```
idx=find(p_value<0.05);

%now apply importance sampling
N1=10^5;
W=zeros(N1,1); % set up likelihood ratio
for i=1:N1
    Y=randn(1,m+n); Y(idx)=Y(idx).*sig(idx)+mu(idx);
    if score(Y(1:m),Y(m+1:m+n),x,c,A,a)>gamma
        W(i)=prod(sig(idx))*...
          exp(sum((Y(idx)-mu(idx)).^2/2./sig(idx).^2-Y(idx).^2/2));
    end
end
ell=mean(W)
std(W)/sqrt(N1)/ell
Reduction_factor=(ell*(1-ell))/var(W)
```

```
function S=score(Z,Zb,x,c,A,a)
% implements the portfolio loss function
X=A*Z'+a.*Zb';
S=c*(X>x');
```

With the above code we obtain a typical estimate of $\widehat{\ell} = 7.43 \times 10^{-5}$ with an estimated relative error of 0.8%. The variance reduction over CMC is about a factor of 10^3, which is comparable with the performance of the exponential twisting used by Glasserman and Li [38]. Note that while Glasserman and Li derive the optimal exponential change of measure from theoretical arguments, here we have learned the optimal importance sampling by means of MCMC simulation.

10.6 SPLITTING METHOD

One of the first Monte Carlo techniques for rare-event probability estimation is the **splitting method**, proposed by Kahn and Harris [48] and later by Hammersley and Handscomb [41]. In the splitting technique, sample paths of a Markov process are split into multiple copies at various stages of the simulation, with the objective of generating more occurrences of the rare event. The method uses a decomposition of the state space into nested subsets so that the rare event is represented as the intersection of a nested sequence of events. Then, the probability of the rare event is the product of conditional probabilities, each of which can be estimated much more accurately than the rare-event probability itself.

A basic description of the classical splitting method is as follows. Consider a Markov process $\{\mathbf{X}_u,\ u \geqslant 0\}$ with state space $\mathscr{X} \subseteq \mathbb{R}^n$, and let S be a real-valued function on $\mathscr{X}$, referred to as the **importance function**. Assume for definiteness that $S(\mathbf{X}_0) = 0$. For any *threshold* or *level* $\gamma > 0$, let U_γ denote the first time that the process $\{S(\mathbf{X}_u),\ u \geqslant 0\}$ hits the set $[\gamma, \infty)$, and let U_0 denote the first time after 0 that the same process hits the set $(-\infty, 0]$. We assume that U_γ and

U_0 are well-defined finite stopping times with respect to the history of $\{\mathbf{X}_u\}$. One is then interested in the probability, ℓ, of the event $E_\gamma = \{U_\gamma < U_0\}$; that is, the probability that $\{S(\mathbf{X}_u)\}$ up-crosses level γ before it down-crosses level 0. Note that ℓ depends on the distribution of $\mathbf{X}_0$.

The splitting method [31, 34] is based on the observation that if $\gamma_2 > \gamma_1$, then $E_{\gamma_2} \subset E_{\gamma_1}$. Therefore, we have that $\ell = c_1 c_2$, with $c_1 = \mathbb{P}(E_{\gamma_1})$ and ☞ 616 $c_2 = \mathbb{P}(E_{\gamma_2} \,|\, E_{\gamma_1})$ by the product rule of probability. In many cases, estimation of $c_1 c_2$ by estimating c_1 and c_2 separately is more efficient than the direct crude Monte Carlo (CMC) estimation of ℓ. Moreover, the same argument may be used when the interval $[0, \gamma]$ is subdivided into *multiple* subintervals $[\gamma_0, \gamma_1), [\gamma_1, \gamma_2), \ldots, [\gamma_{T-1}, \gamma_T]$, where $0 = \gamma_0 < \gamma_1 < \cdots < \gamma_T = \gamma$. Again, let E_{γ_t} denote the event that the process $\{S(\mathbf{X}_u)\}$ reaches level γ_t before down-crossing level 0. Since $E_{\gamma_0} \supseteq E_{\gamma_1} \supseteq \cdots \supseteq E_{\gamma_T}$ is a nested sequence of events, denoting $c_t = \mathbb{P}(E_{\gamma_t} \,|\, E_{\gamma_{t-1}})$, we have $\ell = \prod_{t=1}^{T} c_t$.

