_2 \sim \text{truncated exp}(\text{rate} = \beta_1 + \beta_{12} x_2),$$
$$X_2 | X_1 = x_1 \sim \text{truncated exp}(\text{rate} = \beta_2 + \beta_{12} x_1),$$

where the truncated exponential densities are simply exponential densities truncated to and renormalized over the appropriate support sets. We denote these densities (likelihoods) by $\pi_{x_2}(x)(h_{x_2}(x))$ and $\pi_{x_1}(x)(h_{x_1}(x))$, respectively. To be precise,

$$\pi_{x_2}(x) = \frac{\beta_1 + \beta_{12} x_2}{1 - \exp(-(\beta_1 + \beta_{12} x_2)(-\beta_2/\beta_{12}))} \exp[-(\beta_1 + \beta_{12} x_2)x]$$
$$\text{for } 0 < x < -\frac{\beta_2}{\beta_{12}}$$

and

$$\pi_{x_1}(x) = \frac{\beta_2 + \beta_{12} x_1}{1 - \exp(-(\beta_2 + \beta_{12} x_1)(-\beta_1/\beta_{12}))} \exp[-(\beta_2 + \beta_{12} x_2)x]$$
$$\text{for } 0 < x < -\frac{\beta_1}{\beta_{12}}.$$

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

Since β_{12} is negative, this model is *attractive* in the sense that a small (large) x_1 will produce a small (large) x_2 and vice versa. The Gibbs sampler is then stochastically monotone if we consider the partial ordering

$$(x_1, y_1) \preccurlyeq (x_2, y_2) \Leftrightarrow x_1 \leq x_2 \text{ and } y_1 \leq y_2$$

with a "lowest point" of (0,0) and a "highest point" at $\big(-(\beta_2/\beta_{12}), -(\beta_1/\beta_{12})\big)$; that is, a sample path started at any point (x_1, x_2) in the state space will always stay in between sample paths started from (0,0) and $\big(-(\beta_2/\beta_{12}), -(\beta_1/\beta_{12})\big)$.

Updating Sample Paths

Suppose that at time n, the "lower process," started in state $(l_1^{(0)}, l_2^{(0)}) = (0,0)$, is at some point $(l_1^{(n)}, l_2^{(n)})$ and that the "upper process," started in state $(u_1^{(0)}, u_2^{(0)}) = \big(-(\beta_2/\beta_{12}), -(\beta_1/\beta_{12})\big)$, is at some point $(u_1^{(n)}, u_2^{(n)})$.

We want to draw values for

$$l_1^{(n+1)} \text{ from truncated exp(rate} = \beta_1 + \beta_{12} l_2^{(n)})$$
$$u_1^{(n+1)} \text{ from truncated exp(rate} = \beta_1 + \beta_{12} u_2^{(n)}),$$

and we want these to potentially be the same value in order for the sample paths to coalesce. Suppressing time notation, this means that we want to draw potentially common values for

$$l_1 \text{ from the truncated exponential density } \pi_{l_2}(x) \propto h_{l_2}(x)$$
$$u_1 \text{ from the truncated exponential density } \pi_{u_2}(x) \propto h_{u_2}(x),$$

where both densities have common support $\big(0, -(\beta_2)/(\beta_{12})\big)$.

To achieve these common draws, we

1. Draw, in tandem (using the same random numbers), coupled values l_1' and u_1' directly from $\pi_{l_2}(x)$ and $\pi_{u_2}(x)$ so that $l_1' \leq u_1'$. One could, for example, evaluate the inverses of the distribution functions at a single value U_1 drawn from the Uniform(0,1) distribution. These values give the locations of the initial vertical slices for each density.
2. Draw $V_1 \sim$ Uniform(0,1) and set $Y_l = V_1 h_{l_2}(l_1')$ and $Y_u = V_1 h_{u_2}(u_1')$. These values give the heights of the horizontal slices for each density.
3. Find the right end points of the horizontal slices $R_l = R_l(Y_l)$ and $R_u = R_u(Y_u)$, where $R_l = h_{l_2}^{-1}(Y_l)$ and $R_u = h_{u_2}^{-1}(Y_u)$. Note that the left end points are always at zero in this example and that $R_l \leq R_u$.
4. Draw uniformly from the horizontal slices as follows:
   (i) Draw $W_1 \sim$ Uniform(0,1) and set $W_1' = -(\beta_2/\beta_{12})W_1$. This value is uniformly distributed over $\big(0, -(\beta_2/\beta_{12})\big)$, the *widest* possible horizontal slice.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

(ii) If $W_1' \le R_u$, then W_1' is uniform over $(0, R_u)$, so we set $u_1 = W_1'$. On the other hand, if $W_1' > R_u$, we obtain the outcome u_1 by folding W_1' into the interval $(0, R_u)$ according to

$$u_1 = \frac{W_1' - R_u}{R - R_u} R_u,$$

where $R = -(\beta_2/\beta_{12})$.

(iii) Similarly, if $W_1' \le R_l$, then W_1' is uniform over $(0, R_l)$, so we set $l_1 = W_1'$. On the other hand, if $W_1' > R_l$, we obtain the outcome l_1 by folding W_1' into the interval $(0, R_l)$ according to

$$l_1 = \frac{W_1' - R_l}{R - R_l} R_l,$$

where $R = -(\beta_2/\beta_{12})$.

Note that coupling of the lower and upper sample paths is achieved when $W_1' \le R_l$; that is, coupling is achieved when the low process accepts a horizontal uniform draw without folding. In this case, $l_1 = u_1 = W_1'$, so it is not necessary to run the upper process at all.

We have described only the updates for one of the first components of the lower and upper processes. The second components,

$$l_2 = l_2^{(n+1)} \text{ from truncated exp(rate} = \beta_2 + \beta_{12} l_1^{(n+1)}),$$
$$u_2 = u_2^{(n+1)} \text{ from truncated exp(rate} = \beta_2 + \beta_{12} u_1^{(n+1)}),$$

are updated similarly with independent uniform variates U_2, V_2, and W_2.

We can now describe the algorithm for generating a perfect sample from π.

