Two Recursive Simulation Schemes

Duncan Murdoch

Department of Statistical & Actuarial Sciences, University of Western Ontario

June 19, 2009

Outline

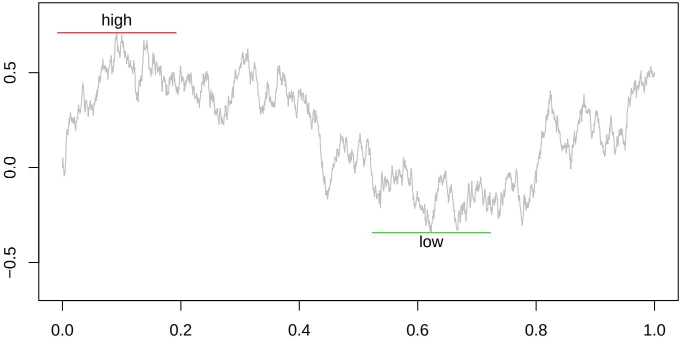
Outline

Introduction

- About 12 years ago I studied perfect simulation, including Propp and Wilson's CFTP algorithm.
- I realized that CFTP is an example of the following general principle: to simulate from a target density $f(\cdot)$, often we can generate a finite sequence of approximations, and be certain that a draw from the final one is drawn exactly from $f(\cdot)$.
- Today I will talk about two applications of this principle. This is joint work with Tingting Gou and John Braun.

Outline

Simulating Extremes of a Diffusion



Given a stochastic differential equation

$$dX_s = \mu(X_s)ds + \sigma(X_s)dW_s$$

our ultimate goal is to simulate functionals such as the high and low points and where they occur, without simulating the entire path.

Just the high for Brownian motion

McLeish (2002) described a simple algorithm to simulate the High or Low values of a Brownian Motion over an interval $[0;T]$, conditional on the values at the end points $W_0 = o; W_T = c$.

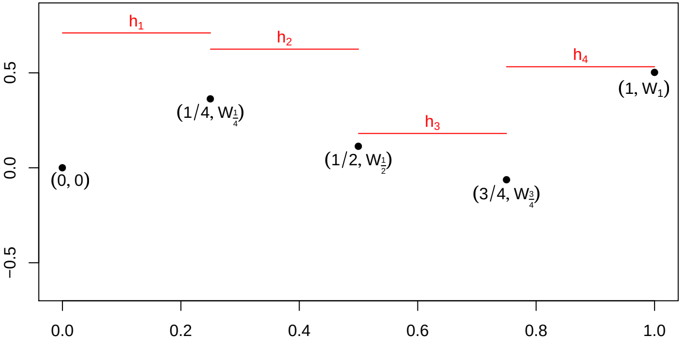
Algorithm

High$(o;c;T)$

$\quad Y \sim \text{Unif}(0; \exp[-(c-o)^2 = 2T)]$

$\quad$ Return $[c + o + \sqrt{-2T\log(Y)}] = 2$

Both the high and its time



The Euler method:

- Divide the interval into N subintervals.
- Discretize and use McLeish on each subinterval, then pick the biggest.

What is wrong with Euler?

- The Euler method gets the distribution of the high exactly right, but only obtains the time to within an interval of length $1/N$.
- This is inaccurate if N is small, slow if N is large.
- We can speed it up by a recursive approach...

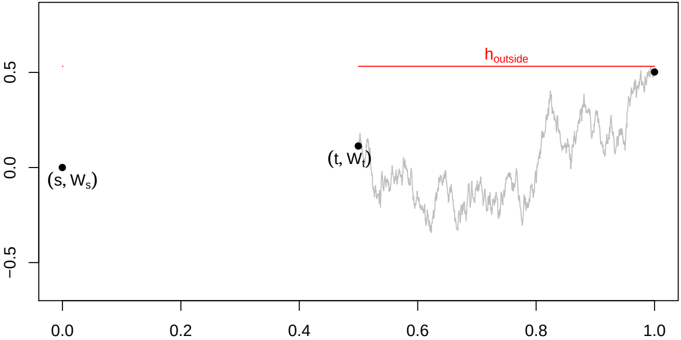
A Recursive Rejection Algorithm

Principle: Divide the interval into two parts: the "inside" $[s;t]$ (containing the max) and the "outside" $[0;1] \setminus [s;t]$. Recursively shrink the inside part.

