

## A Note on the Folding Coupler

Hörmann, Wolfgang; Leydold, Josef

*DOI:*

[10.57938/25a77265-1578-4ddd-be7c-7f0274d8cf44](https://doi.org/10.57938/25a77265-1578-4ddd-be7c-7f0274d8cf44)

Published: 01/12/2006

*Document Version*

Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for published version (APA):*

Hörmann, W., & Leydold, J. (2006). *A Note on the Folding Coupler*. Research Report Series / Department of Statistics and Mathematics No. 36 <https://doi.org/10.57938/25a77265-1578-4ddd-be7c-7f0274d8cf44>

# A Note on the Folding Coupler

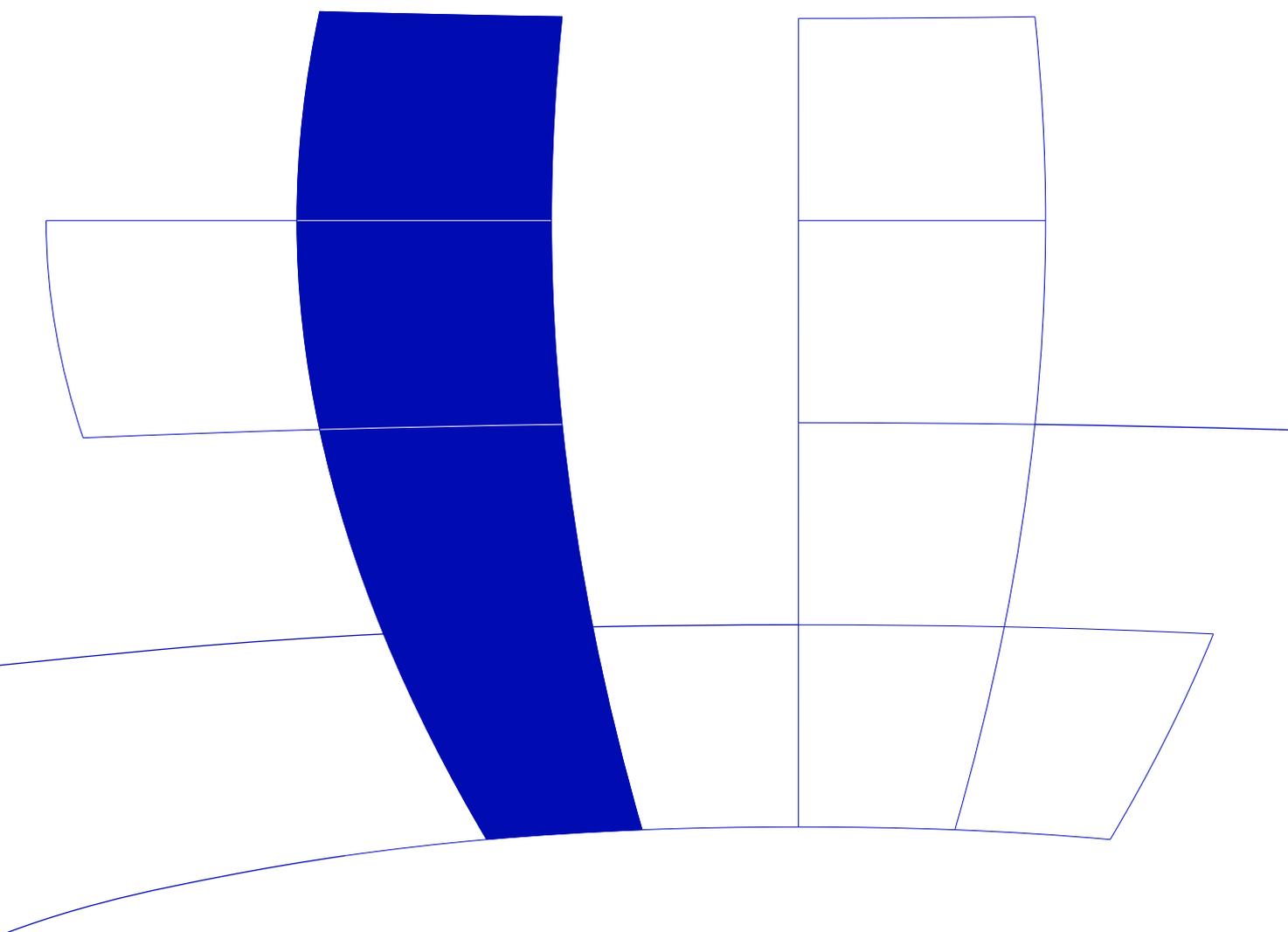
Wolfgang Hörmann, Josef Leydold

Department of Statistics and Mathematics  
Wirtschaftsuniversität Wien

## Research Report Series

Report 36  
May 2006

<http://statmath.wu-wien.ac.at/>



# A Note on the Folding Coupler \*

Wolfgang Hörmann ([hormannw@boun.edu.tr](mailto:hormannw@boun.edu.tr))

*Department of Statistics and Mathematics, Vienna University of Economics and Business Administration, Augasse 2-6, A-1090 Vienna, Austria, and  
Department of Industrial Engineering, Bogazici University Istanbul, 80810 Bebek-Istanbul, Turkey*

Josef Leydold ([leydold@statistik.wu-wien.ac.at](mailto:leydold@statistik.wu-wien.ac.at))

*Department of Statistics and Mathematics, Vienna University of Economics and Business Administration, Augasse 2-6, A-1090 Vienna, Austria*

May 11, 2006

**Abstract.** Perfect Gibbs sampling is a method to turn Markov Chain Monte Carlo (MCMC) samplers into exact generators for independent random vectors. We show that a perfect Gibbs sampling algorithm suggested in the literature is not always generating from the correct distribution.

**Keywords:** Markov chain Monte Carlo method, perfect Gibbs sampling, coupling from the past

## 1. Introduction

Markov Chain Monte Carlo (MCMC) samplers are very powerful methods for drawing random samples from quite arbitrary distributions. In particular they are used in the case of simulations that invoke high dimensional integrals. However, as they produce dependent random variables (vectors) they require a convergence