The estimation of each c_t is performed in the following way. At stage $t = 1$ we run $s_1 N_0$ (a fixed number) independent copies of $\{\mathbf{X}_u\}$ and evolve the corresponding process $\{S(\mathbf{X}_u)\}$. Each copy of $\{\mathbf{X}_u\}$ is evolved until $\{S(\mathbf{X}_u)\}$ either hits the set $(-\infty, 0]$ or up-crosses the level γ_1; that is, each copy is evolved for a time period equal to $\min\{U_{\gamma_1}, U_0\}$. The number s_1 is an integer referred to as the **splitting factor** at stage $t = 1$. Define I_j^1 to be the indicator that the j-th copy of $\{S(\mathbf{X}_u)\}$ hits the set $[\gamma_1, \infty)$ before $(-\infty, 0]$, $j = 1, \ldots, s_1 N_0$, and let N_1 be the total number of copies that up-cross γ_1; that is,

$$N_1 = \sum_{j=1}^{s_1 N_0} I_j^1 \,.$$

An unbiased estimate for c_1 is $\widehat{c}_1 = \frac{N_1}{s_1 N_0}$. For every realization of $\{S(\mathbf{X}_u)\}$ which up-crosses γ_1, we store the corresponding state $\mathbf{X}_u$ at the time u of crossing in memory. Such a state is referred to as the **entrance state** [30]. In the next stage, when $t = 2$, we start s_2 new independent copies of the chain $\{\mathbf{X}_u\}$ from each of the N_1 entrance states, giving a total of $s_2 N_1$ new chains. Again, if we let I_j^2 indicate whether the j-th copy of $\{S(\mathbf{X}_u)\}$ hits the set $[\gamma_2, \infty)$ before $(-\infty, 0]$ (with the process $\{\mathbf{X}_u\}$ starting from an entrance state at level γ_1), then $\widehat{c}_2 = \frac{N_2}{s_2 N_1}$, where $N_2 = \sum_{j=1}^{s_2 N_1} I_j^2$, is an estimate of c_2. This process is repeated for each subsequent $t = 3, \ldots, T$, such that $s_t N_{t-1}$ is the **simulation effort** at stage t, and N_t is the number of entrance states at stage t. The indicators $\{I_j^t\}$ at stage t are usually dependent, and hence the success probabilities $\{\mathbb{P}(I_j^t = 1)\}$ depend on the entrance state from which a copy of the chain $\{\mathbf{X}_u\}$ is started. It is well known [7, 34] that despite this dependence, the estimator

$$\widehat{\ell} = \prod_{t=1}^{T} \widehat{c}_t = \frac{N_T}{N_0} \prod_{t=1}^{T} s_t^{-1} \tag{10.26}$$

is unbiased.

The idea of the splitting method is illustrated in Figure 10.4, where three level sets $\{\mathbf{x} : S(\mathbf{x}) = \gamma_t\}$, $t = 0, 1, 2$ are plotted. Here three independent paths of the process $\{S(\mathbf{X}_u)\}$ are started from level $\gamma_0 = 0$. Two of these paths *die out* by down-crossing level 0 and one of the paths up-crosses level γ_1. Three new independent copies of the chain are started from the entrance state at level γ_1 (encircled

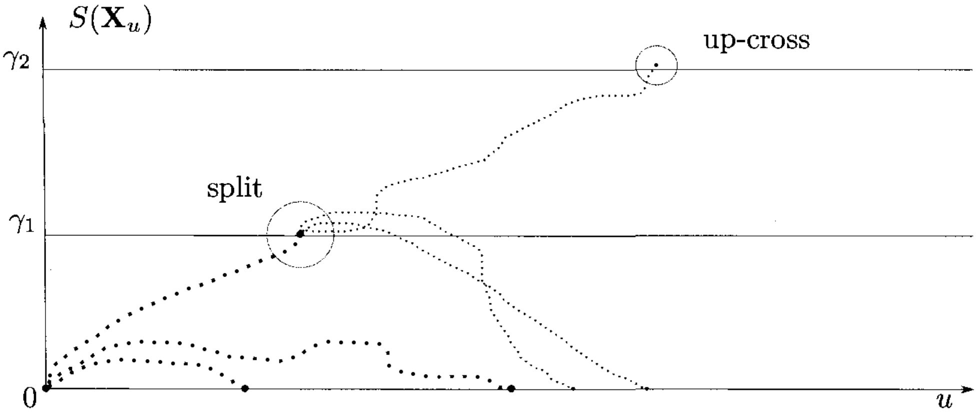

Figure 10.4 Typical evolution of the splitting of $\{S(\mathbf{X}_u)\}$.

