

Markov chain Monte Carlo for rare-event simulation in heavy-tailed settings

Thorbjörn Gudmundsson

Abstract

In this thesis a method based on a Markov chain Monte Carlo (MCMC) algorithm is proposed to compute the probability of a rare event. The conditional distribution of the underlying process given that the rare event occurs has the probability of the rare event as its normalising constant. Using the MCMC methodology a Markov chain is simulated, with that conditional distribution as its invariant distribution, and information about the normalising constant is extracted from its trajectory.

The algorithm is described in full generality and applied to four different problems of computing rare-event probability. The first problem considers a random walk $Y_1 + \dots + Y_n$ exceeding a high threshold, where the increments Y are independent and identically distributed and heavy-tailed. The second problem is an extension of the first one to a heavy-tailed random sum $Y_1 + \dots + Y_N$ exceeding a high threshold, where the number of increments N is random and independent of Y_1, \dots, Y_n . The third problem considers a stochastic recurrence equation $X_n = A_n X_{n-1} + B_n$ exceeding a high threshold, where the innovations B are independent and identically distributed and heavy-tailed. The final problem considers the ruin probability for an insurance company with risky investments.

An unbiased estimator of the reciprocal probability for each corresponding problem is constructed whose normalised variance vanishes asymptotically. The algorithm is illustrated numerically and compared to existing importance sampling algorithms.

Sammanfattning

I denna avhandling presenteras en metod baserad på MCMC (Markov chain Monte Carlo) för att beräkna sannolikheten av en sällsynt händelse. Den betingade fördelningen för den underliggande processen givet att den sällsynta händelsen inträffar har den sökta sannolikheten som sin normaliseringskonstant. Med hjälp av MCMC-metodiken skapas en Markovkedja med betingade fördelningen som sin invarianta fördelning och en skattning av normaliseringskonstanten baseras på den simulerade kedjan.

Algoritmen beskrivs i full generalitet och tillämpas på fyra exempelproblem. Första problemet handlar om en slumpvandring $Y_1 + \dots + Y_n$ som överskrider en hög tröskel, då stegen Y är oberoende, likafördelade med tungsvansad fördelning. Andra problemet är en utvidgning av det första till summa av ett stokastiskt antal termer. Tredje problemet behandlar sannolikheten att lösningen X_n till en stokastisk rekurrenslikning $X_n = A_n X_{n-1} + B_n$ överskrider en hög tröskel då innovationerna B är oberoende, likafördelade med tungsvansad fördelning. Sista problemet handlar om ruinsannolikhet för ett försäkringsbolag med riskfyllda investeringar.

För varje exempelproblem konstrueras en väntevärdesriktig skattning av den reciproka sannolikheten. Skattningarna är effektiva i meningen att deras normaliserade varians går mot noll. Vidare är de konstruerade Markovkedjorna likformigt ergodiska. Algoritmerna illustreras numeriskt och jämförs med existerande importance sampling algoritmer.

Acknowledgements

I want to express my deepest appreciation for the support and help that I have received from my supervisor Henrik Hult. I am truly grateful for be given the opportunity to work under his guidance.

I want to offer my special thanks to colleagues at the faculty for their advice and help, in particular Filip Lindskog and Tobias Rydén. Also want thank my fellow Ph.D. students, Björn, Johan and Pierre, for countless discussions and practice sessions on the blackboard.

Finally, I want to thank my two special ones Rannveig and Gyða for their immense support and love.