Folding backward coupling algorithm

1. Draw three independent sequences of Uniform(0,1) variables $U_1^{(n)}$, $V_1^{(n)}$, and $W_1^{(n)}$ for the first component of the Gibbs chain, and draw three independent sequences of Uniform(0,1) variables $U_2^{(n)}$, $V_2^{(n)}$, and $W_2^{(n)}$ for the second component of the Gibbs chain, for $n = 0, -1, -2, \ldots$.
2. For each time $-n = -1, -2, \ldots$, start a lower path $(l_1^{(-n)}, l_2^{(-n)})$ at (0,0).
3. a. For the first component, for $k = n, n-1, \ldots, 1$, update from $l_1^{(-k)}$ to $l_1^{(-k+1)}$ by
   (i) using $U_1^{(-k+1)}$ to draw a value l_1' directly from $\pi_{l_2^{(-k)}}(x)$
   (ii) setting $Y_1 \equiv Y_1^{(-k+1)} = V_1^{(-k+1)} h_{l_2^{(-k)}}(l_1')$ (the height of the vertical slice)
   (iii) setting $R_1 \equiv R_1^{(-k+1)} = h^{-1}_{l_2^{(-k)}}(Y_l)$ (the right end point of the horizontal slice)

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

(iv) setting $W_1' \equiv W_1'^{(-k+1)} = -(\beta_2/\beta_{12})W_1^{(-k+1)}$ and setting

$$l_1^{(-k+1)} = \begin{cases} W_1' & \text{if } W_1' \le R_1 \\ \dfrac{W_1' - R_1}{R - R_1} R_1 & \text{if } W_1' > R_1, \end{cases}$$

where $R = -(\beta_2/\beta_{12})$.

b. For the second component, for $k = n, n-1, \ldots, 1$, update from $l_2^{(-k)}$ to $l_2^{(-k+1)}$ by

(i) using $U_2^{(-k+1)}$ to draw a value l_2' directly from $\pi_{l_1^{(-k)}}(x)$

(ii) setting $Y_2 \equiv Y_2^{(-k+1)} = V_2^{(-k+1)} h_{l_1^{(-k)}}(l_2')$ (the height of the vertical slice)

(iii) setting $R_2 \equiv R_2^{(-k+1)} = h^{-1}_{l_1^{(-k)}}(Y_l)$ (the right end point of the horizontal slice)

(iv) setting $W_2' \equiv W_2'^{(-k+1)} = -(\beta_1/\beta_{12})W_2^{(-k+1)}$ and setting

$$l_1^{(-k+1)} = \begin{cases} W_2' & \text{if } W_2' \le R_2 \\ \dfrac{W_2' - R_2}{R - R_2} R_2 & \text{if } W_2' > R_2, \end{cases}$$

where $R = -(\beta_1/\beta_{12})$.

4. Continue until time $T = \max(T_1, T_2)$, where T_i is the minimum n such that $W_i'^{(-n+1)} \le R_i^{(-n+1)}$.

Simulation Results

We simulated 100,000 draws from the distribution given by (2) using $\beta_1 = 2$, $\beta_2 = 3$, and $\beta_{12} = -1$. Table 1 gives resulting estimates for the probabilities that draws come from selected regions in the plane and Figures 3 and 4 show the esti-

Table 1. Sample Proportion of Paths Ending in Region R

R	$\int\int_R \pi(x_1, x_2)$	95% Confidence Interval	Proportion of Draws in R
$[0,1] \times [0,1]$	0.7340195142	(0.7300817, 0.7356783)	0.73288
$[0,0.5] \times [0,1]$	0.5135615395	(0.5104004, 0.5167227)	0.51280
$[0.2,3] \times [0,0.5]$	0.4811806338	(0.4780206, 0.4843407)	0.48228
$[0,1] \times [1,2]$	0.0547009096	(0.05326273, 0.0561390)	0.05475
$[1,3] \times [0,1.5]$	0.1955191153	(0.1930107980, 0.1980274326)	0.19713

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

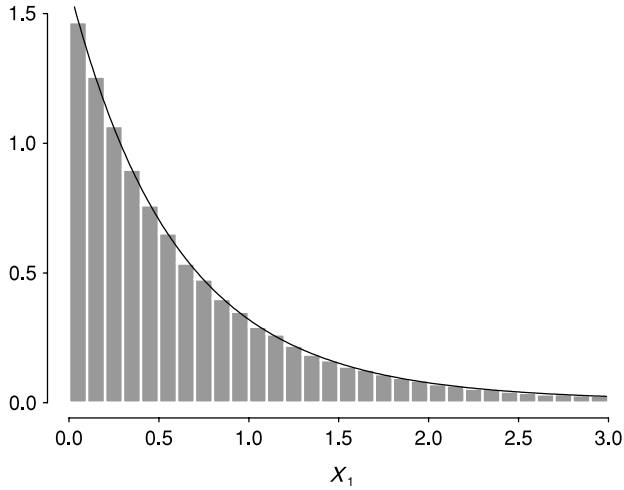


FIGURE 3. 100,000 draws from $\pi(x_1, x_2)$: the marginal distribution of X_1 with the true marginal density superimposed.

mated marginal densities for X_1 and X_2 from 100,000 draws along with curves showing the true marginal densities. The mean backward coupling time in 100,000 draws was 3.44, with a minimum of 1 and a maximum of 26. A histogram of backward coupling times is given in Figure 5.

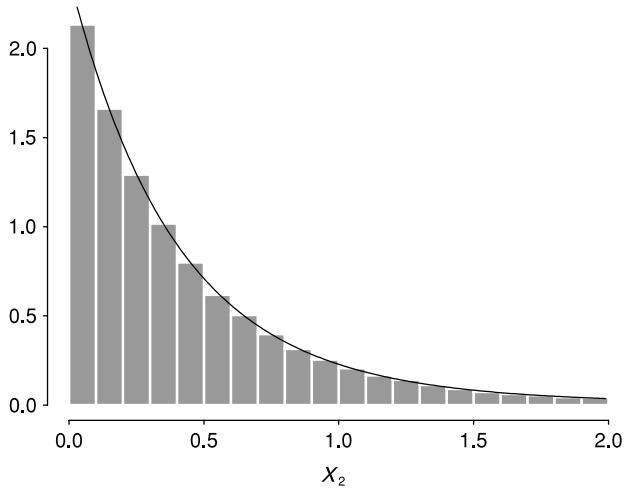


FIGURE 4. 100,000 draws from $\pi(x_1, x_2)$: the marginal distribution of X_2 with the true marginal density superimposed.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

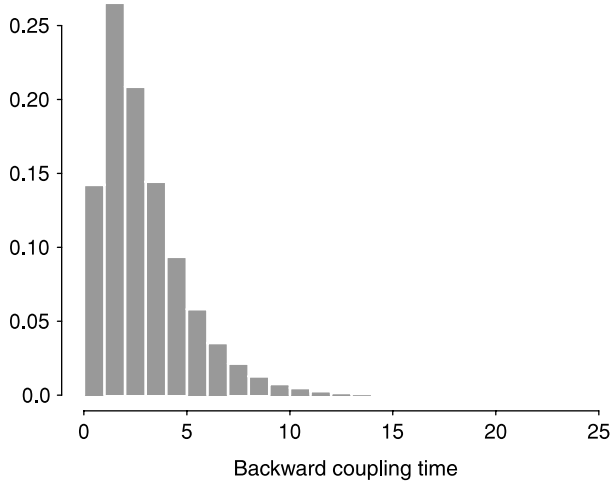


FIGURE 5. 100,000 draws from $\pi(x_1, x_2)$: the backward coupling times.

2.3.2. The repulsive auto-gamma distribution. The following example uses an accept/reject step instead of a folding coupler, which seems to be more appropriate for this problem. It is, however, a natural counterpart to the auto-exponential distribution described in Section 2.3.1, so we choose to add it to this subsection.

