Perfect Sampling
2. Other algorithms

Duncan Murdoch
University of Western Ontario

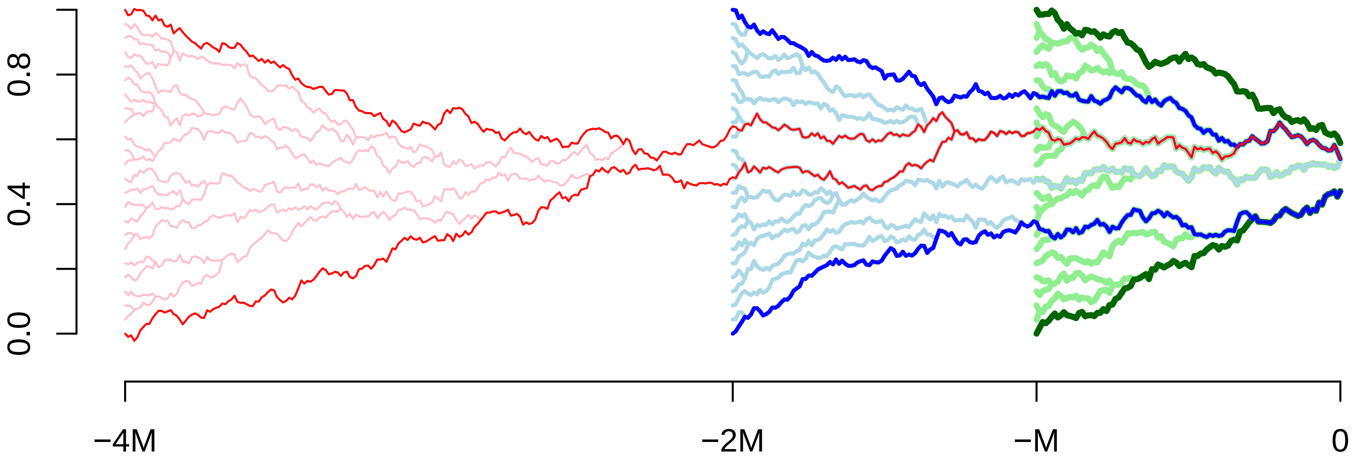
Outline

1. Read-once CFTP
2. Fill's rejection sampler
3. Generalizations

1. A CFTP implementation problem

When M is increased to $2M$, CFTP needs $U_{-2M+1}, \ldots, U_0$, but $U_{-M+1}, \ldots, U_0$ must not change.

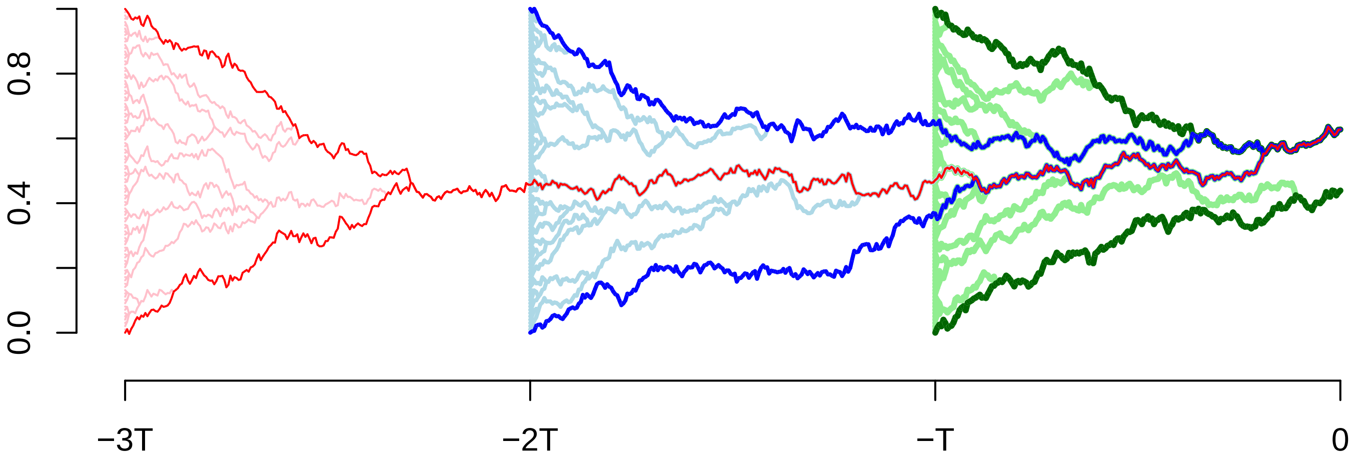
This requires either lots of storage, or tricky programming.