- Recursion: At each step, we start with $(s;t;W_s;W_t;h_{\text{outside}})$; use 2-step Euler and apply McLeish twice to choose one half of $[s;t]$ as the new inside, and to update h_{outside}.
- Rejection: The high inside must be bigger than h_{outside}. Repeat Euler and McLeish until it is.

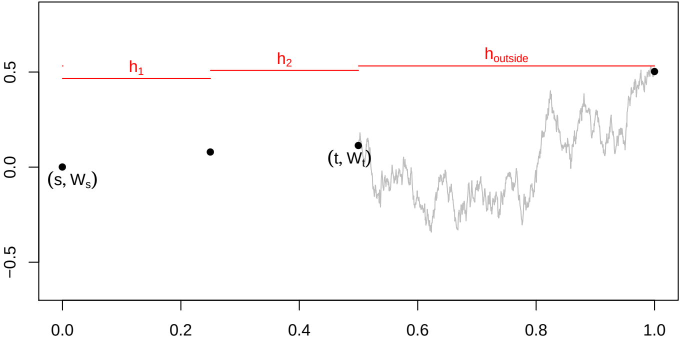
Advantage: Order n steps for 2^n step accuracy: much more efficient than the Euler Method.

RRA at step one



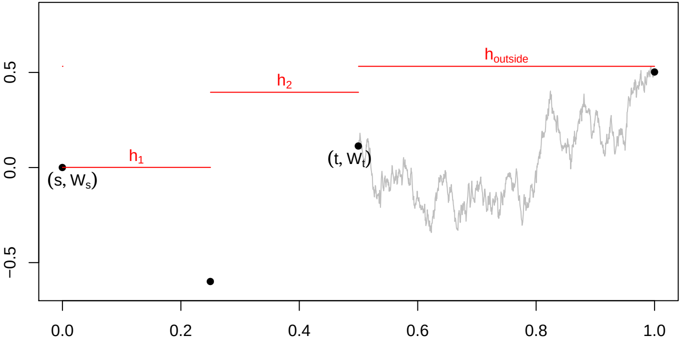
After one step we might have this. (Don't simulate the full path, but consider it fixed...)

RRA first proposal



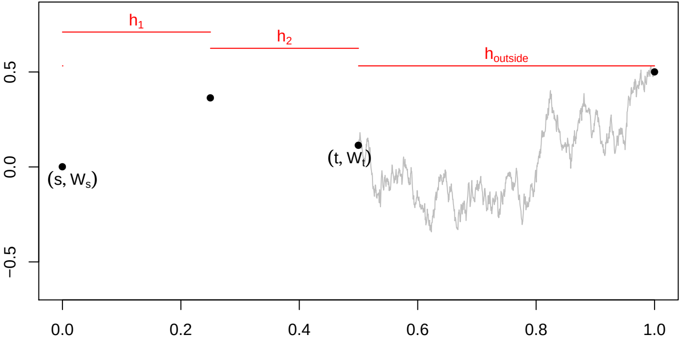
Simulate the inside interval until $\max(h_1; h_2) > h_{\text{outside}}$. This one failed!

RRA second proposal



Try again: failed again!

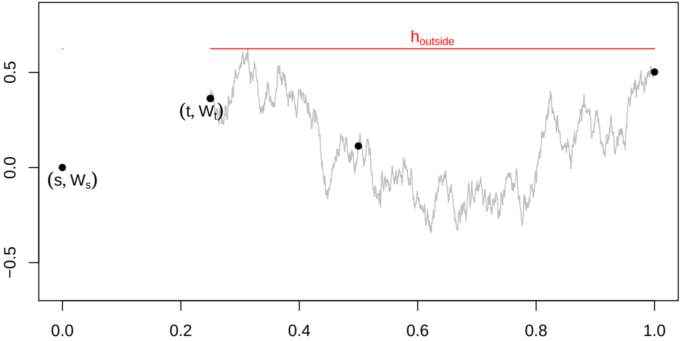
RRA third proposal



Try again: success!