We now consider the bivariate distribution

$$\pi(x_1, x_2) \propto x_1^{\alpha_1 - 1} x_2^{\alpha_2 - 1} \exp\{-\beta_1 x_1 - \beta_2 x_2 - \beta_{12} x_1 x_2\}, \tag{3}$$

where α_1, α_2, β_1, β_2, and β_{12} are positive. Note that the conditional densities are given by

$$X_1 | X_2 = x_2 \sim \Gamma(\alpha_1, \beta_1 + \beta_{12} x_2),$$
$$X_2 | X_1 = x_1 \sim \Gamma(\alpha_2, \beta_2 + \beta_{12} x_1),$$

where the scaling parameter appears as follows:

$$X \sim \Gamma(\alpha, \beta) \Rightarrow X \text{ has density } f(x; \alpha, \beta) \propto x^{\alpha - 1} e^{-\beta x} \mathbb{1}_{(0,\infty)}(x).$$

The model can be generalized to a k-variate density and is known as the *auto-gamma model* (see, for instance, Møller [10]). The algorithm described below, based on an idea of Kendall [8], can be easily modified to simulate from a k-variate density; however, for simplicity, we will restrict ourselves to the case $k = 2$.

Repulsive Gibbs Chain

Since $\beta_{12} > 0$, this model is *repulsive* in the sense that a large (small) x_1 will produce a large (small) scaling parameter for the conditional gamma density for X_2 and, therefore, result in a small (large) value for x_2 (and vice versa). If we wish to run

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

a Gibbs sampler in order to draw values from $\pi(x_1, x_2)$, the Gibbs chain will be *stochastically antimonotone* or *repulsive* with respect to the natural partial ordering on $\mathbf{R}^2$ (as defined in Sect. 2.3.1).

Running the Chain Perfectly

To run the chain in a perfect sampling setting, we wish to start at a *highest* point and *lowest* point. In order to maintain an upper and lower process, $u^{(n)} = (u_1^{(n)}, u_2^{(n)})$ and $l^{(n)} = (l_1^{(n)}, l_2^{(n)})$ (n denotes time), respectively, we use the upper process to update the lower chain and the lower process to update the upper chain; that is,

$$\text{draw } u_1^{(n+1)} \sim \Gamma(\alpha_1, \beta_1 + \beta_{12} l_2^{(n)}) \quad \text{and} \quad l_1^{(n+1)} \sim \Gamma(\alpha_1, \beta_1 + \beta_{12} u_2^{(n)}),$$

$$\text{draw } u_2^{(n+1)} \sim \Gamma(\alpha_2, \beta_2 + \beta_{12} l_1^{(n+1)}) \quad \text{and} \quad l_2^{(n+1)} \sim \Gamma(\alpha_2, \beta_2 + \beta_{12} u_1^{(n+1)}).$$

This idea, due to Kendall [8] and implemented by Møller [10], will preserve monotonicity, so that all sample paths will be sandwiched between the lower and upper processes.

Møller [10] showed that once the lower and upper process differ only by an amount ε, they will, with the above-described updates, stay within a distance ε of each other (ε-*coupling*).

The "ε-perfect" sampling algorithm (which results in draws from a target distribution up to any desired accuracy ε) requires that one initializes lower and upper chains at successively more distant times in the past and run the two processes forward to time 0 until the chains at time 0 differ by at most ε (ε-*coupling*). Again, we wish to emphasize that it is necessary to reuse random number streams as described in Section 2.3.

We now describe how to run these sample paths using the slice sampling approach from Section 2.2 in order to achieve actual coupling and, therefore, perfect draws from (3).

A Bounding Process

Since the state space is $(0,\infty) \times (0,\infty)$, (0,0) is clearly a "smallest point" to start the lower process; however, there is no "highest point." However, as mentioned in Section 2.3, we can find a bounding process by observing that the conditional distributions for X_i (which have a fixed shape parameter α_i) have a smallest scale parameter β_i, which will yield a "largest" X_i, so that $D_i \sim \Gamma(\alpha_i, \beta_i)$ is an upper bounding or "dominating" process for the ith component of the Gibbs chain. We therefore initialize the lower and upper processes in the following manner:

$$l^{(0)} = (0,0) \quad \text{and} \quad u^{(0)} = (d_1, d_2).$$

Updating and Coupling Sample Paths

We will now describe in detail how to update the sample paths using slice sampling in order to potentially get common draws. Assume we wish to update the first component of each (upper and lower) process. As with the auto-exponential model in Section 2.3.1, we will draw for both processes simultaneously.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

We choose to describe the case for $\alpha_1 \leq 1$. The case $\alpha_1 > 1$ works analogously, although one has to keep in mind that the shape of the density function is quite different when doing the slice sampling step.

Suppose we need to draw $l_1^{(n+1)} \sim \Gamma(\alpha_1, \beta_l)$ and $u_1^{(n+1)} \sim \Gamma(\alpha_1, \beta_u)$, where $\beta_l = \beta_1 + \beta_{12} u_2^{(n)}$ and $\beta_u = \beta_1 + \beta_{12} l_2^{(n)}$ are the scale parameters defined by the second component of the upper and lower process, respectively. Note that we have $\beta_1 < \beta_u < \beta_l$. Since we do not need to know the constant of proportionally in order to do slice sampling, we use the likelihood $h(x) = (\beta x)^{\alpha-1} e^{-\beta x}$ to define the shape of the curve. For the sake of simplicity, let $h_1(x)$ denote the gamma likelihood with parameters α_1 and β_1 for the dominating process and let $h_{l_1}(x)$ and $h_{u_1}(x)$ denote the likelihoods with parameters α_1 and β_l, and α_1 and β_u, respectively.

Let s_1 and s_2 denote seeds to initialize the random number generator used. (We describe the algorithm for a generic time step and omit the time index for ease of exposition. For example, $s_i = s_i^{(n)}$.)

To update l_1 and l_2

1. Seed the random number generator (RNG) using s_1.
2. Draw (e.g., by an accept/reject method) $X \sim \Gamma(\alpha_1, \beta_1)$ from the dominating process.
3. Draw $U \sim$ Uniform(0,1) and set $Y = Uh_1(X)$.
4. Approximate the exact end point $R = h_1^{-1}(Y)$ defined by the horizontal slice $H(Y) = [0, h_1^{-1}(Y)]$ by R' (where $R < R'$) and draw $X' \sim \text{Uniform}(0, R')$ until $X' \in (0, R)$, updating R' by X' anytime $X' \notin (0, R)$. (Note that $X' \in (0, R)$ can be verified by checking if $h_1(X') < Y$. Also, note that such an approximation is only used as an intermediate step and that the ultimate outcome will not be an approximation.)
5. • If $h_{u_1}(X') < Y$, set $u_1 = X'$.
   • Otherwise, seed the RNG using s_2, set $R'' = X'$, and then perform the following:
   (a) Draw $X'' \sim \text{unif}(0, R'')$.
   (b) If $h_{u_1}(X'') < Y$, set $u_1 = X''$, otherwise set $R'' = X''$ and return to step 5(a).
6. • If $h_{l_1}(X') < Y$, set $l_1 = X'$,
   • Otherwise seed the RNG using s_2, set $R'' = X'$, and then perform the following:
   (a) Draw $X'' \sim \text{unif}(0, R'')$.
   (b) If $h_{l_1}(X'') < Y$, set $l_1 = X''$, otherwise set $R'' = X''$ and return to step 6(a).