(This figure and later ones use Wilson's (2000a) multishift coupler.)

Read-once CFTP Wilson (2000b)

Do CFTP, but only use each U_t once.



$$
\begin{aligned}
C_{-n} &= \{\text{Paths from } -nT \text{ coalesce by } -(n-1)T\} \\
X^*_{-n} &= \begin{cases} X_{-(n-1)T} & \text{if } C_{-n} \text{ occurred} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

The pairs (C_{-n}, X^*_{-n}) are IID across n.

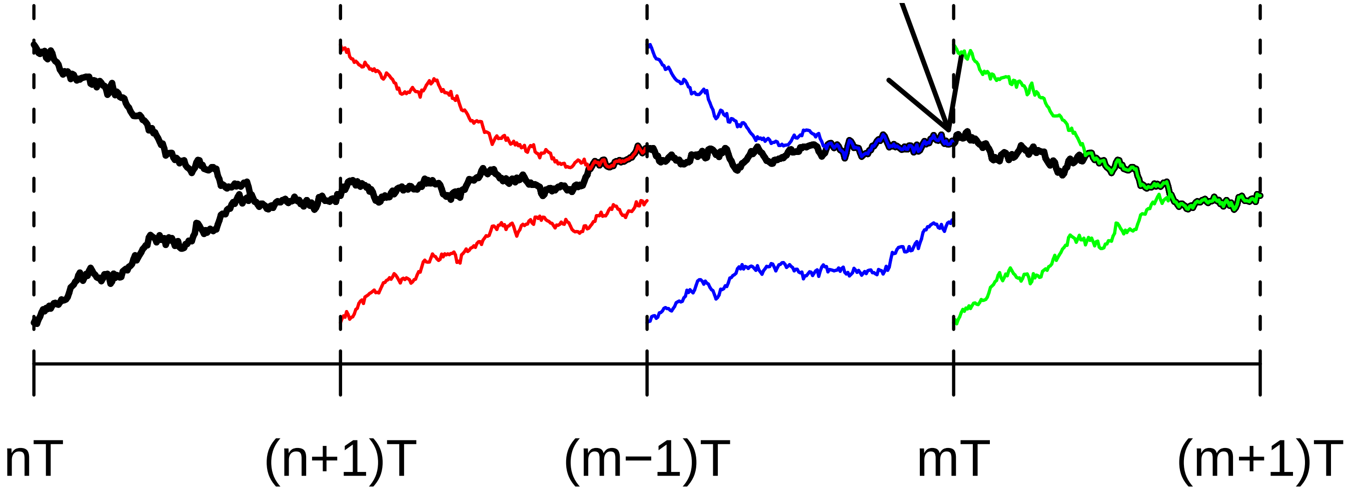
- The independence arises since the pairs depend only on $\{U_t\}_{t=-n+1}^{-(n-1)T}$.
- CFTP(mT) coalesces provided that at least one C_{-n}, $n = 1, \ldots, m$ occurs, so we could make CFTP search backwards for the first previous C_{-n} event. The output value will be the update to $t = 0$ of a path started at X^*_{-n}.
- This may be inefficient: CFTP(mT) may coalesce even if no C_{-n} events occur. If so, it will give the same output as this variation.
- Read-once CFTP makes one further change $\ldots$

ROCFTP–The algorithm

1. Run from $t = 0$ forward to the first C_n event.

2. Follow the path from $X_{(n+1)T} = X_n^*$. Run from $t = (n+1)T$ until C_m, $m > n$, occurs.

3. Output X_{mT}.



ROCFTP–Discussion

- The pairs (C_n, X_n^*) are independent and identically distributed across n. Thus conditioning on the occurence of C_m does not change the distribution of X_{m-1}^*.