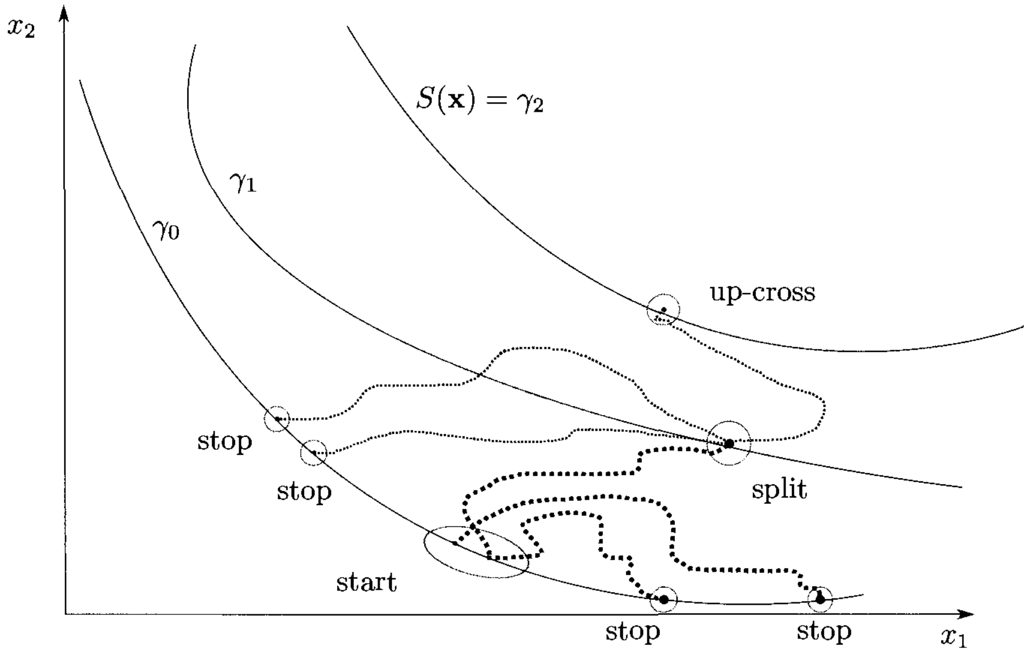

Figure 10.5 Typical evolution of the splitting algorithm for a two-dimensional Markov process $\{(X_u^{(1)}, X_u^{(2)}), u \geqslant 0\}$.

on the graph), two of these copies down-cross 0, but one copy up-crosses level γ_2. Figure 10.5 shows a typical realization of a two-dimensional Markov process $\{(X_u^{(1)}, X_u^{(2)}), u \geqslant 0\}$ that corresponds to the scenario described on Figure 10.4.

Of great importance is the choice of the importance function S, which determines how the sample space is partitioned into nested subsets. The problem of selecting an efficient importance function is similar to the problem of selecting an optimal change of measure for importance sampling. Hence, it is not surprising that it is an unresolved problem in general [32, 58]. Dean and Dupuis [25] suggest an efficient design of the importance function derived from the solution of a nonlinear partial differential equation, for cases where large deviation approximations are available.

For a given importance function S, the efficiency of the splitting method depends crucially on the number of levels T, the choice of the intermediate levels $\gamma_1, \ldots, \gamma_{T-1}$, and the splitting factors $s_1, \ldots, s_T$. Ideally one would select the levels so that the conditional probabilities $\{c_t\}$ are not too small and easily estimated via CMC. Assuming that the cost of running the Markov process is independent of t, the total simulation effort is a random variable with expected value

$$\begin{aligned}\sum_{t=1}^{T} s_t \mathbb{E} N_{t-1} &= N_0 \sum_{t=1}^{T} s_t \, \ell(\gamma_{t-1}) \prod_{j=1}^{t-1} s_j = N_0 \sum_{t=1}^{T} s_t \prod_{j=1}^{t-1} c_j \, s_j \\ &= N_0 \sum_{t=1}^{T} \frac{1}{c_t} \prod_{j=1}^{t} c_j \, s_j \, .\end{aligned} \tag{10.27}$$