Remarks:

1. The above-described update eventually yields potentially common values for u_1 and l_1 and therefore allows the upper and lower processes to eventually couple.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

2. Since it is not a priori known how many random numbers will be used to update the sample paths, it is convenient to store seeds instead. For this algorithm, two seeds are needed for each component and time step.
3. To initialize the upper process, we need to draw the ith component from $\Gamma(\alpha_i, \beta_i)$. In order to run the sample paths preserving the Markov property, the draw for the dominating process must be the X' obtained in steps 1–4 in the above-described algorithm.
4. In the slice sampling described in Section 2.3, we needed to scale (and shift) the starting point X to make it a starting point for the other distribution(s) (say, X_l and X_u) from which we are sampling. In this case, this step is unnecessary since we would have $X_l = (\beta_1/\beta_l)X$ and $X_u = (\beta_1/\beta_u)X$. However, by omitting certain constants in the choice of h, we get that $h_l(X_l) = h_u(X_u) = h(X)$, so that the y value which defines the horizontal slice is the same for all three distributions.
5. In the six steps, we describe only the update in one component for one time step, $-n \to -n+1$. To actually run the perfect-sampling algorithm, this has to be extended for the second component as well as for starting points at successively further distant times in the past until the lower and upper processes have coupled at time 0. The random seeds must be reused for the corresponding time steps.

Simulation Results

Again, we simulated 100,000 draws from the distribution given by (3) using $\alpha_1 = \alpha_2 = 0.5$ and $\beta_1 = 2$, $\beta_2 = 3$, and $\beta_{12} = 1$. Table 2 gives resulting estimates for the probabilities that draws come from particular regions in the plane and Figures 6 and 7 show the estimated marginal densities for X_1 and X_2 along with curves showing the true marginal densities. The mean backward coupling time in 100,000 draws was 1.07, with a minimum of 1 and a maximum of 3.

3. THE LAYERED MULTISHIFT COUPLER

The folding coupler of Section 2 allows us to draw common values from two differently shaped distributions. In this section, we describe the layered multishift coupler of Wilson [15], which allows one to draw common values from distributions

TABLE 2. Sample Proportion of Paths Ending in Region R

R	$\int\int_R \pi(x_1, x_2)$	95% Confidence Interval	Proportion of Draws in R
$[0,0.5] \times [0,0.2]$	0.630553	(0.627561, 0.633545)	0.628010
$[0.2,1] \times [0.5,2]$	0.0200762	(0.019207, 0.020946)	0.020350
$[0.1,\infty] \times [0.2,3]$	0.124523	(0.122477, 0.126569)	0.124140
$[0.2,2] \times [0,1]$	0.347604	(0.344652, 0.350556)	0.347740

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

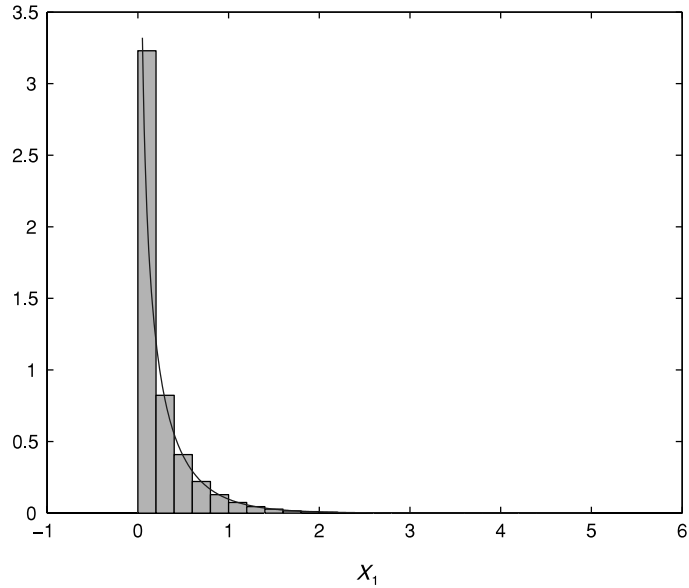


FIGURE 6. 100,000 draws from $\pi(x_1, x_2)$: the marginal distribution of X_1 with the true marginal density superimposed.

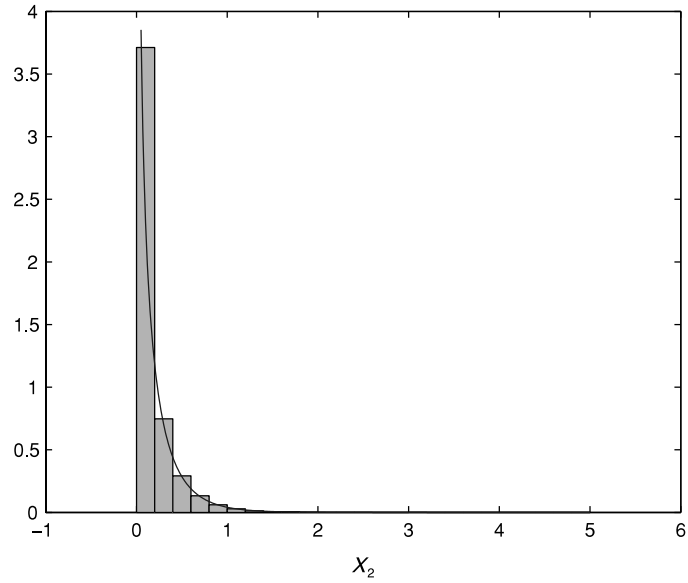


FIGURE 7. 100,000 draws from $\pi(x_1, x_2)$: the marginal distribution of X_2 with the true marginal density superimposed.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

with different location parameters, and we extend these methods to noninvertible distributions. In this section, we combine folding and shifting methods to enable common draws from differently shaped distributions and different (but overlapping) supports.

3.1. The Uniform Distribution

We begin by describing the layered multishift coupler of Wilson [15] for the uniform distribution. Let X be uniformly distributed over the interval (L, R) and consider the mapping

$$g(s) = g_X(s; L, R) = \left\lfloor \frac{s + R - X}{R - L} \right\rfloor (R - L) + X, \tag{4}$$

where $\lfloor \cdot \rfloor$ is the greatest integer or floor function. It is easy to see that for any fixed s, $g(s)$ is uniformly distributed over the interval $(s + L, s + R)$ since

$$\begin{aligned} g(s) &= \left\lfloor \frac{s + R - X}{R - L} \right\rfloor (R - L) + X \\ &= \left[\left(\frac{s + R - X}{R - L} \right) - \mathrm{frac}\left(\frac{s + R - X}{R - L} \right) \right] (R - L) + X \\ &= s + R - \mathrm{frac}\left(\frac{s + R - X}{R - L} \right) (R - L). \end{aligned}$$

(Note that the second equality holds if the argument of the greatest integer function is nonnegative. For $x < 0$ we can proceed to the same end by writing $\lfloor x \rfloor = x - (1 - \mathrm{frac}(x))$.)