- The IID property of the sequence means we are free to re-label the time axis so that $m = 0$. Then X_{m-1}^* is re-labelled as X_0 and is the output of CFTP.

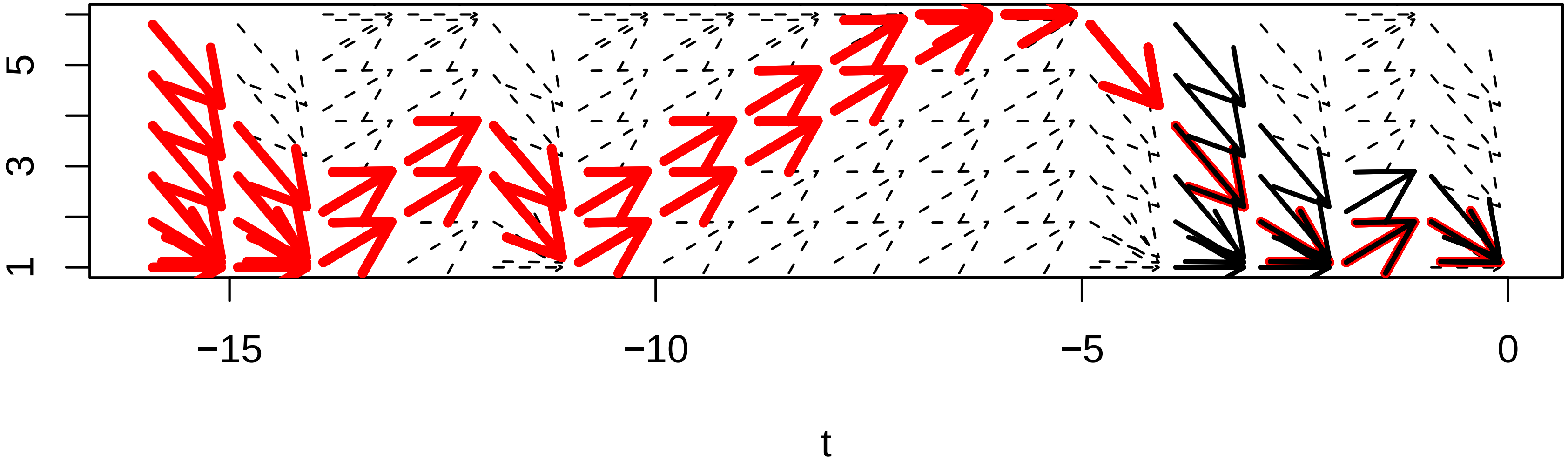
- We can output multiple *independent* values by using m as n and repeating steps 2 and 3.

- This is the easiest perfect sampler to program. We don't need to re-use U_t values, we only need to save the $X_{(n+1)T}$ values for possible output.

2. Impatience bias

CFTP may terminate, but there is no upper bound on how long it will take. If we give up when $M > M^*$, we introduce an *impatience bias*: we are more likely to see outcomes arising from fast coalescence.