Contents

1	Introduction	1
1.1	Stochastic simulation	2
1.1.1	Sampling a random variable	2
1.1.2	Markov chain Monte Carlo	4
1.1.3	Rare-event simulation	5
1.1.4	Importance sampling	6
1.1.5	Heavy-tailed distributions	6
1.2	Markov chain Monte Carlo in rare-event simulation	7
1.2.1	Formulation	7
1.2.2	Controlling the normalised variance	8
1.2.3	Ergodic properties	10
1.2.4	Efficiency of the MCMC algorithm	11
1.3	Outline and contribution of this thesis	11
2	General Markov chain Monte Carlo formulation	13
2.1	Asymptotic efficiency criteria	14
3	Heavy-tailed Random Walk	15
3.1	A Gibbs sampler for computing $\mathbf{P}(S_n > a_n)$	15
3.2	Constructing an efficient estimator	18
3.3	Numerical experiments	19
4	Heavy-tailed Random Sum	21
4.1	A Gibbs sampler for computing $\mathbf{P}(S_{N_n} > a_n)$	22
4.2	Constructing an efficient estimator	25
4.3	Numerical experiments	27
5	Stochastic Recurrence Equations	28
5.1	A Gibbs sampler for computing $\mathbf{P}(X_m > c_n)$	28
5.2	Constructing an efficient estimator	31
5.3	Numerical experiments	33
6	Ruin probability in an Insurance Model with Risky Investments	35
6.1	A Gibbs sampler for computing the ruin probability	36
6.2	Constructing an efficient estimator of the reciprocal ruin probability	36

1 Introduction

Mathematical modelling of systems, in for instance natural sciences has been one of the key building blocks of scientific understanding. The system of interest may be the motion of the planets, the dynamic flow in a liquid, changes in stock prices or the total amount of insurance claims made in a year. Often the model involves the system's dynamic laws, long-time behavior and different possible scenarios. Such models nearly always include a parameter, or a set of parameters, which, though unknown in advance are still needed to calibrate the model to reality. Thus in order to have a fully specified model capable of forecasting the future properties or value, then one needs to measure the values of the the unknown parameters and thereby most likely introducing some measurement error. This error is assumed to be random and thus the resulting forecast is the outcome of a stochastic mathematical model.

With the ever increasing computational capacity in recent decades the models are becoming more and more complex. Minor aspects that were ignored in the simpler models can now be included in the computations, with increasing complexity. Researchers and practitioners alike strive to enhance current models and introduce more and more details to it, in the hope of increasing their forecasting ability. Weather systems and finance processes are examples of models that today are so involved that it is becoming difficult to give analytical and closed form answers to property and forecasting questions. This has given rise to alternative approaches to handling such complex stochastic models, namely stochastic simulation.

Briefly, simulation is the process of sampling the underlying random factors of a model to generate many instances of it, in order to make inferences about its properties. This has proved to be a powerful tool for computation in many academic fields such as physics, chemistry, economics, finance, insurance. Generating instances of even the highly advanced stochastic models, multi-dimensional, non-linear and highly stochastic models can be done in a few milliseconds. Stochastic simulation has thus played its part in the scientific progress of recent decades and the simulation themselves has grown into an academic field in its own right.

In physics, hypothesis are often tested and verified via a number of experiments. One experiment is carried out after another, and if sufficiently many of the experiments support the hypothesis then it acquires a certain validity and becomes a theory. This was for instance the case at CERN in the summer of 2012, when the existence of the Higgs boson was confirmed through experiments which supported the old and well known hypothesis. However, one can not always carry out experiments to validate hypotheses. Sometimes it is simply impossible to replicate the model in reality, as is the case when studying the effects of global warming. Obviously, since we can only generate a single physical instance of the Earth, any simulations need to be done via computer modelling. To better reflect reality, the resolution needs to be high and many different physical and meteorological factors need to be taken into account. The surface of the Earth is broken into 10km times 10km squares, each with its temperature, air pressure, moisture and more. The dynamics of these weather factors need to be simulated with small times steps, perhaps many years into the future. The Mathematics and Climate Research Network (MCRN) carries out extensive stochastic simulations, replicating the Earth using different types

of scenarios to forecast possible climate changes. Clearly, this type of stochastic simulation is immensely computationally costly. This scientific work alone justifies the importance of continuing research and improvement in the field of stochastic simulation.

A subfield of stochastic simulation which deals with unlikely events of small probability is called rare-event simulation. Examples of rare-event simulation is when calculating capital requirements of a financing firm subject to Basel III regulations, or of a insurance company subject to Solvency II regulations. Natural catastrophes such as avalanches, volcanic eruptions, to name but few, are also types rare-events for which we are interested in analysing. This is of particular importance when it comes to computationally heavy models. That is because, if an event is rare a computer needs many simulations to get a fair picture of its frequency and the circumstances in which it occurred. And if every simulation takes up a lot of computational time, then a thorough study would require a prohibitive amount of computer time would indeed be required. Therefore the improvement of efficient rare-event stochastic simulation is of high importance.