Now since $(s + R - X)/(R - L)$ is uniformly distributed over the interval $\big(s/(R - L), [s/(R - L)] + 1\big)$, the fractional part of this expression is uniformly distributed over the interval (0,1). Therefore, from $s + R$, we subtract a quantity that is uniform over $(0, R - L)$, and we have that $g(s)$ is uniform over $(s + L, s + R)$.

As a result of the truncation provided by the greatest integer function, it is possible for different values of s to map to the same $g(s)$. Consequently, we can use this function to draw a common value from both a particular uniform distribution and a shifted version. Additionally, this transformation is monotone in the sense that for $s_1 \leq s_2$, it will result in random variables $g(s_1) \leq g(s_2)$.

3.2. Other Distributions

In this subsection we describe how Wilson [15] extended the procedure of Section 3.1 to enable one to draw a common value from other nonuniform distributions with different location parameters using the same slice sampler approach as in Section 2.2. We again illustrate with the Normal distribution.

Assume that the goal is to draw from two Normal distributions $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$.

Let $h(x)$ denote the $N(0, \sigma^2)$ likelihood function,

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

1. Draw a value X from the $N(0,\sigma^2)$ distribution.
2. Draw a value U from the Uniform(0,1) distribution. Let $Y = Uh(X)$.
3. Compute the end points of the horizontal slice at Y:

$$L = -\sigma\sqrt{-2\log Y}$$
$$R = \sigma\sqrt{-2\log Y}.$$

4. Draw a value $X = x$ from the Uniform(L, R) distribution.
5. Shift this draw by computing $g(\mu_1)$ and $g(\mu_2)$.

$g(\mu_1)$ and $g(\mu_2)$ are values drawn from the $N(\mu_1,\sigma^2)$ and $N(\mu_2,\sigma^2)$ distributions, respectively, and are potentially the same. Again, since $g(\cdot)$ is monotone, this algorithm will result in draws x_1 and x_2, with $x_1 \le x_2$, of the stochastically ordered random variables $X_1 \sim N(\mu_1,\sigma^2)$ and $X_2 \sim N(\mu_2,\sigma^2)$.

3.3. Noninvertible Distributions

The final slicing step in the multishift coupler requires a draw from a horizontal slice, which can be problematic if the distribution is not invertible. Of course, one can use an accept/reject scheme to draw uniformly from an interval with unknown end points, but the true end points are still needed for the shift coupler in (4); that is, if X is distributed uniformly on the interval $[L, R]$ but we only have approximate end points L' and R', where $L' \le L < R \le R'$ at our disposal, then

$$g'(s) = \left\lfloor \frac{s + R' - X}{R' - L'} \right\rfloor (R' - L') + X \not\sim \text{Uniform}(s + L, s + R). \qquad \textbf{(5)}$$

In the following, we assume that L and R are unknown but that one can tell whether a given point is or is not in the interval $[L, R]$. This is the case, for example, when slice sampling from a unimodal density, as in Section 4.2.1. An accept/reject scheme can then be used to sample uniformly from $[L, R]$.

The Shift-and-Patch Algorithm

We begin by considering using (4) to send draws from a uniform distribution on $[L, R]$ to $[L + s, R + s]$. For a specific example, we take $L = 1$, $R = 3$, and $s = 1$. As shown by the shading in Figure 8, the first half of the interval maps to the second

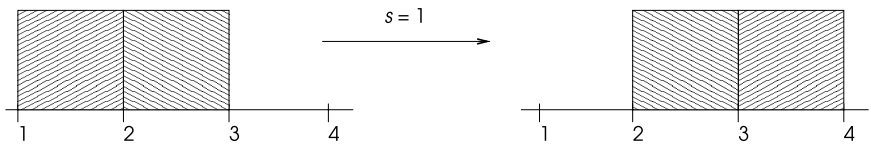


FIGURE 8. $g_X(1;1,3)$ for $X \sim$ uniform(1,3).

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

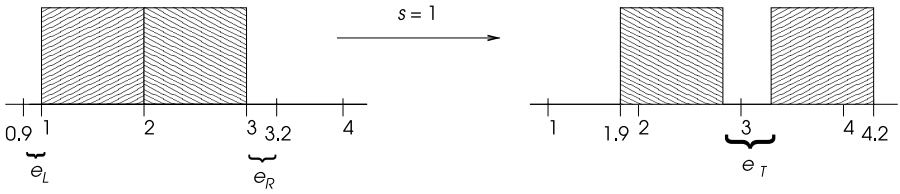


FIGURE 9. $g_X(1;0.9,3.2)$ for $X \sim \text{uniform}(1,3)$.

half, and the second half of the interval maps to the first. Points within each half maintain their ordering.

Figure 9 shows the result using (5) with the approximate end points $L' = 0.9$ and $R' = 3.2$. Here, $e_L = L - L'$ and $e_R = R' - R$ denote, respectively, the (unknown) left and right approximation errors and $e_T = e_L + e_R$ denotes the total end-point approximation error. Points falling outside of the interval $[2,4]$ can be detected (e.g., see step 4 in the algorithm described in Section 2.3.2) and hence rejected, but in order to obtain a draw uniformly distributed on $[2,4]$, we must "fill in" or "patch" the gap with unknown end points in the center.

We achieve this "patch" by reusing the rejected draws falling in $[1.9,2]$ and $[4,4.2]$. These draws are "manually" shifted back by simple subtraction (Fig. 10a) and the result is then reshifted with the map in (5) (Figure 10b).

In general, this shift-and-patch algorithm applies whenever $e_T < |s| < R - L$. We refer the interested reader to the Appendix for details. In Section 4.2.1, we apply the shift-and-patch algorithm to gamma densities.

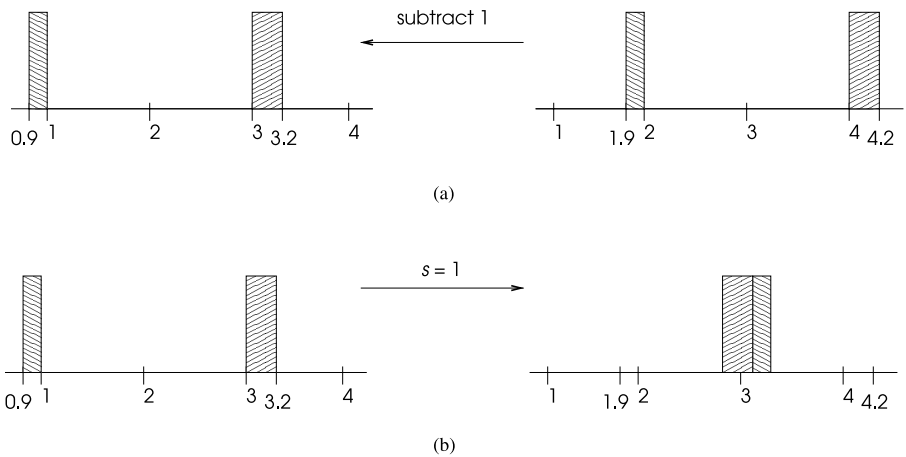


FIGURE 10. Patching the gap.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

4. THE SHIFTING AND FOLDING COUPLER

In this section, we combine the folding coupler with the layered multishift coupler of Wilson [15] to increase the rate of common draws for two distributions with significantly different shapes. In Section 4.2.2, we apply both folding and shifting methods to simultaneously draw from distributions with different (but overlapping) supports.