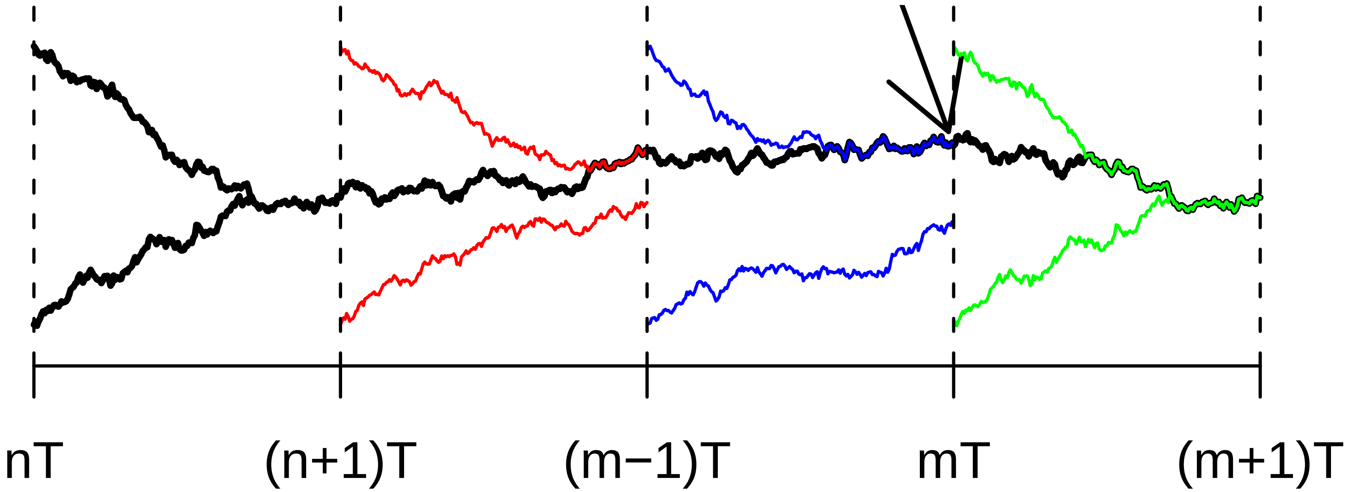
E.g. asymmetric random walk on $\{1, \ldots, N\}$ that steps up 1, down 2 can coalesce at 1 in $N/2$ steps, but needs at least $N - 1$ steps to coalesce at N.



Fill's algorithm (Fill, 1998; Fill et al., 2000)

Fill's algorithm is a rejection sampler that does *not* have impatience bias, i.e. it is interruptible. It is related to ROCFTP.

Recall: *Given coalescence, the value at the end of the block is independent of the value at the start.*



The reverse process

Fill's algorithm makes use of the reversal of a Markov chain. The reversal Y_t of X_t has the same transition probabilities, in reverse:

$$P(Y_{t+1} \in A | Y_t \in B) = P(X_t \in A | X_{t+1} \in B)$$

where we assume that marginally X_t is in steady-state.

For reversible chains (common in MCMC), Y_t has the same transition probabilities as X_t. If not, we use Bayes Rule to find the transition probabilities:

$$\begin{aligned} P(X_t \in A | X_{t+1} \in B) &= \frac{P(X_{t+1} \in B | X_t \in A) P(X_t \in A)}{P(X_{t+1} \in B)} \\ &= P(X_{t+1} \in B | X_t \in A) \pi(A) / \pi(B) \end{aligned}$$