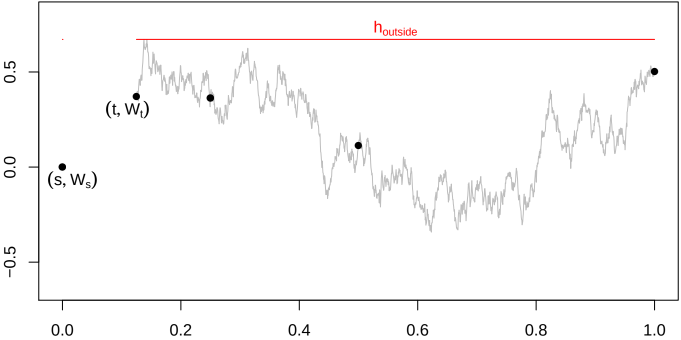
Accept this simulation, set $h_{\text{outside}} = \max(h_{\text{outside}}; h_2)$, discard h_1.

RRA at step two



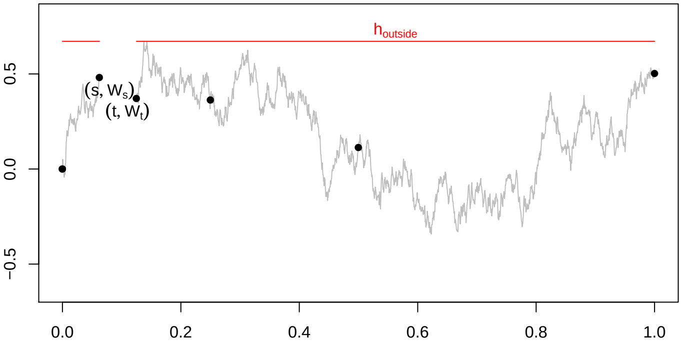
Update to the new state.

RRA at step three

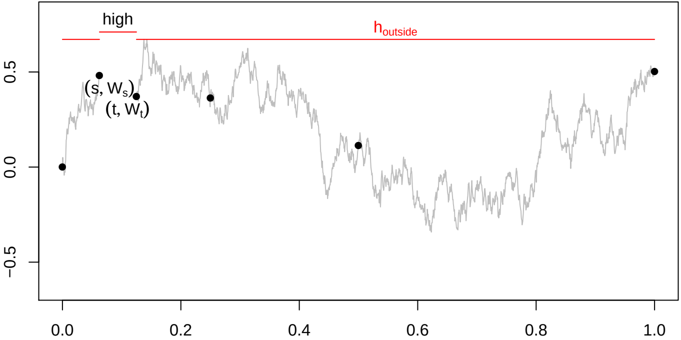


Repeat the whole recursive step to refine the interval. Continue until $|t-s|$ is small enough.

RRA at step four



RRA is done



Apply McLeish one more time at the end (or just use the $\max(h_1; h_2)$ value from the previous step).

Extensions

- Simulating lows instead of highs—use mins not maxes.
- Barrier crossing times and other functionals can be simulated in a similar way.
- Simulating both lows and highs and both locations—more complicated:
  - Invert distribution from Billingsley (1999) to simulate high and low simultaneously.
  - In RRA, the “inside” eventually becomes two disjoint intervals, one containing the high, the other containing the low.
  - We maintain both high and low in the “outside”.
- More general diffusions—Beskos and Roberts (2005), Beskos *et al.* (2006) described an exact algorithm (EA) for simulating some diffusions. First generate a random skeleton; conditional on the skeleton, simulate Brownian bridges between.

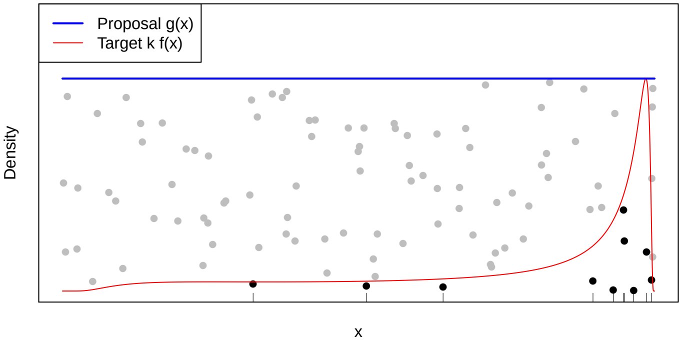
Refinement

Our goal was exact simulation, and RRA only gives us the time(s) to within 2^{-n}. Shepp (1979) derived the joint density of the high h, its time θ, and closing value c for a Brownian motion on $[0,T]$, which allows us to derive

$$f(\theta|h,c,T) \propto \frac{1}{\theta^{3/2}(T-\theta)^{3/2}} \exp\left[-\frac{h^2}{2\theta} - \frac{(h-c)^2}{2(T-\theta)}\right]$$

This is a non-standard density, but we can construct a rejection sampler for it.