The effect of heavy-tails in stochastic modelling is an important factor not to be overlooked. By heavy tails we mean essentially that there is a non-negligible probability of extreme outcomes that differ significantly from the average. Such extreme outcomes may have a considerable impact on a stochastic system. For instance, large claims due to a catastrophic event arrive at an insurance company causing serious financial distress for the company. Similarly, large fluctuations on the financial market may lead to insolvency of financial institutions. In data networks the arrival of huge files may cause serious delays in the network, and so on.

This thesis presents a new methodology in rare-event simulation based on the theory of Markov chain Monte Carlo. The general method presented in Section 2 makes very modest probabilistic assumptions and in subsequent sections (random walk in Section 3, random sum in Section 4, stochastic recurrent equations in Section 5, ruin probability in Section 6) is applied to few concrete examples and shown to be efficient.

1.1 Stochastic simulation

In this section we introduce the basic tools in stochastic simulation, such as pseudo random number, the inversion method and Monte Carlo. We present the Markov chain Monte Carlo methodology and discuss briefly ergodicity.

1.1.1 Sampling a random variable

In this section we present the foundations of stochastic simulation, namely the generation of a pseudo random number by a computer and how it can be used to sample a random variable via the inversion method.

Most statistical software programs provide methods for generating a uniformly distributed pseudo random number on the interval, say, $[0, 1]$. These algorithms are deterministic, at its core, and can only imitate the properties and behaviour of a uniformly distributed random variable. The early designs of such algorithms showed flaws in the sense that the pseudo random numbers generated followed a pattern which could easily be identified and predicted.

Nowadays there exists many highly advanced algorithms that generate pseudo random numbers, mimicking a true random number quite well. For the purposes of this thesis we assume the existence of an algorithm producing a uniformly distributed pseudo random number, and ignore any deficiencies and errors arising from the algorithm. In short, we assume that we can sample a perfectly uniformly distributed random variable in some computer program. For a more thorough and detailed discussion we refer to [48].

Now consider a random variable X and denote by F its probability distribution. Say we would like, via some computer software, to sample the random variable X . One approach is the inversion method. The inversion method involves only applying the quantile function to uniformly random variable. More formally the algorithm is as follows.

1. Sample U from the standard uniform distribution.
2. Compute $Z = F^{-1}(U)$,

where $F^{-1} = \min\{x \mid F(x) \geq p\}$. The random variable Z has the same distribution as X as the following display shows.

$$\mathbf{P}(Z \leq x) = \mathbf{P}(F^{-1}\{U\} \leq x) = \mathbf{P}(U \leq F(x)) = F(x).$$

The method can easily be extended to sampling X conditioned on being larger than some constant c . Meaning that we want to sample from the conditional distribution

$$\mathbf{P}(X \in \cdot \mid X > c).$$

The algorithm is formally as follows.

1. Sample U from the standard uniform distribution.
2. Compute $Z = F^{-1}\left((1 - F(c))U + F(c)\right)$.

The distribution of Z is given by,

$$\begin{aligned} \mathbf{P}(Z \leq x) &= \mathbf{P}((1 - F(c))U + F(c) \leq F(x)) = \mathbf{P}\left(U \leq \frac{F(x) - F(c)}{1 - F(c)}\right) \\ &= \frac{F(x) - F(c)}{1 - F(c)} = \frac{\mathbf{P}(c \leq X \leq x)}{\mathbf{P}(X > c)} = \mathbf{P}(X \leq x \mid X > c). \end{aligned}$$

Thus the inversion method provides a simple way of sampling a random variable, conditioned on being larger than c , based solely on the generation of a uniformly distributed random number.

The most standard tool for stochastic simulation is the Monte Carlo technique. The power of Monte Carlo is its simplicity. Let X be a random variable and assume we want to compute the probability that $\{X \in A\}$ for some Borel set A . The idea of Monte Carlo is to sample independent and identically distributed copies of random variable, say X_1, \dots, X_n and simply compute the frequency of hitting the set A . More formally, the Monte Carlo estimator of $\mathbf{P}(X \in A)$ is given by

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n I\{X_i \in A\}.$$

While the procedure is easy and simple there are drawbacks that will be discussed in Section 1.1.3.