4.1. Shift and Fold

Suppose that we want a (potentially) common value from draws from $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$. Assume that $\sigma_1 > \sigma_2$. We proceed as follows:

1. Start with a value X_1 drawn from the $N(\mu_1, \sigma_1^2)$ distribution.
2. Also, draw a value X_2 in tandem (using the same random numbers) from the $N(\mu_1, \sigma_1^2)$ distribution; more simply, let $x_2 = \sigma_2\big((x_2 - \mu_1)/\sigma_1\big) + \mu_2$.
3. Use a common draw from Uniform(0,1) to draw heights for the horizontal slices and find the left and right end points L_1 and R_1, and L_2 and R_2; that is, draw a value U from the Uniform(0,1) distribution, let $Y_1 = Uh_1(X_1)$ and $Y_2 = Uh_2(X_2)$, and let

$$L_1 = -\sigma_1\sqrt{-2\log(\sigma_1 Y_1)} + \mu_1, \qquad L_2 = -\sigma_2\sqrt{-2\log(\sigma_2 Y_2)} + \mu_2,$$

$$R_1 = \sigma_1\sqrt{-2\log(\sigma_1 Y_1)} + \mu_1, \qquad R_2 = \sigma_2\sqrt{-2\log(\sigma_2 Y_2)} + \mu_2.$$

4. Follow the procedure of Section 3.1 to draw values uniformly from $[L_1, R_1]$ and $[L_1 + (\mu_2 - \mu_1), R_1 + (\mu_2 - \mu_1)]$.
5. Follow the procedure of Section 2.1 to accept/fold the uniform draw from $[L_1 + (\mu_2 - \mu_1), R_1 + (\mu_2 - \mu_1)]$ as/into a uniform draw on $[L_2, R_2]$.

4.2. Examples

4.2.1. Coupled gamma random variates. In this subsection, we combine shifting, patching, and folding of Sections 3.3 and 4.1 in order to produce (potentially common) draws from two gamma distributions. Again, our notation is

$$X \sim \Gamma(\alpha, \beta) \Rightarrow X \text{ has density } f(x;\alpha,\beta) \propto x^{\alpha-1}e^{-\beta x}\mathbb{1}_{(0,\infty)}(x).$$

We used a "peak-to-peak" shift to produce 100,000 draws from the $\Gamma(3,2)$ and the $\Gamma(2,3)$ distributions; that is, we used the layered multishift coupler (along with the patching algorithm of Section 3.3) to draw from the $\Gamma(3,2)$ distribution with density

$$f(x;3,2) \propto x^2 e^{-2x}\mathbb{1}_{(0,\infty)}(x)$$

and from the shifted density

$$h(x) \propto (x + 5/3)^2 e^{-2(x+5/3)}\mathbb{1}_{(-5/3,\infty)}(x).$$

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

The $\frac{5}{3}$ is the distance between the mode of the $\Gamma(3,2)$ distribution and the mode of the $\Gamma(2,3)$ distribution.

The draws from f and the draws from h are shown in Figures 11a and 11b. Roughly 60% of the draws were common draws. A histogram of these common draws are shown in the Figure 11c. Finally, the draws from h are then folded down to become draws from $\Gamma(2,3)$. These draws are shown in Figure 11d.

4.2.2. A storage model. We now illustrate the application of the folding and shifting coupler for a finite storage system on $[0,K]$ with independent and identically distributed exponential replenishments with mean $1/\mu$ at the arrival times of a Poisson process with rate λ. Excessive input above $K < \infty$ is considered overflow and cannot be saved for future use. Between arrivals, the content is released deterministically at rate $r(u)$. The Markov chain embedded just prior to arrival times satisfies the PASTA property (see Asmussen [1]), which ensures that its stationary distribution is identical to the stationary distribution of the continuous time chain. The assumption of finiteness of the system is not critical, as we may, as in Tweedie and Corcoran [14], adapt the idea (due to Kendall [8]) of an upper bounding random

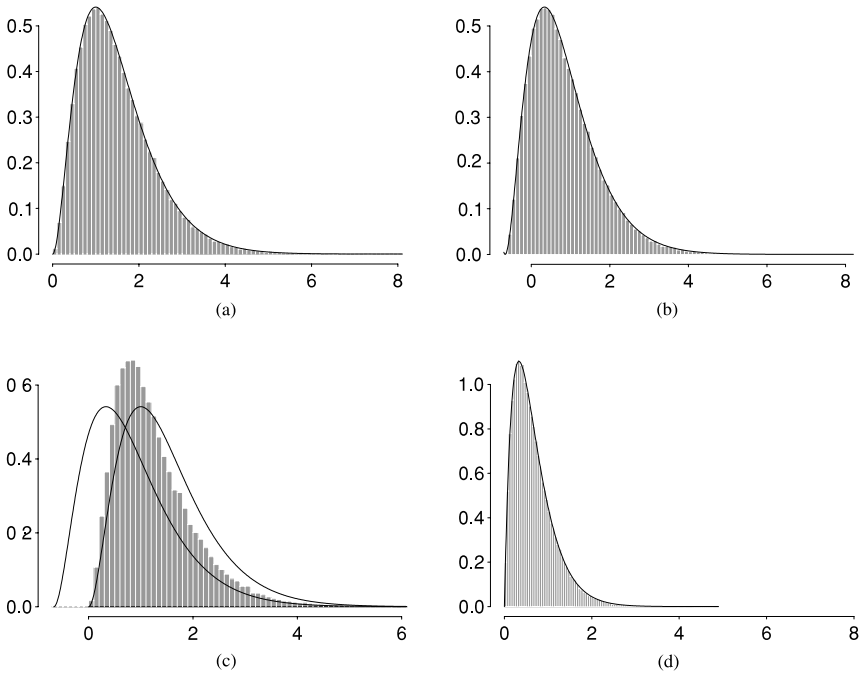


FIGURE 11. (a) Draws from $\Gamma(3,2)$; (b) draws from shifted $\Gamma(3,2)$; (c) common draws for $\Gamma(3,2)$ and shifted version; (d) draws from $\Gamma(2,3)$ (60% in common with (b)).

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

process in order to remove this restriction. Wilson (personal communication) has also developed a system for doing this.