Rejection Sampling



Suppose you want to sample from density $f(\cdot)$, and know how to sample from density $g(\cdot)$. Find k such that $g(x) \geq kf(x)$ for all x. Then:

1. Sample Y from $g(\cdot)$.
2. Sample U from $\mathrm{Unif}(0; g(Y))$.
3. If $U < kf(Y)$, output Y; else repeat.

The probability of acceptance is k.

The Rejection Sampler Can Be Slow

It is simple to compute the mode (or modes) of the Shepp density, and then use a Unif(0,1) proposal in a rejection sampler. But this can be very slow (i.e. k can be very small). Some solutions:

1. Identify the values of h, c and T that lead to a slow sampler, and use another RRA step in those cases.
2. Work out a smarter proposal density.
3. Use an adaptive proposal.

Outline

When does rejection sampling work well?

Rejection sampling works very well in low dimensions:

- We can sample even if we do not know the normalizing constants on the densities.
- We get IID samples from the target, unlike MCMC, which gives correlated values from an approximation to the target.
- It is often not hard to find a bounding function in one dimension.
- Gilks and Wild (1992) presented an adaptive rejection sampler: with each rejection, $g(\cdot)$ was adjusted to be a better approximation to $f(\cdot)$. It produced very tight approximations.

Why not use rejection sampling?

In high dimensions, rejection sampling is not so successful:

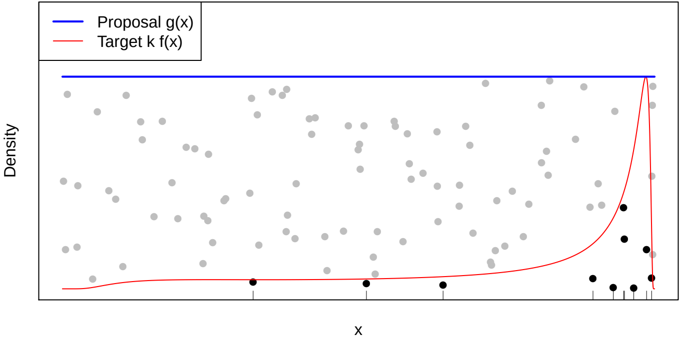
- It is hard to find a proposal that gives tight bounds. (Sometimes this is hard even in one dimension.)
- Typically k will be extremely small, so the sampler will be very inefficient.
- Multidimensional proposal distributions are hard to work with.
- Gilks and Wild (1992) required strong conditions (log-concavity) on $f(\cdot)$; these are not always available and verifiable.

Our strategy

We would like to construct an adaptive sampler, with weak conditions on $f(\cdot)$.

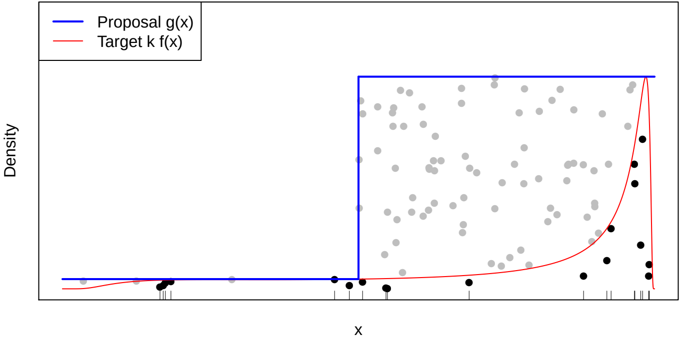
- Start with any bound, one region.
- Split regions where there are a lot of rejections to get tighter bounds.

Example: Shepp's density



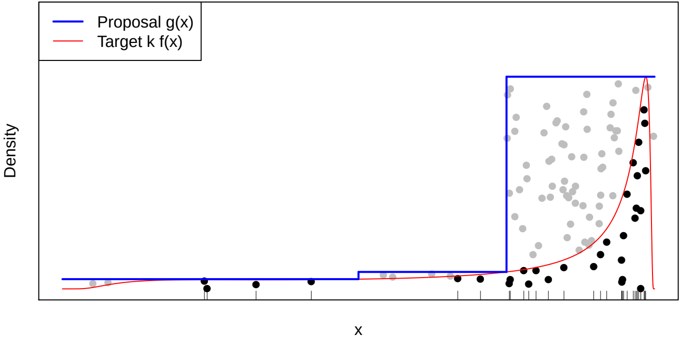
We accepted 10=100 proposals. Can we improve this?