1.1.2 Markov chain Monte Carlo

In this section we present a simulation technique called Markov chain Monte Carlo (MCMC) for sampling a random variable X despite only having limited information about its distribution.

MCMC is typically useful when sampling a random variable X having a density f that is only known up to a constant, say

$$f(x) = \frac{\pi(x)}{c},$$

where π is known but $c = \int \pi(x)dx$ is unknown. This may seem strange setup at first but once noted that the normalising constant c may be difficult to determine, say there is no known closed form for c , then this is a natural formulation. An example of this type of setup can be found in Bayesian statistics and hidden Markov chains.

In short, the basic idea of sampling via MCMC is to generate a Markov chain $(Y_t)_{t \geq 0}$ whose invariant density is the same as of X , namely f . There exists plentiful of MCMC algorithms but we shall only name two in this thesis, the Metropolis-Hastings algorithm and the Gibbs algorithm.

The method first laid out by Metropolis [41] and then extended by Hastings [26] is based on a proposal density, which we shall denote by g . Firstly the Markov chain $(Y_t)_{t \geq 0}$ is initialised with some $Y_0 = y_0$. The idea behind the Metropolis-Hastings algorithm is to generate a proposal state Z using the proposal density g . The next state of the Markov chain is then assigned the value Z with the acceptance probability α , otherwise the next state of the Markov chain stays unchanged (i.e. retains the same value as before). More formally the algorithm is as follows.

Algorithm 1.1. Set $Y_0 = y_0$. For a given state Y_k , for some $k = 0, 1, \dots$, the next state Y_{k+1} is sampled as follows

1. Sample Z from the proposal density g .
2. Let

$$Y_{k+1} = \begin{cases} Z & \text{with probability } \alpha(Y_k, Z) \\ Y_k & \text{otherwise} \end{cases}$$

$$\text{where } \alpha(y, z) = \min\{1, r(y, z)\}, \quad r(y, z) = \frac{\pi(z)g(z, y)}{\pi(y)g(y, z)}.$$

This algorithm produces a Markov chain $(Y_k)_{k \geq 1}$ whose invariant density is given by f . For more details on the Metropolis-Hastings algorithm we refer to [3] and [23].

Another method of MCMC sampling is the Gibbs sampler, which was originally introduced by Geman and Geman in [22]. If the random variable X is multi-dimensional $X = (X_1, \dots, X_d)$, the Gibbs sampler updates each component at the time by sampling from the conditional marginal distributions. Let $f_{k|\mathcal{Y}}(x_k \mid x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_d)$, $k = 1, \dots, d$, denote the conditional density of X_k given $X_1, \dots, X_{k-1}, X_{k+1}, \dots, X_d$. The Gibbs sampler can be viewed as a special case of the Metropolis-Hastings algorithm where, given $Y_k = (Y_{k,1}, \dots, Y_{k,d})$, one first updates $Y_{k,1}$ from the conditional density $f_{1|\mathcal{Y}}(\cdot \mid Y_{k,2}, \dots, Y_{k,d})$, then $Y_{k,2}$ from the conditional density $f_{2|\mathcal{Y}}(\cdot \mid Y_{k+1,1}, Y_{k,3}, \dots, Y_{k,d})$,

etc. By sampling from these proposal densities the acceptance probability is always equal to 1, so no acceptance step is needed.

An important property of a Markov chain is its ergodicity. Informally, ergodicity measures the how quickly the Markov chain mixes and thus how soon the dependency of the chain dies out. This is a highly desired property since good mixing speeds up the convergence of the Markov chain.

1.1.3 Rare-event simulation

In some specific cases we are interested in computing the probability of a rare event. This may be the probability of ruin of a financial company due to randomness in the future value of assets and liabilities. The multidimensional system of investments and bonds may be so complex that a simulation of the catastrophic event of a ruin may be feasible. For another example, consider a graph of some sort and say we send out a particle on a random walk along the graph given some starting position. Computing the small, and quickly decreasing probability, of that particle returning to its starting position may be of interest as it is an indicator of that graph's dimension. For these reasons and many other, the computation of the probability for a rare-event is relevant.