We shall consider specifically the case $r(u) = \beta u$ for $\beta > 0$. For $x \in (0, K]$, it can be shown (Lund, private communication) that the density $\pi(x)$ of the stationary distribution π is given by

$$\pi(x) = \frac{x^{(\lambda\beta^{-1}-1)}e^{-\mu x}}{\int_0^K x^{(\lambda\beta^{-1}-1)}e^{-\mu x}\,dx} \quad \text{for } x \in (0, K] \tag{6}$$

and, in general, the denominator cannot be integrated.

In Corcoran and Tweedie [3] and Tweedie and Corcoran [14], we describe perfect sampling algorithms applicable to this system involving minorizations and chain splitting, but the following described folding and shifting technique is much easier to apply.

The "natural" way to simulate an embedded time step of this storage process is to make a transition from state x to state y as follows:

1. Draw replenishment jump and interarrival values j and t from the exponential distributions with rates μ and λ, respectively.
2. Jump to $z = x + j$.
3. End at $y = ze^{-t}$. (This is the result of the release rate $r(u) = u$.)

Alternatively, we may make this transition as follows:

1. Draw a jump value j from the exponential distribution with rate μ.
2. Jump to $z = x + j$.
3. End at value y drawn from the distribution with density $f_Y^z(y) = (\lambda/z^\lambda)y^{\lambda-1}I_{(0,z)}(y)$.

The density in step 3 is simply that of $Y = ze^{-T}$, where T is exponentially distributed with rate λ.

Now, to attempt to couple two sample paths with current values x_1 and x_2, we make a transition as follows.

1. Draw a jump value j from the exponential distribution with rate μ.
2. Jump the paths to $z_1 = x_1 + j$ and $z_2 = x_2 + j$.
3. Use shifting and folding to draw values (potentially common) from $f_Y^{z_1}$ and $f_Y^{z_2}(y)$.

We consider here only the case $\lambda > 1$. The relative shapes of the two densities in step 3 are shown in Figure 12. We obtain (potentially common) draws from these distributions by first using the shift method described in Section 3.2 to draw values from $f_Y^{z_2}(y)$ and

$$f(y) = \frac{\lambda}{z_2^\lambda}(y + z_2)^{\lambda-1}I_{(z_1-z_2,\,z_1)}(y)$$

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

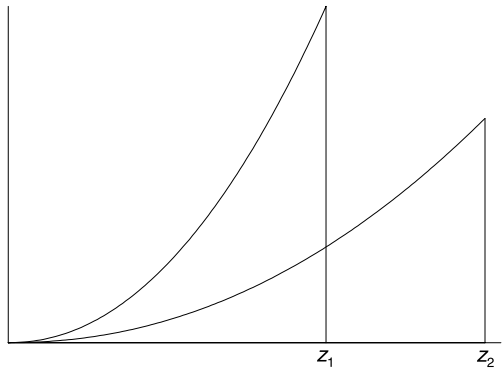


FIGURE 12. Relative shapes of two densities for the storage model.

(one can imagine sliding the wider density to the left until the vertical lines match up) and then folding this shifted wide draw (potentially negative at this point) using (1).

Simulation Results

We simulated 100,000 draws from the storage model with $\lambda = 2$, $\mu = 2$, and $K = 10$. A histogram of these draws along with the true density curve given by (6) is shown in Figure 13.

The mean backward coupling time in 100,000 draws was 12.18, with a minimum of 1 and a maximum of 67. A histogram of backward coupling times is given

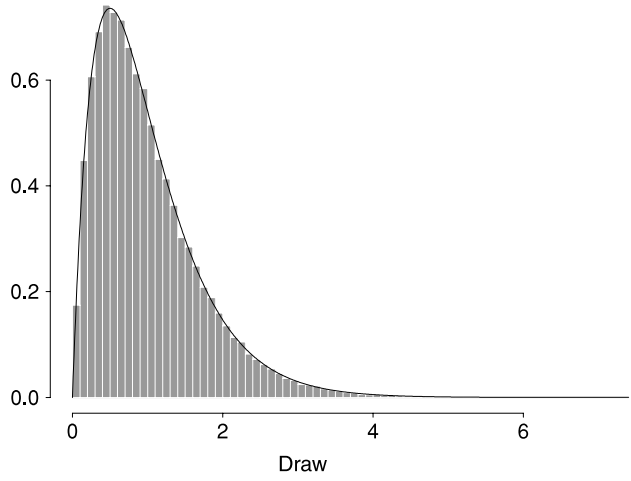


FIGURE 13. Storage model: 100,000 draws from $\pi(x)$ with $\lambda = 2$, $\mu = 2$, and $K = 10$.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

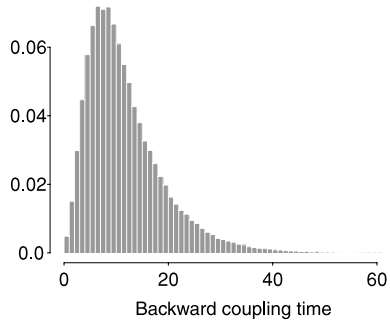


FIGURE 14. Storage model: 100,000 values of the backward coupling time with $\lambda = 2$, $\mu = 2$, and $K = 10$.

in Figure 14. For this model, our shifting and folding coupler was particularly easy to apply. The mean backward coupling time is approximately four times that given by the so-called Harris coupler in Corcoran and Tweedie [3] but was much easier to implement, as the Harris coupler required that we find a minorizing measure for the chain and draw from a complicated "residual" transition law.

Wilson [15] points out that for his layered multishift coupler, the coupling times are inversely related to the lengths of the horizontal slices encountered during sampling. The same can be said for the folding coupler, and it is possible to speed up either algorithm by manipulating the shape of the densities in such a way that we preserve areas over any given interval on the x axis. We refer the interested reader to [15] for details.

With the small backward coupling times given by the shifting and folding coupler, we did not find that the preprogramming setup effort involved in density manipulation or required by the Harris coupler was worth the small reward of efficiency.