assessment. To overcome this problem Propp and Wilson (1996) suggested so called *perfect sampling algorithms* that allow to decide exactly, when convergence is reached. Although first developed for discrete state spaces perfect sampling also can be applied to Markov chains with state space  $\mathbb{R}^d$  albeit this is not easy, see (Green and Murdoch, 2000; Wilson, 2000; Murdoch, 2000; Murdoch and Meng, 2001).

*Coupling from the past* (CFTP) suggested by Propp and Wilson (1996) is probably the most popular of these perfect sampling algorithms. The main building block of all CFTP algorithms is the *randomizing operation*. It is a deterministic function  $\phi$  taking as input the state  $X_t$  of the chain  $\mathbf{X}$  at time  $t$  and some intrinsic randomness  $U_t$ . The randomizing operation returns the new output state

$$X_{t+1} = \phi(X_t, U_t).$$

---

\* This work was supported by the Austrian Science Foundation (FWF), project no. P16767-N12.

Roughly spoken CFTP starts Markov chains from all possible points of the state space at some time  $-T$  in the past. Using the randomizing operation with the same intrinsic randomness these chains develop in parallel or coalesce. If at time 0 all have coalesced in a single state this state is returned. Otherwise the chains are restarted at some earlier time  $-T' < -T$ ; see e.g. Wilson (2000) for a short tutorial.

Corcoran and Schneider (2003) suggest a new randomizing operation and call it *folding coupler*. They use it to realize simple perfect Gibbs sampling algorithms for different examples. However, it turns out that these algorithms are not always sampling from the desired distribution as the claimed monotonicity property of the folding coupler is not fulfilled.

The paper is organized as following: Section 2 describes the folding coupler and its monotonicity property. In Section 3 the folding coupler is used to compile a Gibbs sampler and it is shown that there are small deviations from the target distribution. Section 4 discusses the difficulties to detect this problem in the design of that sampler.