Consider an unbiased estimator \hat{p} of the probability p and investigate its performance as the probability gets smaller $p \rightarrow 0$. A useful performance measure is the relative error:

$$\text{RE}(\hat{p}) = \frac{\text{Std}(\hat{p})}{p}.$$

An estimator is said to have *vanishing relative error* if $\text{RE}(\hat{p}) \rightarrow 0$ as $p \rightarrow 0$ and *bounded relative error* if $\text{RE}(\hat{p}) < \infty$ as $p \rightarrow 0$.

It is well known that the Monte Carlo estimator is inefficient for computing rare-event probabilities as the following argument shows. Let X be a given random variable with distribution function F and say we would like to compute $p = \mathbf{P}(X \in A)$. We sample number of i.i.d. copies of X , denoted by X_1, \dots, X_n and compute

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n I\{X_i \in A\}.$$

The variance of the estimator is $\text{Var}(\hat{p}) = \frac{1}{n}p(1-p)$, which clearly tends to zero as $n \rightarrow \infty$ but that is not main concern here. What is more interesting is its relative error as the probability p tends to zero. Its relative error is given by

$$\frac{\text{Std}(\hat{p})}{p} = \sqrt{\frac{1}{n} \left(\frac{1}{p} - 1 \right)}.$$

The relative error tends to infinity as $p \rightarrow 0$. Thus making the Monte Carlo algorithm very costly when it comes to rare-event simulation. For example, if a relative error at 1% is desired and the probability is of order 10^{-6} then we need to take n such that $\sqrt{(10^6 - 1)/n} \leq 0.01$. This implies that $n \approx 10^{10}$ which is infeasible on most computer systems.

To improve on standard Monte Carlo a control mechanism needs to be introduced that steer the samples towards the relevant part of the state space, thereby increasing the relevance of each sample. There are several ways to do this, for instance by importance sampling described briefly below, or by splitting

schemes as by L'Ecyer [39], or interacting particle systems as by Del Moral in [14].

1.1.4 Importance sampling

The simulation method of importance sampling comes as a remedy to the problem arising in rare-event simulation. The underlying problem of the Monte Carlo simulation for rare-event studies is the fact that we get *too few samples in the important part of the output space*, meaning that we get too few samples where $\{X \in A\}$. The basic idea of importance sampling is that instead of sampling from the original distribution F the X_1, \dots, X_n are sampled from a so-called sampling distribution, say G . The sampling distribution G is chosen such that we obtain more samples where $\{X \in A\}$. The importance sampling is then simply the average of hitting the event, weighted with the relevant Radon-Nikodym derivative,

$$\hat{p}_{\text{IS}} = \frac{1}{n} \sum_{i=1}^n \frac{dF}{dG} I\{X_i \in A\}.$$

This is a unbiased and consistent estimator since

$$\mathbf{E}_G[\hat{p}_{\text{IS}}] = \int_A \frac{dF}{dG} dG = \mathbf{P}(X \in A).$$

The main difficulty in importance sampling is to design the sampling distribution. Traditionally the functionality and reliability of new stochastic simulation algorithms is “proved” by running extensive numerical experiments. But numerical evidence alone is insufficient. There are numerous examples where the standard heuristics fail and the numerical evidence indicates that the algorithm has converged when, in fact, it is severely biased [24]. The limited evidence provided by simply running numerical experiments has generated the need for a deeper theoretical understanding and analysis of the performance of stochastic simulation algorithms. Over the last decade mathematical tools from stability theory and control theory have been developed with the aim to theoretically quantify the performance of stochastic simulation algorithms for computing probabilities of rare events. In the context of importance sampling two main approaches have been studied; the subsolution approach, based on control theory, by Dupuis, Wang, and collaborators, see e.g. [18, 19, 17], and the approach based on Lyapunov functions and stability theory by Blanchet, Glynn, and others, see [5, 6, 7, 10].

In the theoretical work on efficient importance sampling an algorithm is said to be efficient if relative error per sample, $\text{Std}(\hat{p})/p$ does not grow too rapidly as $p \downarrow 0$.