Fill's Algorithm

1. Arbitrarily choose X_T.

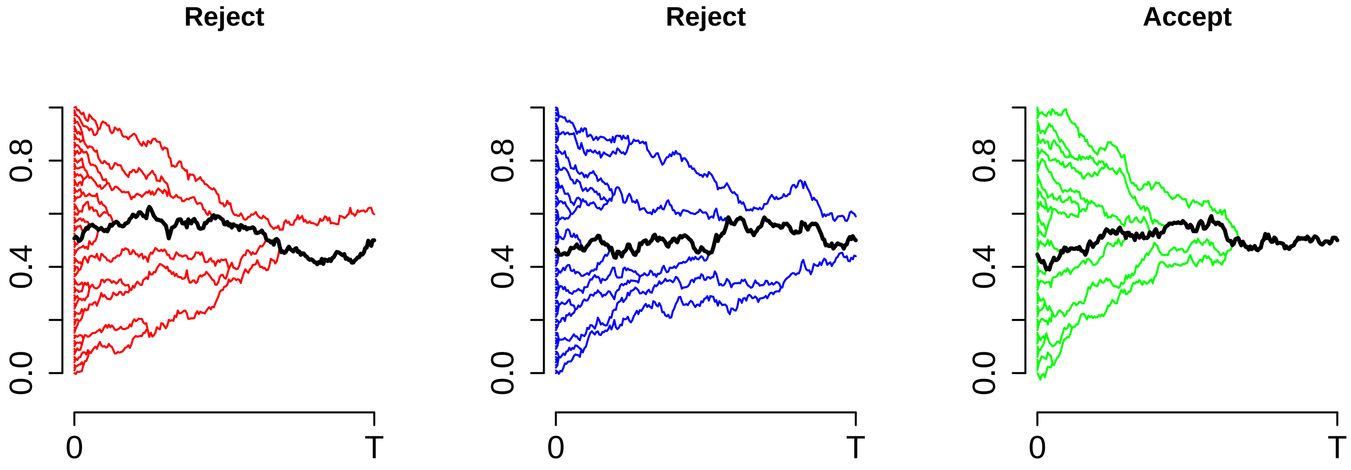
2. Using the reverse process, simulate $X_{T-1}, \ldots, X_0$.

3. Choose $U_1, \ldots, U_T$ conditional on $X_{t+1} = \phi(X_t, U_{t+1})$.

4. Use $U_1, \ldots, U_T$ to simulate all paths from 0 to T.

5. If all paths coalesce, output X_0.

6. Otherwise, reject and repeat independently.



Is Fill's $X_0 \sim \pi(\cdot)$?

Yes. We are constructing the value at the beginning of a coalescent block, just as ROCFTP does. The fact that X_T is chosen arbitrarily doesn't matter as long as X_T is in the support of π, since X_0 is independent of it.

Is Fill interruptible?

- Yes (well, maybe). We always take T steps to generate a block, and each block is independent of all others. There is no dependence between the number of steps and the output value.
- There may be dependence on the length of time to calculate each step; if so, Fill is not interruptible.

Practical difficulties with Fill

- Usually it is possible to work out the transition probabilities of the reversal of X_t, but it may be hard.
- It is usually quite difficult to draw the U_t values. U_t are i.i.d. from a distribution that is designed to be straightforward to sample, but we require draws conditional on $X_t = \phi(X_{t-1}, U_t)$. Since $\phi(\cdot, \cdot)$ may be quite complicated in practical situations, this is not easy.

3. The ideas behind CFTP

In CFTP:

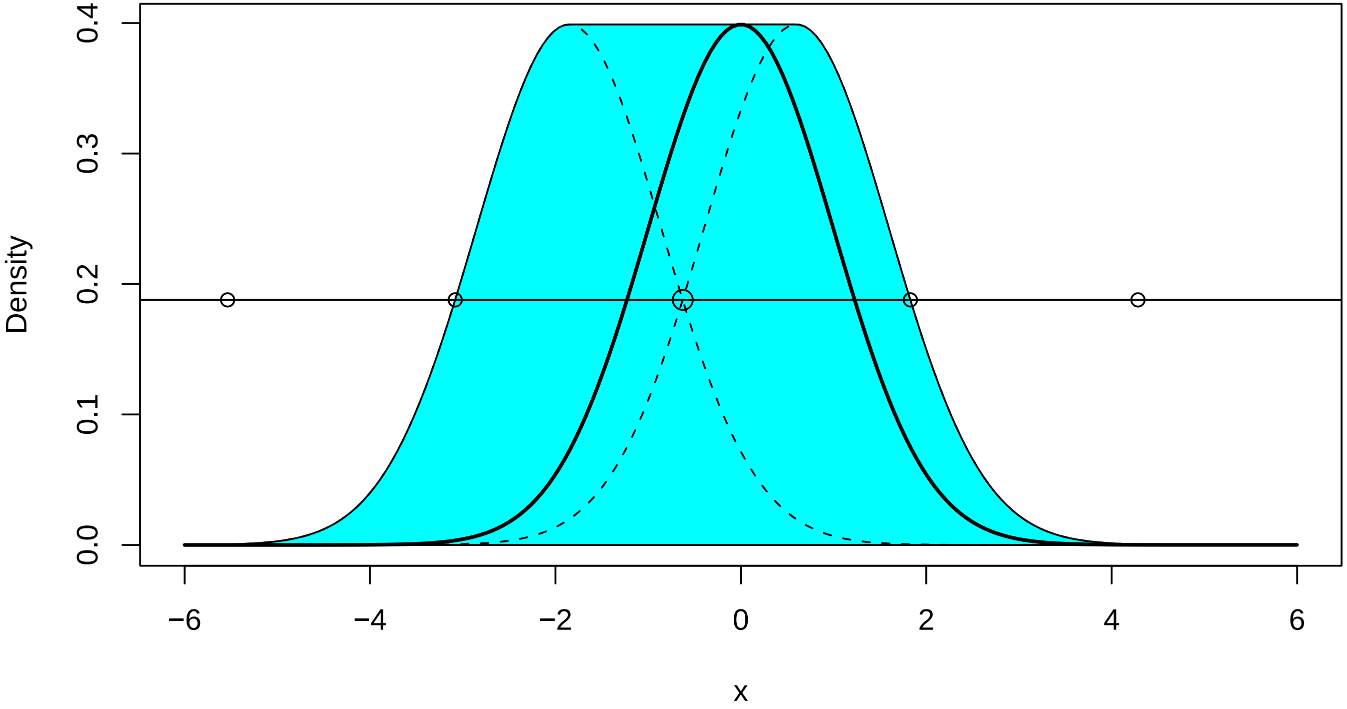
- By going further and further into the past, we make the T-step distributions from each starting point more similar.
- CFTP couples these approximations and goes into the past far enough to get an identical sample from all of them.

What can we abstract from this?

- It is not necessary to know a distribution exactly in order to sample from it: nearby distributions can give identical samples.
- The more similar the distributions, the easier this is.