## 2. The Folding Coupler and its Monotonicity

The folding coupler can be used as a building block for perfect sampling algorithms. Assume we want to generate numbers uniformly in the interval  $(c, d)$  that is contained in  $(a, b)$ . Then for a step of the Markov chain the folding coupler draws a trial  $U$  uniformly distributed in  $(a, b)$ . It is accepted if it falls into the desired interval  $(c, d)$ , otherwise  $U$  is “folded” into the desired interval from the left and from the right. In detail the folding coupler is defined by the function:

$$f(u, a, b, c, d) = \begin{cases} \frac{u-a}{c-a+b-d}(d-c) + c & \text{for } u \in (a, c) \\ u & \text{for } u \in (c, d) \\ d - \frac{b-u}{c-a+b-d}(d-c) & \text{for } u \in (d, b) \end{cases}$$

The folding coupler is a generalization of the multishift and the multiscale couplers introduced by Wilson (2000). Its advantage lies in the fact that it can cope with the multiscale-multishift case; one disadvantage of it is that it does not lead to maximal coalescence probabilities, and thus chains need longer time till coalescence than necessary.

Much more important is the fact that the folding coupler is not always monotone in its last two arguments. This is of greatest practical importance as the monotonicity of a coupler allows that only the maximal and the minimal chain must be generated and stored in a perfect sampling algorithm. The problem can be seen quite easily. For

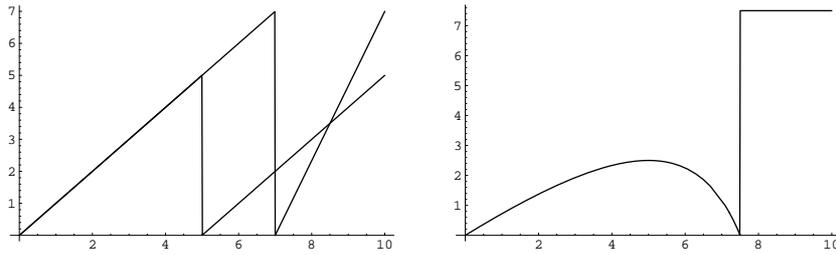


Figure 1. Folding couplers for  $d = 5$  and  $d = 7$  as a function of  $u \in [0, 10]$  (l.h.s.); and folding coupler for a fixed value  $u = 7.5$  as a function of  $d \in [0, 10]$  (r.h.s.). For both sides  $a = 0$ ,  $b = 10$ , and  $c = 0$ .

example, fix  $(a, b) = (0, 10)$  and  $c = 0$ . Then  $f(u, 0, 10, 0, 7)$  is larger than  $f(u, 0, 10, 0, 5)$  for values of  $u$  between 7 and 8. For these values no direct acceptance is possible and thus they are transformed into values close to 0. Figure 1 (l.h.s.) shows the graphs of these two folding couplers for all possible values  $u$  between 0 and 10. It already indicates that the folding coupler is not always monotone in its last argument. This becomes even clearer when we look at the r.h.s. of Figure 1. It shows the result of the folding coupler with  $a = 0$ ,  $b = 10$  and  $c = 0$  for a fixed value of  $u = 7.5$  and  $d$  taking all values between 0 and 10. Clearly this function and thus the folding coupler is not monotone in  $d$ .

*Remark.* There seems to be no simple way to “repair” this problem of the folding coupler. As we need the uniform distribution there must be values that are transformed into 0 if  $u$  is larger than  $c$  and this destroys the monotonicity.

It must also be noted that there are parameter that lead to a monotone folding coupler. Figure 2 shows such an example. For  $a = 0$  and  $b = 10$  the folding coupler for  $(c, d) = (2, 7)$  is below or equal to the folding coupler for  $(c, d) = (2, 8)$ . Nevertheless, it is obvious from the Figure 1 that the general monotonicity of the folding coupler in its last two arguments is not fulfilled.