1.1.5 Heavy-tailed distributions

In this thesis we consider in particular probability distributions F with heavy-tails. The notion of heavy tails refers to the rate of decay of the tail $\bar{F} = 1 - F$ of a distribution function F . A popular class of heavy-tailed distributions is the class of subexponential distributions. A distribution function F supported on the positive axis is said to belong to the subexponential distributions if

$$\lim_{x \rightarrow \infty} \frac{\mathbf{P}(X_1 + X_2 > x)}{\mathbf{P}(X_1 > x)} = 2,$$

for independent random variables X_1 and X_2 with distribution F . A subclass of the subexponential distributions is the regularly varying distributions. \overline{F} is called regularly varying (at ∞) with index $-\alpha \leq 0$ if

$$\lim_{t \rightarrow \infty} \frac{\overline{F}(tx)}{\overline{F}(t)} = x^{-\alpha}, \quad \text{for all } x > 0.$$

The heavy-tailed distributions are often described with the “one big jump” analogy, meaning that the event of a sum of heavy-tailed random variables being large is dominated by the case of one of the variables being very large whilst the rest are relatively small. This is in sharp contrast to the case of light-tails, where the same event is dominated by the case of every variable contributing equally to the total. As a reference to the one big jump analogy we refer the reader to [28, 30, 15].

This one big jump phenomena has been observed in empirical data. For instance, when we consider stock market indices such as Nasdaq, Dow Jones etc. it turns out that the distribution of daily log returns typically has a heavy left tail, see Hult et al. in [29]. Another example is the well studied Danish fire insurance data, which consists of real-life claims caused by industrial fires in Denmark. While the arrivals of claims is showed to be not far from Poisson, the claim size distribution shows clear heavy-tail behavior. The data set is analysed by Mikosch in [43] and the tail of the claim size is shown to be fit well with a Pareto distribution.

Stochastic simulation in the presence of heavy-tailed distributions has been studied with much interest in recent years. The conditional Monte Carlo technique was applied on this setting by Asmussen et al. [2, 4]. Dupuis et al. [16] used importance sampling algorithm in a heavy-tailed setting. Finally we mention the work of Blanchet et al. considering heavy-tailed distributions in [11, 8].

1.2 Markov chain Monte Carlo in rare-event simulation

In this section we describe a new methodology based on Markov chain Monte Carlo (MCMC), for computing probabilities of rare events. A more general version of the algorithm, for computing expectations, is provided in Section 2 along with a precise asymptotic efficiency criteria.

1.2.1 Formulation

Let X be a real-valued random variable with distribution F and density f with respect to the Lebesgue measure. The problem is to compute the probability

$$p = \mathbf{P}(X \in A) = \int_A dF. \quad (1.1)$$

The event $\{X \in A\}$ is thought of as rare in the sense that p is small. Let F_A be the conditional distribution of X given $X \in A$. The density of F_A is given by

$$\frac{dF_A}{dx}(x) = \frac{f(x)I\{x \in A\}}{p}. \quad (1.2)$$

Consider a Markov chain $(X_t)_{t \geq 0}$ with invariant density given by (1.2). Such a Markov chain can be constructed by implementing an MCMC algorithm such as a Gibbs sampler or a Metropolis-Hastings algorithm, see e.g. [3, 23].

To construct an estimator for the normalising constant p , consider a non-negative function v , which is normalised in the sense that $\int_A v(x)dx = 1$. The function v will be chosen later as part of the design of the estimator. For any choice of v the sample mean,

$$\frac{1}{T} \sum_{t=0}^{T-1} \frac{v(X_t)I\{X_t \in A\}}{f(X_t)},$$

can be viewed as an estimate of

$$\mathbf{E}_{F_A} \left[\frac{v(X)I\{X \in A\}}{f(X)} \right] = \int_A \frac{v(x)}{f(x)} \frac{f(x)}{p} dx = \frac{1}{p} \int_A v(x)dx = \frac{1}{p}.$$

Thus,

$$\hat{q}_T = \frac{1}{T} \sum_{t=0}^{T-1} u(X_t), \quad \text{where} \quad u(X_t) = \frac{v(X_t)I\{X_t \in A\}}{f(X_t)}, \quad (1.3)$$

is an unbiased estimator of $q = p^{-1}$. Then $\hat{p}_T = \hat{q}_T^{-1}$ is an estimator of p .