Split the interval and bound separately



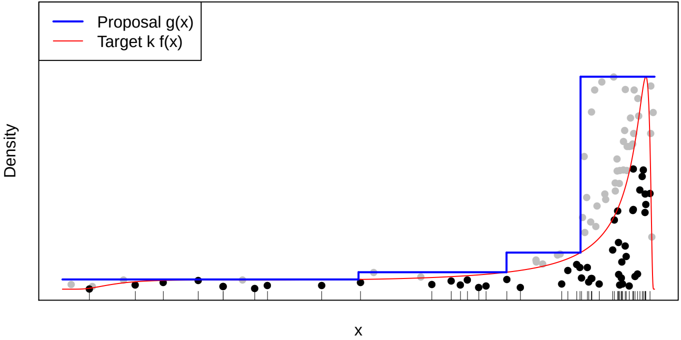
Now we accept 19=100 proposals.

Split again



We chose to split the region with the highest expected number of rejections. Now we accept 31=100 proposals.

And again..



We accept 52=100 proposals with this approximation. We may now draw a large sample using this sampler, which is very fast.

How did we choose where to subdivide?

We can estimate the rejection rate in each region in several ways:

1. Just count how many rejections there were in each region.
2. Better: Find the average of $P(\text{reject})$ in each region, and multiply by the number sampled in that region.
3. Best: Use the computed volume of each region as the multiplier.

Higher Dimensions

We don’t really need the adaptive rejection sampler in one dimension: our first uniform proposal was good enough. But how to handle higher dimensions? Our strategy:

- Divide the space into rectangular regions, and use the same strategy as before to select regions to subdivide.
- Use a proposal that is independent in the coordinates on each subregion.
- Subdivide the target region one coordinate at a time to improve the bound.
- After choosing the region, try all coordinate choices, and pick the best one.

Try to sample from $kf(x;y) = 1{=}(0{:}01 + |x - 0{:}9|^{:4} + |y - 0{:}1|^{:6})$, $0 < x < 1$, $0 < y < 1$, using uniform proposals.

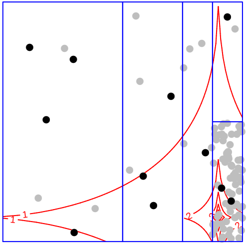
Finding a bound

If $x \in [x_0; x_1]$ and $y \in [y_0; y_1]$, then an upper bound on $kf(x; y)$ is $kf(x^*; y^*)$, where

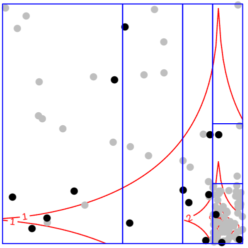
$$x^* = \begin{cases} x_0 & \text{if } x_0 > 0{:}9 \\ x_1 & \text{if } x_1 < 0{:}9 \\ 0{:}9 & \text{otherwise} \end{cases}$$

with a similar formula for y^*.

Sampling from $f(x; y)$

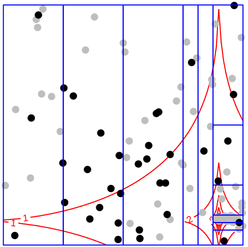


Accepted 11 proposals

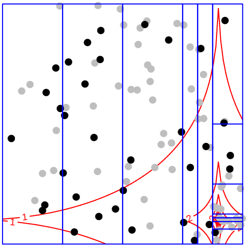


Accepted 16 proposals

Continuing...

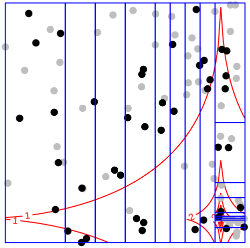


Accepted 35 proposals

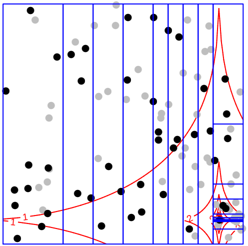


Accepted 35 proposals

Continuing...