An inappropriate choice of the splitting factors may lead to an exponential growth of the simulation effort. For example, if $c_j s_j = a > 1$ for all j, then the simulation effort (10.27) grows exponentially in T. This is referred to in the splitting literature as an **explosion** [35]. On the other hand, if $c_j s_j = a < 1$ for all j, then $\mathbb{E} N_T = N_0 \, a^T$ decays exponentially, and with high probability N_T and $\widehat{\ell}$ will be 0, making the algorithm inefficient. Thus, it is desirable that $c_j s_j = 1$ for all j; that is, the splitting is at the **critical value** [35]. In practice, one obtains rough estimates $\{\varrho_j\}$ of $\{c_j\}$ via a pilot run and then initializes $s_t = \varrho_t^{-1}$ paths from each entrance state $j = 1, \ldots, N_t$, at every stage t. In the case where $1/\varrho_j$ is not an integer, one can generate a Bernoulli random variable with success probability $\varrho_j^{-1} - \lfloor \varrho_j^{-1} \rfloor$ and then add it to $\lfloor \varrho_j^{-1} \rfloor$ to obtain a random integer-valued splitting factor S_j with expected value $1/\varrho_j$, see [35]. This version of the splitting algorithm is called the fixed splitting implementation, because at every stage t one generates a fixed expected number of copies ϱ_t^{-1} from each entrance state. An alternative to the *fixed splitting* implementation is the **fixed effort** implementation, where the simulation effort is fixed to N at each stage, instead of fixing the number of copies. The estimator is then

$$\widehat{\ell}_{\mathrm{FE}} = \prod_{t=1}^{T} \frac{N_t}{N} \, .$$

The fixed effort implementation prevents explosions in the number of total Markov chain copies, but has the disadvantage that it is more difficult to analyze the variance of ℓ_{FE}; see [31].

Other implementations of the splitting method include **fixed success** splitting [33], where the number of entrance states at each stage is maintained to be a predetermined number of trajectories; and **fixed probability of success** splitting [20, 30], where the conditional probabilities $\{c_t\}$ are (approximately) equal under the proposed simulation strategy. We now summarize the fixed effort splitting [31].

Algorithm 10.11 (Fixed Effort Splitting) *Set the counter $t = 1$. Given the importance function S and the levels $\gamma_1, \ldots, \gamma_T$, execute the following steps.*

1. **Initialization.** *Generate N copies of the Markov process $\{\mathbf{X}_u\}$ where each copy is run until $\{S(\mathbf{X}_u)\}$ either hits the set $(-\infty, 0]$ or up-crosses the level γ_1. If I_j^1 is the indicator that the j-th copy of $\{S(\mathbf{X}_u)\}$ hits the set $[\gamma_1, \infty)$ before $(-\infty, 0]$, then $N_1 = \sum_{j=1}^{N} I_j^1$ is the total number of copies that up-cross γ_1. Store the entrance state for each of the N_1 copies in memory.*

2. **Bootstrap Resampling.** *Resample the $N_t (\leqslant N)$ entrance states uniformly to obtain a new population of N entrance states (with possibly repeated values).*

3. **Markov Chain Evolution.** *Start N independent copies of the Markov process $\{\mathbf{X}_u\}$ from each of the N bootstrapped entrance states from Step 2. Run each copy until $\{S(\mathbf{X}_u)\}$ either hits the set $(-\infty, 0]$ or up-crosses the level γ_t. If I_j^t is the indicator that the j-th copy of $\{S(\mathbf{X}_u)\}$ hits the set $[\gamma_t, \infty)$ before $(-\infty, 0]$, then $N_t = \sum_{j=1}^N I_j^t$ is the total number of copies that up-cross γ_t. Store the entrance states for these N_t copies in memory.*

4. **Stopping Condition.** *If $t = T$ go to Step 5. If $N_t = 0$, set $N_{t+1} = N_{t+2} = \cdots = N_T = 0$ and go to Step 5; otherwise, set $t = t + 1$ and repeat from Step 2.*

5. **Final Estimator.** *Deliver the estimator $\widehat{\ell}_{\mathrm{FE}} = \prod_{t=1}^T N_t/N$.*

An analysis of the fixed effort splitting method in an idealized asymptotic setting [31] suggests that good values for the splitting parameters are $T \approx -\ln(\ell)/2$ with thresholds $\{\gamma_t\}$ chosen such that $c_1 = \cdots = c_T \approx \mathrm{e}^{-2}$. Such a choice will (approximately) give a variance of $\mathrm{e}^2 \ell^2 (\ln \ell)^2/4$ per single simulation trial. For more elaborate results in this direction, including central limit results, see [21, 35] and the summary in [58].