Example: Sampling from $N(\mu, 1)$

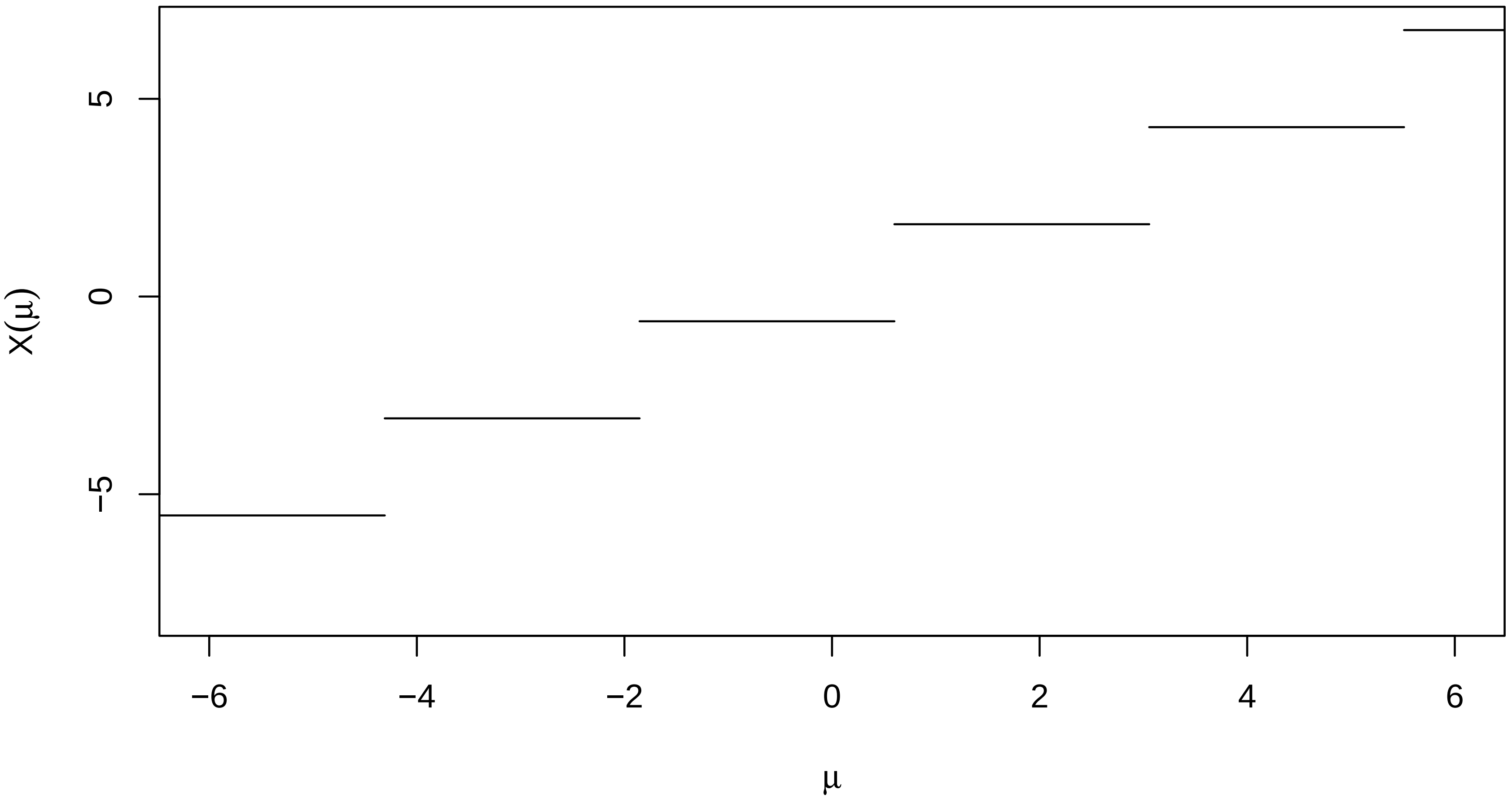
If we know that μ is either μ_1 or μ_2 but we don't know which one, a shift coupler can sample $X_1 \sim N(\mu_1, 1)$ and $X_2 \sim N(\mu_2, 1)$ with a high probability that $X_1 = X_2$.



If $X_1 = X_2$, that value will work for *any* $\mu \in [\mu_1, \mu_2]$.

Random functions

For each value of U, a shift coupler gives a step function $X(\mu)$.



Example: Alternating sums

Suppose $a_n \searrow 0$, and we are interested in evaluating the alternating sum

$$S_n = \sum_{i=0}^{n} (-1)^i a_i$$

- The even partial sums $S_0, S_2, \ldots$ give a decreasing sequence of upper bounds; the odd partial sums $S_1, S_3, \ldots$ give an increasing sequence of lower bounds.
- The intervals $[S_{2n-1}, S_{2n}]$ give a sequence of better and better approximations to the limit $S = S_\infty$.

We can generate samples with expected value exactly S without being able to evaluate S exactly.

Sampling without knowing the target distribution

We want $X_i \sim N(S, \sigma^2)$; we know σ^2, but we don't know S. How?

- Couple all $N(\mu, \sigma^2)$ distributions using Wilson's shift coupler.
- Sample one piecewise constant function $X(\cdot)$ s.t. $X(\mu) \sim N(\mu, \sigma^2)$.
- Evaluate enough partial sums so that the interval $[S_{2n-1}, S_{2n}]$ falls entirely within a constant segment of $X(\cdot)$; then $X(S)$ must equal that constant value, so we have our X_i.

How to use X_i

We have $X_i \sim N(S, \sigma^2)$ for any chosen σ^2.