Accepted 43 proposals



Accepted 44 proposals

Pump Data Example

Gaver and O'Muircheartaigh (1987) described data on pump failures at a nuclear power plant. A number of authors have analyzed this using the following Bayesian hierarchical model:

- $s_1, \ldots, s_{10}$ count failures after operation for known times $t_1, \ldots, t_{10}$.
- $s_k \sim \text{Poisson}(\lambda_k t_k)$, $k = 1, \ldots, 10$.
- $\lambda_k \sim \text{Gamma}(\alpha, \beta)$, $k = 1, \ldots, 10$, with $\alpha = 1.802$ treated as known.
- $\beta \sim \text{Gamma}(\gamma, \delta)$, with $\gamma = 0.01$ and $\delta = 1$.

We want to study the joint posterior distribution of (β, λ), where $\lambda = (\lambda_1, \ldots, \lambda_{10})$.

The Target Density

The joint posterior is (up to normalizing constants):

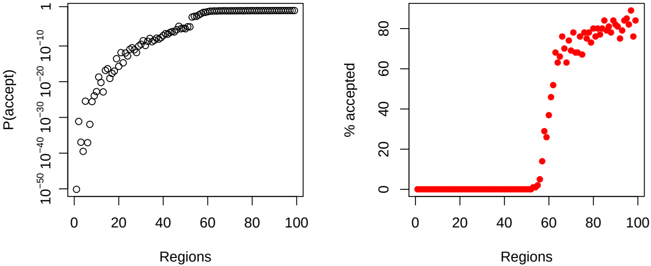
$$f(\beta;\lambda) = \beta^{\gamma+10\alpha-1}e^{-\beta\delta}\prod_{k=1}^{10}\lambda_k^{s_k+\alpha-1}e^{-\lambda_k t_k}e^{-\lambda_k\beta}$$

If $\beta > \beta_0$ and $\lambda_k > \lambda_{k0}$ then

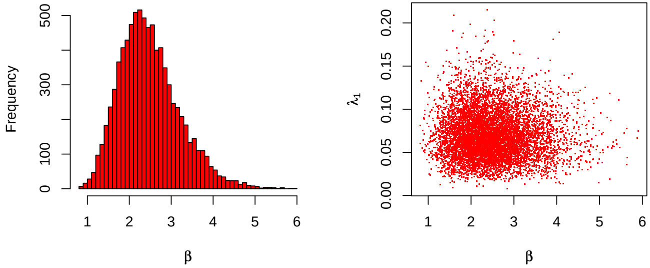
$$\begin{aligned} g(\beta;\lambda) \quad = \quad & e^{\beta_0\sum\lambda_{k0}} \\ & \times\beta^{\gamma+10\alpha-1}e^{-\beta(\delta+\sum\lambda_{k0})} \\ & \times\prod_{k=1}^{10}\lambda_k^{s_k+\alpha-1}e^{-\lambda_k(t_k+\beta_0)} \end{aligned}$$

dominates $f(\beta;\lambda)$, so we may use independent truncated Gamma proposals on rectangular regions.

The Acceptance Rate



The acceptance rate starts out very low (less than 10^{-50}), but quickly rises to acceptable levels.



We obtain IID samples from the posterior, which we can use in whatever further inference we like.

Issues in Multidimensional Case

Implementing the pump data example was both easy and difficult:

- Finding the bounds was very easy, because the target density is mainly made up of easy factors. We expect this to be quite common in Bayesian hierarchical models.
- Evaluating the bounds, and implementing the sampler, was a little trickier than we expected:
    - The problem was in evaluating the truncated Gamma proposals. In many cases, the samples come from far out in the tails, and we were experiencing underflows and huge rounding errors.
    - The solution in this case was to work on a log scale, and to evaluate probabilities using *both* the CDF and the survival function.
- Experience has shown that the pump data is unusually well suited to our algorithm. We can't handle general densities with 11 parameters.

Outline

References

- Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *JRSS B*, 68:1–29.
- Beskos, A. and Roberts, G. O. (2005). Exact simulation of diffusions. *Ann. Appl. Prob.*, 15:2422–2444.
- Billingsley, P. (1999). *Convergence of Probability Measures*. Wiley.
- Gaver, D. and O'Muircheartaigh, I. (1987). Robust empirical Bayes analysis of event rates. *Technometrics*, 29:1–15.
- Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Appl. Stat.*, 41:337–348.
- McLeish, D. L. (2002). Highs and lows: Some properties of the extremes of a diffusion and applications in finance. *CJS*, 30:243–267.
- Shepp, L. A. (1979). The joint density of the maximum and its location for a Wiener process with drift. *JAP*, 16:423-427.