### 3. Using the Folding Coupler for Perfect Gibbs Sampling

The folding coupler samples from the uniform distribution. It can therefore easily be used to obtain a perfect Gibbs sampler for distributions with uniform full conditional distributions. As example let us consider the two-dimensional uniform distribution on the area between density

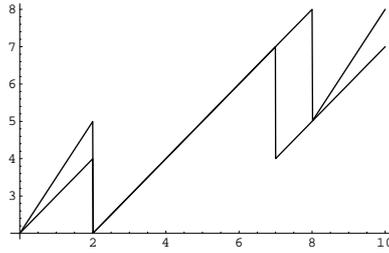


Figure 2. Comparing the folding couplers for  $d = 7$  and  $d = 8$  for  $a = 0$ ,  $b = 10$ ,  $c = 2$  and  $u = (0, 10)$ .

and the  $x_1$ -axis for the standard exponential distribution cut off at 20. That area is is the set

$$S = \{(x_1, x_2) | 0 \leq x_1 \leq 20 \text{ and } 0 \leq x_2 \leq \exp(-x_1)\}.$$

The Gibbs sampler for this uniform distribution and a given starting value  $0 \leq x_2 \leq 1$  draws a random variate  $X_1$  uniformly distributed on  $(0, \min(20, -\log(x_2)))$ ,  $X_1 \sim U(0, \min(20, -\log(x_2)))$ , and then samples  $X_2 \sim U(0, \exp(-X_1))$ . The next steps repeat this pattern and update the previous values of  $X_1$  and  $X_2$ .

To turn this very simple Gibbs sampler into a perfect sampling algorithm it is in theory necessary to run all chains (from all possible starting values) in parallel, i.e. using the same randomness for all of them. Then the CFTP algorithms start farer and farer in the past of the chains till all chains have coalesced to a single value at time 0. Therefore CFTP algorithms need a coupler that allows for coalescence. In other words it must be possible to obtain – when using the same piece of randomness – the same uniform random number for two uniform  $U(\alpha, \beta)$  distributions with different parameter  $\beta$ . This is the case for the folding coupler defined above but not the case if we just generate the uniform distribution with the standard approach.

The density of the exponential distribution is monotone decreasing. Thus the Gibbs sampler has the property, that large values of  $x_1$  lead to small values of  $X_2$  and large values of  $x_2$  lead to small values of  $X_1$  (and vice versa). This property is useful if we can find a coupler that is monotone for the upper border of the interval. (The lower border is always 0 so no monotonicity is required for the lower border for this special example.) Then, for a fixed piece of randomness, the generated variate of  $U(0, a)$  is smaller or equal to the variate of  $U(0, b)$  as long as  $a \leq b$ . Such a monotonicity is important as we know that all possible chains (with arbitrary starting value  $x_2$ ) will always be between the chains starting with  $x_2 = 0$  and starting with  $x_2 = 1$ . It is therefore

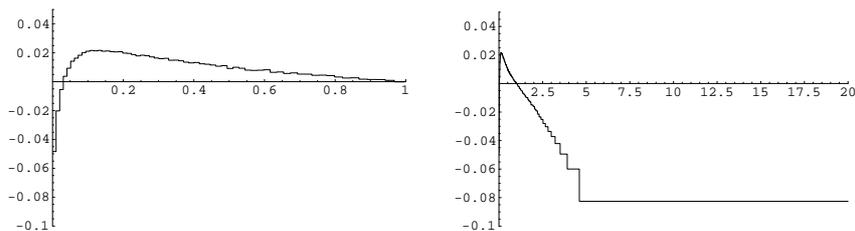


Figure 3. Relative difference between the values generated by the folding coupler and the exact values for the marginal distribution of  $X_1$

enough to run only these two chains as we know that coalescence of all chains is reached when these two chains have coalesced.

Unfortunately the folding coupler is monotone for a large subset of possible parameters, but not for all, as we have seen in Sect. 2. What happens if we use it for Gibbs sampling as if it were a monotone coupler? We tried this for the uniform distribution over the set  $S$  defined above. We ran a CFTP algorithm using the folding coupler and made a chi-square test for the marginal distribution of  $X_1$  which should follow a standard exponential distribution truncated at 20. The chi-square test indicates clear problems for  $n = 10^6$  but no problems for  $n = 10^4$ . Figure 3 displays the relative difference between the histogram of the marginal distribution of  $x_1$  coordinate of the points generated by the Gibbs sampler using the folding coupler and the exact (truncated exponential) distribution. As expected the differences are not large but it clearly indicates that the algorithm does not produce enough values very close to 0 and above 1.