The expected value above is computed under the invariant distribution F_A of the Markov chain. It is implicitly assumed that the sample size T is sufficiently large that the burn-in period, the time until the Markov chain reaches stationarity, is negligible or alternatively that the burn-in period is discarded. Another remark is that it is theoretically possible that all the terms in the sum in (1.3) are zero, leading to the estimate $\hat{q}_T = 0$ and then $\hat{p}_T = \infty$. To avoid such nonsense one can simply take \hat{p}_T as the minimum of \hat{q}_T^{-1} and one.

There are two essential design choices that determine the performance of the algorithm: the choice of the function v and the design of the MCMC sampler. The function v influences the variance of $u(X_t)$ in (1.3) and is therefore of main concern for controlling the rare-event properties of the algorithm. It is desirable to take v such that the normalised variance of the estimator, given by $p^2 \text{Var}(\hat{q}_T)$, is not too large. The design of the MCMC sampler, on the other hand, is crucial to control the dependence of the Markov chain and thereby the convergence rate of the algorithm as a function of the sample size. To speed up simulation it is desirable that the Markov chain mixes fast so that the dependence dies out quickly.

1.2.2 Controlling the normalised variance

This section contains a discussion on how to control the performance of the estimator \hat{q}_T by controlling its normalised variance.

For the estimator \hat{q}_T to be useful it is of course important that its variance is not too large. When the probability p to be estimated is small it is reasonable to ask that $\text{Var}(\hat{q}_T)$ is of size comparable to $q^2 = p^{-2}$, or equivalently, that the standard deviation of the estimator is roughly of the same size as p^{-1} . To this end the normalised variance $p^2 \text{Var}(\hat{q}_T)$ is studied.

Let us consider $\text{Var}(\hat{q}_T)$. With

$$u(x) = \frac{v(x)I\{x \in A\}}{f(x)},$$

it follows that

$$\begin{aligned}
p^2 \text{Var}_{F_A}(\hat{q}_T) &= p^2 \text{Var}_{F_A} \left(\frac{1}{T} \sum_{t=0}^{T-1} u(X_t) \right) \\
&= p^2 \left(\frac{1}{T} \text{Var}_{F_A}(u(X_0)) + \frac{2}{T^2} \sum_{t=0}^{T-1} \sum_{s=t+1}^{T-1} \text{Cov}_{F_A}(u(X_s), u(X_t)) \right), \quad (1.4)
\end{aligned}$$

Let us for the moment focus our attention on the first term. It can be written as

$$\begin{aligned}
\frac{p^2}{T} \text{Var}_{F_A}(u(X_0)) &= \frac{p^2}{T} \left(\mathbf{E}_{F_A}[u(X_0)^2] - \mathbf{E}_{F_A}[u(X_0)]^2 \right) \\
&= \frac{p^2}{T} \left(\int \left(\frac{v(x)}{f(x)} I\{x \in A\} \right)^2 F_A(dx) - \frac{1}{p^2} \right) \\
&= \frac{p^2}{T} \left(\int \frac{v^2(x)}{f^2(x)} I\{x \in A\} \frac{f(x)}{p} dx - \frac{1}{p^2} \right) \\
&= \frac{1}{T} \left(\int_A \frac{v^2(x)p}{f(x)} dx - 1 \right).
\end{aligned}$$

Therefore, in order to control the normalised variance the function v must be chosen so that $\int_A \frac{v^2(x)}{f(x)} dx$ is close to p^{-1} . An important observation is that the conditional density (1.2) plays a key role in finding a good choice of v . Letting v be the conditional density in (1.2) leads to

$$\int_A \frac{v^2(x)}{f(x)} dx = \int_A \frac{f^2(x) I\{x \in A\}}{p^2 f(x)} dx = \frac{1}{p^2} \int_A f(x) dx = \frac{1}{p},$$

which implies,

$$\frac{p^2}{T} \text{Var}_{F_A}(u(X)) = 0.$$

This motivates taking v as an approximation of the conditional density (1.2). This is similar to the ideology behind choosing an efficient importance sampling estimator.