References

1. Asmussen, S. (1987). *Applied probability and queues*. New York: Wiley.
2. Casella, G., Lavine, M., & Robert, C. (2000). Explaining the perfect sampler. Working Paper 00-16, State University of New York at Stony Brook/Duke University, Durham, NC.
3. Corcoran, J.N. & Tweedie, R.L. (2001). Perfect sampling of ergodic harris chains. *Annals of Applied Probability* 11(2): 438–451.
4. Fill, J.A. (1998). An interruptible algorithm for perfect sampling via Markov chains. *Annals of Applied Probability* 8: 131–162.
5. Foss, S.G. & Tweedie, R.L. (1998). Perfect simulation and backward coupling. *Stochastic Models* 14: 187–203.
6. Foss, S.G., Tweedie, R.L., & Corcoran, J.N. (1998). Simulating the invariant measures of Markov chains using horizontal backward coupling at regeneration times. *Probability in the Engineering and Informational Sciences* 12: 303–320.
7. Häggström, O., van Liesholt, M.N.M., & Møller, J. (1999). Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Bernoulli* 5: 641–659.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

8. Kendall, W.S. (1998). Perfect simulation for the area-interaction point process. In L. Accardi & C.C. Heyde (eds.), *Probability towards the year 2000*. New York: Springer-Verlag, pp. 218–234.
9. Lindvall, T. (2000). On simulation of stochastically ordered life-length variables. *Probability in the Engineering and Informational Sciences* 14: 1–7.
10. Møller, J. (1999). Perfect simulation of conditionally specified models. *Journal of the Royal Statistical Society, Series B* 61(1): 251–264.
11. Murdoch, D.J. & Green, P.J. (1998). Exact sampling from a continuous state space. *Scandinavian Journal of Statistics* 25: 483–502.
12. Propp, J.G. & Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* 9: 223–252.
13. Ross, S.M. (2002). Coupling from the past. In *Simulation*, 3rd ed. San Diego, CA: Academic Press, pp. 267–269.
14. Tweedie, R.L. & Corcoran, J.N. (2001). Perfect sampling and queueing models. In R.S. Sreenivas & D.L. Jones (eds.), *Proceedings of the 38th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL*, pp. 1125–1134.
15. Wilson, D.B. (2000). Layered multishift coupling for use in perfect sampling algorithms (with a primer on cftp). *Fields Institute Communications* 26: 141–176.

APPENDIX
The Shift-and-Patch Algorithm

The Algorithm

1. *Shift*: Let

$$X' = g_X(s) = \lfloor (s + R' - X)/(R' - L') \rfloor (R' - L') + X.$$

2. If $X' \notin (L + s, R + s)$, then *patch*:

   - Shift back manually by subtracting s
   - Use the truncation map again.

CLAIM: *Let* $e_L = L - L'$ *and* $e_R = R' - R$, *and* $e_T = e_L + e_R$ *and* $l_{int} = R - L$. *Assume that*

$$e_T < |s| < l_{\text{int}}.$$

Then the shift-and-patch algorithm defined in Section 4.2.2 shifts $X \sim \text{Uniform}(L, R)$ to $Z \sim \text{Uniform}(L + s, R + s)$.

PROOF: We will prove the claim by carefully examining where the shift-and-patch step maps numbers. Consider first the *shift step*. We look at the interval $((R' + s - R)/(R' - L'), (R' + s - L)/(R' - L'))$, which is the domain for the floor function used in the truncation map: Assume that $s > 0$ (the case when $s < 0$ works similarly). We observe that $1 \in ((R' + s - R)/(R' - L'), (R' + s - L)/(R' - L'))$ since $(R' + s - X)/(R' - L') = 1$ for $X = L' + s$ and that by assumption, we have that $L = L' + e_L < L' + s < L' + l_{\text{int}} = R$ so that $X = L' + s \in (L, R)$.

Since the interval has length $(R - L)/(R' - L') = l_{\text{int}}/(l_{\text{int}} + e_T) < 1$, and $1 \in ((R' + s - R)/(R' - L'), (R' + s - L)/(R' - L'))$, the floor function in the truncation map can only take on values in {0,1}. In fact, we have

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

$$\left\lfloor \frac{R' + s - X}{R' - L'} \right\rfloor = \begin{cases} 0, & X \in (L' + s, R) \\ 1, & X \in (L, L' + s). \end{cases}$$

With this information, the truncation map in the shift step can be written as

$$g_X(s) = \begin{cases} X, & X \in (L' + s, R) \\ X + (R' - L'), & X \in (L, L' + s). \end{cases}$$

It is obvious that $g_X(s)$ maps $(L' + s, R)$ uniformly onto $(L' + s, R)$. The second interval will simply be shifted by the constant amount of $R' - L'$. To see where the second interval will be mapped to exactly, we evaluate $g_X(s)$ at the corresponding end points L and $L' + s$: We find that $g_L(s) = L + (R' - L') = R + (R' - R) + (L - L') = R + e_T$ and $\lim_{X \to (L'+s)^-} g_X(s) = L' + s + (R' - L') = R' + s$, so that the shift step can be summarized as uniformly mapping the following intervals onto:

$$(L' + s, R) \to (L' + s, R),$$

$$(L, L' + s) \to (R + e_T, R' + s).$$

Note that this leaves us with the unwanted intervals $(L' + s, L + s)$ and $(R + s, R' + s)$, as well as a gap of size e_T in the middle of the interval (see Fig. A1 for illustration).

We will now show that the *patch step* sticks these unwanted intervals outside $(L + s, R + s)$ right into this gap. In the patch step, we first shift back "manually," so that the domain for the truncation map becomes $(L', L) \cup (R, R')$. Since

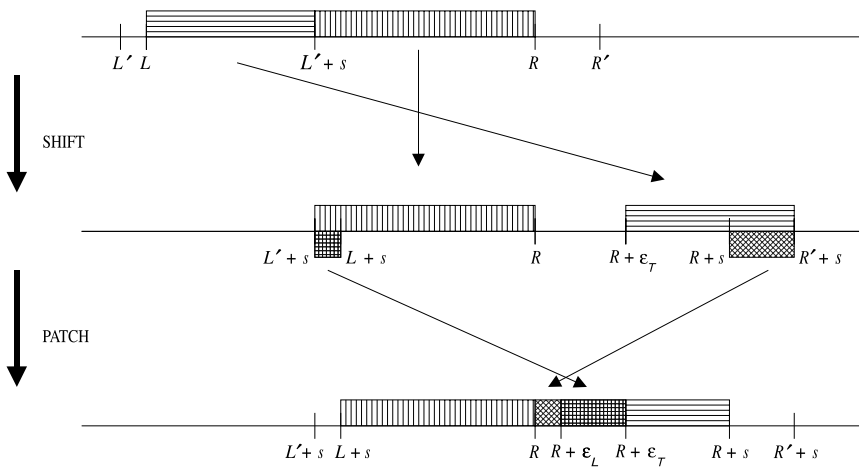


FIGURE A1. Patching the gap.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press

$$\left\lfloor \frac{R' + s - X}{R' - L'} \right\rfloor = \begin{cases} 0, & X \in (R, R') \\ 1, & X \in (L', L), \end{cases}$$

the truncation map for these intervals can be written

$$g_X(s) = \begin{cases} X, & X \in (R, R') \\ X + (R' - L'), & X \in (L', L), \end{cases}$$

which shows that the patch step uniformly maps the following intervals onto:

$$(R + s, R' + s) \to (R, R + e_L),$$
$$(L' + s, L + s) \to (R + e_L, R + e_T),$$

which "fills the gap." ■

Remark: Note that the assumption $e_T < |s|$ can be satisfied by choosing small enough increments when searching for the estimated end points L' and R'. The assumption that $|s| < l_{\text{int}}$ is natural in the sense that with $|s| > l_{\text{int}}$ it is impossible to get common draws anyway.

https://doi.org/10.1017/S0269964803173019 Published online by Cambridge University Press