■ EXAMPLE 10.14 (Hitting Probability)

Suppose the position of the particle in the plane is described by the process $\{(X_t, Y_t), t \geqslant 0\}$, where $\{X_t\}$ and $\{Y_t\}$ are independent copies of the Ornstein–Uhlenbeck SDE (see [7]): ☞ 199

$$\mathrm{d}Z_t = -Z_t \,\mathrm{d}t + \mathrm{d}W_t, \quad Z_0 = 1 \ .$$

We are interested in computing the probability that the particle hits the quarter circle,

$$\{(x, y) :\ x > 0, y > 0,\ x^2 + y^2 = 25\} \ ,$$

before the x- or y-axis; see Figure 10.6.

To apply the splitting method, we first need to specify an importance function. A natural, but not necessarily optimal, choice is

$$S(x, y) = \begin{cases} \sqrt{x^2 + y^2} & \text{if } x > 0 \text{ and } y > 0, \\ 0, & \text{if } x \leqslant 0 \text{ or } y \leqslant 0. \end{cases}$$

The levels

$$0 = \gamma_0 < \gamma_1 < \gamma_2 < \cdots < \gamma_T = 5$$

can then be interpreted as the increasing radii of a nested set of quarter circles. In other words, we aim to hit a circle with smaller radius before attempting to hit a circle with larger radius. These intermediate circles serve as stepping stones toward the rare-event set.

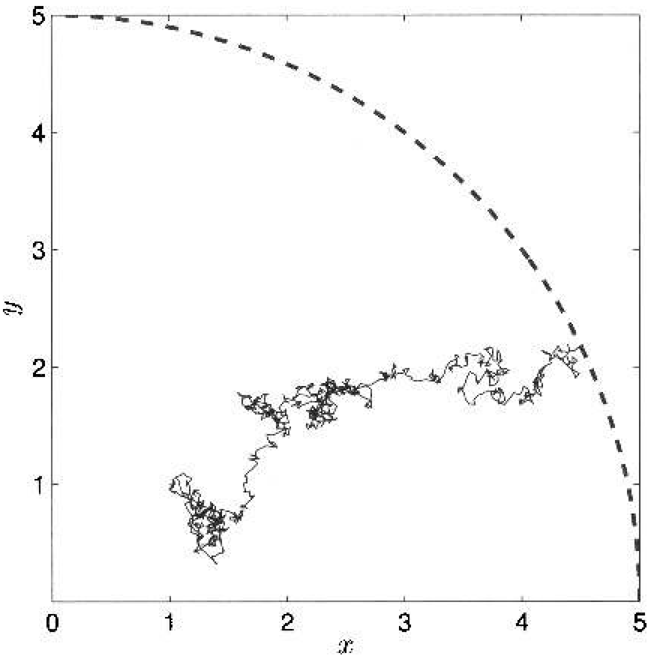

Figure 10.6 A realization of the Ornstein–Uhlenbeck process hitting the quarter circle before the x- or y-axis.

The code below uses fixed effort splitting with $N = 10^4$ and estimates the relative error from 10 independent runs. We use the levels $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6) = (3,\ 3.5,\ 4,\ 4.5,\ 4.7,\ 5)$. The function ou_split.m implements Algorithm 5.25 with step size of $h = 0.01$ and determines if the Ornstein–Uhlenbeck process has hit a quarter circle with radius γ_t. We obtained the estimate 5.6026×10^{-10} with an estimated relative error of 0.049. Figure 10.6 shows a path of the process conditional on hitting the quarter circle before the axis.

```
%OU_process_splitting.m
clear all,clc
gam=[3:0.5:4.5,4.7,5]; %splitting levels
N=10^4;
for iter=1:10 % repeat 10 independent times to estimate RE
    x_ini=1;y_ini=1;
    data=repmat([x_ini,y_ini],N,1);
    for t=1:length(gam)
        %resample the paths
        data=data(ceil(rand(N,1)*size(data,1)),:);
        elite=[];
        for i=1:N
            [indicator,x,y]=ou_split(gam(t),data(i,1),data(i,2));
            if indicator
                elite=[elite;x(end),y(end)];%store successful hits
            end
    end
    t
    c(t)=size(elite,1)/N; % conditional probability estimate
    data=elite;
  end
  ell(iter)=prod(c);
end
mean(ell)
std(ell)/sqrt(10)/mean(ell)
```

```
function [indicator,x,y,tau]=ou_split(gam,x_ini,y_ini)
% implements the exact sampling of an OU process
h=0.001; % step size
% OU updating formula for exact simulation
f=@(x,z)(exp(-h)*x+sqrt((1-exp(-2*h))/2)*z);
x(1)=x_ini; y(1)=y_ini;
tau_axis=inf; tau_circ=inf;
for i=2:10^7 % choose an arbitrarily large loop to ensure hitting
  x(i)=f(x(i-1),randn);
  y(i)=f(y(i-1),randn);
  if (x(i-1)*x(i)<0)|(y(i-1)*y(i)<0)
    tau_axis=h*i; % determine axis hitting time
  end
  if (x(i-1)^2+y(i-1)^2<gam^2)&(x(i)^2+y(i)^2>gam^2)
    tau_circ=h*i; % determine circle hitting time
  end
  tau=min(tau_circ,tau_axis);
  if isinf(tau)==0
    break
  end
end
indicator=tau_circ<tau_axis; %did we hit the circle first?
```