The fact that there are only small deviations from the correct distribution needs some explanations. The reason is simply that the algorithm is a correct MCMC algorithm converging to the given distribution. It is the monotonicity assumption that is not satisfied and thus the “perfect” algorithm sometimes returns wrong values as coalescence was only achieved for the two observed chains, but not for all possible chains. For illustration purposes we ran the naive Gibbs sampler on  $S$  with a fixed number of 16 updates and with both a uniform starting distribution and a fixed starting point  $x_1 = 10$ . For this approximate algorithm the generated distribution is so close to the desired distribution that the chi-square test did not indicate any problems for  $n = 10^6$ .

#### 4. Application to the Autoexponential Distribution

Corcoran and Schneider (2003) explain how the folding coupler can be used to implement the Gibbs sampler for non-uniform full conditional

distributions. Following Wilson (2000) they use the slice sampling idea for this task.

To check if the problem with the folding coupler is also visible for their example we coded the “folding backward coupling algorithm” on p. 285 of (Corcoran and Schneider, 2003). We only changed the coalescence condition as coalescence in one variable automatically implies that the second variable coalesces in the next step for a two variable Gibbs sampler. We tested the resulting algorithm – as in the paper – for the two dimensional autoexponential distribution with density proportional to

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

with the parameter values  $\beta_1 = 2$ ,  $\beta_2 = 3$  and  $\beta_{12} = -1$ .

For the marginal distribution of the first variable we conducted chi-square tests for sample sizes  $n = 10^6$ ,  $10^7$  and  $10^8$  and the number of classes close to  $\sqrt{n}$ . We used a high-quality multiple-recursive uniform random number generator. In this setting problems were only clearly visible for  $n = 10^8$ .

We can therefore conclude that the difference between the generated and the exact distribution is so small for this example that the extensive empirical tests in (Corcoran and Schneider, 2003) did not indicate any problem. Thus the fact that the folding coupler is not monotone was overlooked.

To control our test set-up we applied the same tests to autoexponential random variates generated by naive two-dimensional rejection from a constant hat. The resulting chi-square values were in the expected interval even for sample size  $10^8$ .

## 5. Conclusion

We have shown that the folding coupler does not always fulfill the monotonicity property that is required for the CFTP algorithm proposed by Corcoran and Schneider (2003). We demonstrated that this problem leads to small deviations from the target distribution. Although these deviations are sometimes very small we do not think that the folding coupler algorithm is suitable for practical purposes as it is not clear how close generated distributions are to the desired distributions for new applications.

## References

- Corcoran, J. N. and U. Schneider: 2003, ‘Shift and scale coupling methods for perfect simulation’. *Probability in the Engineering and Informational Sciences* **17**, 277–303.
- Green, P. J. and D. J. Murdoch: 2000, ‘Exact sampling for Bayesian inference: towards general purpose algorithms (with discussion)’. In: J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (eds.): *Monte Carlo Methods*, Vol. 6 of *Bayesian Statistics*. Oxford, pp. 301–321, Oxford University Press.
- Murdoch, D. J.: 2000, ‘Exact sampling for Bayesian inference: unbounded state space’. In: N. Madras (ed.): *Monte Carlo Methods*, Vol. 26 of *Fields Institute Communications*. pp. 111–121, American Mathematical Society.
- Murdoch, D. J. and X.-L. Meng: 2001, ‘Towards perfect sampling for Bayesian mixture priors’. In: *ISBA 2000, Proceedings*. p. ??, ISBA and Eurostat.
- Propp, J. G. and D. B. Wilson: 1996, ‘Exact sampling with coupled Markov chains and applications to statistical mechanics’. *Random Structures and Algorithms* **9**, 223–252.
- Wilson, D. B.: 2000, ‘Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP)’. In: N. Madras (ed.): *Monte Carlo Methods*, Vol. 26 of *Fields Institute Communications*. pp. 141–176, American Mathematical Society.