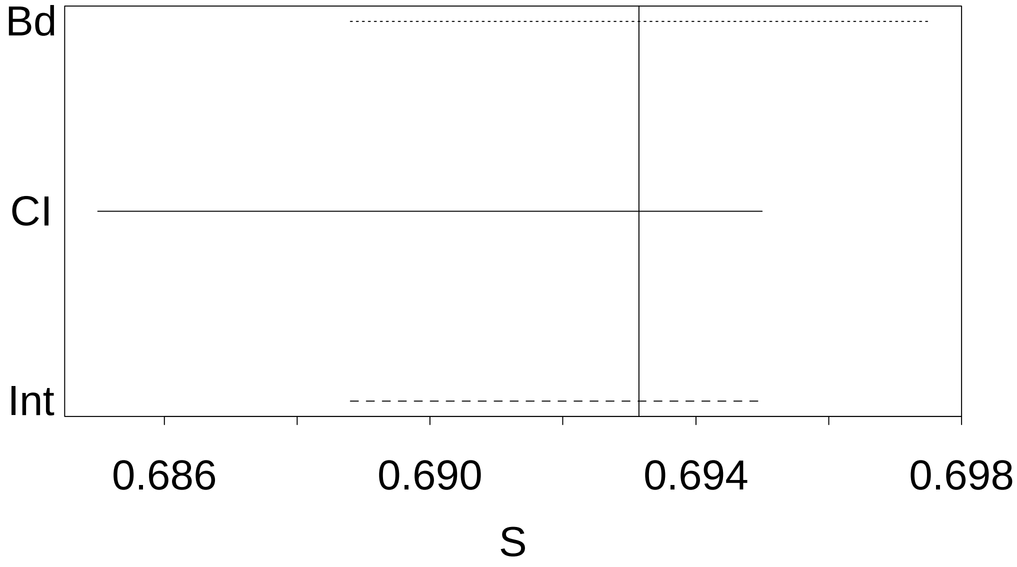
- We can use these samples to calculate 95% confidence intervals for S: $\bar{X} \pm 1.96\sigma/\sqrt{n}$.
- We can choose the length of the desired confidence interval exactly by manipulating σ and/or n.
- We also have bounds: $S \in [S_{2n-1}, S_{2n}]$; we can intersect the bounds with the confidence interval without affecting its confidence level. This can only shorten the confidence interval.

Approximating $\ln 2 = 0.693\ldots$

We use this method to evaluate $1 - 1/2 + 1/3 - \ldots$.

For a 95% confidence interval for the limit of length 0.01, use $n = 10$, $\sigma = 0.0161$.

One simulation required 115 terms from the series:



Bounds = [0.6888, 0.6975], 95% CI = [0.6850, 0.6950], intersection = [0.6888, 0.6950].

Another recent example

Last term Omiros Papaspiliopoulos presented a seminar about simulating stochastic differential equations (Beskos et al., 2004). The algorithm they used is related in some ways to this general sense of perfect sampling: it builds a “skeleton” that would could be drawn from any of a whole range of SDEs, then makes draws conditional on that.

References

Beskos, A., Papaspiliopoulos, O., and Roberts, G. O. (2004), "Retrospective Exact Simulation of Diffusion Sample Paths with Applications," Submitted.

Fill, J. A. (1998), "An Interruptible Algorithm for Perfect Sampling via Markov chains," *The Annals of Applied Probability*, 8, 131–162.

Fill, J. A., Machida, M., Murdoch, D. J., and Rosenthal, J. S. (2000), "Extension of Fill's Perfect Rejection Sampling Algorithm to General Chains," *Random Structures and Algorithms*, 17, 290–316.

Wilson, D. (2000a), "Layered Multishift Coupling for Use in Perfect Sampling Algorithms (with a Primer on

CFTP)," in *Monte Carlo Methods—Fields Institute Communications Vol. 26*, ed. Madras, N., AMS.

Wilson, D. B. (2000b), "How to Couple from the Past Using a Read-Once Source of Randomness," *Random Structures and Algorithms*, 16, 85–113.

Also see David Wilson's web site
`http://dbwilson.com/exact/`.