Further Reading

A good starting point on rare-event simulation is [7, Chapter VI], and a recent volume on many of the topics and applications of rare-event simulation is [58]. See [59] for the cross-entropy method with applications to rare-event simulation. Glasserman and Kou [37] give the first example where exponential twisting does not yield asymptotically optimal estimators. Dupuis, Sezer, and Wang [26] provide the first asymptotically optimal importance sampling for total population overflow in tandem networks; this is further generalized to general Jackson networks in Dupuis and Wang [27]. Anantharam et al. [2] show the connection between the overflow probabilities of a queueing process and its time-reversed process; see also Juneja and Nicola [46].

For an overview of splitting for rare event simulation, see [30, 36]. Various applications of the splitting method include: particle transmission [48], queueing systems [31, 32, 64, 65, 66], and reliability [19, 58]. For theoretical results about the optimal selection of the splitting levels, see [20]. A variant of the splitting method that uses quasi Monte Carlo estimators is given in [51]. Various strategies for the truncation and splitting of the Markov chain are described in [50, 52].

REFERENCES

1. R. J. Adler, R. E. Feldman, and M. S. Taqqu. *A Practical Guide to Heavy Tails: Statistical Techniques and Applications.* Birkhäuser, New York, 1998.
2. V. Anantharam, P. Heidelberger, and P. Tsoucas. Analysis of rare events in continuous time Markov chains via time reversal and fluid approximation. Technical Report RC 16280, IBM, Yorktown Heights, New York, 1990.
3. S. Asmussen. *Applied Probability and Queues.* Springer-Verlag, New York, second edition, 2003.
4. S. Asmussen and H. Albrecher. *Ruin Probabilities.* World Scientific Publishing, River Edge, NJ, 2010.
5. S. Asmussen and K. Binswanger. Simulation of ruin probabilities for subexponential claims. *ASTIN Bulletin*, 27(2):297–318, 1997.
6. S. Asmussen, K. Binswanger, and B. Højgaard. Rare events simulation for heavy-tailed distributions. *Bernoulli*, 6(2):303–322, 2000.
7. S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis.* Springer-Verlag, New York, 2007.
8. S. Asmussen and D. P. Kroese. Improved algorithm for rare event simulation with heavy tails. *Advances in Applied Probability*, 38(2):545–558, 2006.
9. S. Asmussen and L. Rojas-Nandayapa. Asymptotics of sums of lognormal random variables with Gaussian copula. *Statistics & Probability Letters*, 78(16):2709–2714, 2008.
10. S. Asmussen and R. Y. Rubinstein. Steady state rare events simulation in queuing models and its complexity properties. In J. H. Dshalalow, editor, *Advances in Queuing: Theory, Methods, and Open Problems*, pages 429–462. CRC Press, Boca Raton, FL, 1995.
11. A. Bassamboo, S. Juneja, and A. Zeevi. On the efficiency loss of state-independent importance sampling in the presence of heavy-tails. *Operations Research Letters*, 35(2):251–260, 2007.
12. A. Bassamboo, S. Juneja, and A. Zeevi. Portfolio credit risk with extremal dependence: Asymptotic analysis and efficient simulation. *Operations Research*, 56(3):593–606, 2008.
13. J. Blanchet and P. W. Glynn. Strongly efficient estimators for light-tailed sums. *Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools (ValueTools), Pisa*, 180 of ACM International Conference Proceedings Series:article 18, 2006.
14. J. Blanchet and P. W. Glynn. Efficient rare-event simulation for the maximum of heavy-tailed random walks. *The Annals of Applied Probability*, 18(4):1351–1378, 2008.