If for some set $B \subset A$ the probability $\mathbf{P}(X \in B)$ can be computed explicitly, then a candidate for v is

$$v(x) = \frac{f(x) I\{x \in B\}}{\mathbf{P}(X \in B)},$$

the conditional density of X given $X \in B$. This candidate is likely to perform well if $\mathbf{P}(X \in B)$ is a good approximation of p . Indeed, in this case

$$\int_A \frac{v^2(x)}{f(x)} dx = \int_A \frac{f^2(x) I\{x \in B\}}{\mathbf{P}(X \in B)^2 f(x)} dx = \frac{1}{\mathbf{P}(X \in B)^2} \int_B f(x) dx = \frac{1}{\mathbf{P}(X \in B)},$$

which will be close to p^{-1} .

Now, let us shift emphasis to the covariance term in (1.4). As the samples $(X_t)_{t=0}^{T-1}$ form a Markov chain the X_t 's are dependent. Therefore the covariance term in (1.4) is non-zero and may not be ignored. The crude upper bound

$$\text{Cov}_{F_A}(u(X_s), u(X_t)) \leq \text{Var}_{F_A}(u(X_0)),$$

leads to the upper bound

$$\frac{2p^2}{T^2} \sum_{t=0}^{T-1} \sum_{s=t+1}^{T-1} \text{Cov}_{F_A}(u(X_s), u(X_t)) \leq p^2 \left(1 - \frac{1}{T}\right) \text{Var}_{F_A}(u(X_0))$$

for the covariance term. This is a very crude upper bound as it does not decay to zero as $T \rightarrow \infty$. But, at the moment, the emphasis is on small p so we will proceed with this upper bound anyway. As indicated above the choice of v controls the term $p^2 \text{Var}_{F_A}(u(X_0))$. We conclude that the normalised variance (1.4) of the estimator \hat{q}_T is controlled by the choice of v when p is small.

1.2.3 Ergodic properties

As we have just seen the choice of the function v controls the normalised variance of the estimator for small p . The design of the MCMC sampler, on the other hand, determines the strength of the dependence in the Markov chain. Strong dependence implies slow convergence which results in a high computational cost. The convergence rate of MCMC samplers can be analysed within the theory of φ -irreducible Markov chains. Fundamental results for φ -irreducible Markov chains are given in [42, 44]. We will focus on conditions that imply a geometric convergence rate. The conditions given below are well studied in the context of MCMC samplers. Conditions for geometric ergodicity in the context of Gibbs samplers have been studied by e.g. [12, 51, 52], and for Metropolis-Hastings algorithms by [40].

A Markov chain $(X_t)_{t \geq 0}$ with transition kernel $p(x, \cdot) = \mathbf{P}(X_{t+1} \in \cdot \mid X_t = x)$ is φ -irreducible if there exists a measure φ such that $\sum_t p^{(t)}(x, \cdot) \ll \varphi(\cdot)$, where $p^{(t)}(x, \cdot) = \mathbf{P}(X_t \in \cdot \mid X_0 = x)$ denotes the t -step transition kernel and \ll denotes absolute continuity. A Markov chain with invariant distribution π is called geometrically ergodic if there exists a positive function M and a constant $r \in (0, 1)$ such that

$$\|p^{(t)}(x, \cdot) - \pi(\cdot)\|_{\text{TV}} \leq M(x)r^t, \quad (1.5)$$

where $\|\cdot\|_{\text{TV}}$ denotes the total-variation norm. This condition ensures that the distribution of the Markov chain converges at a geometric rate to the invariant distribution. If the function M is bounded, then the Markov chain is said to be uniformly ergodic. Conditions such as (1.5) may be difficult to establish directly and are therefore substituted by suitable minorisation or drift conditions. A minorisation condition holds on a set C if there exist a probability measure ν , a positive integer t_0 , and $\delta > 0$ such that

$$p^{(t_0)}(x, B) \geq \delta \nu(B),$$

for all $x \in C$ and Borel sets B . In this case C is said to be a small set. Minorisation conditions have been used for obtaining rigorous bounds on the convergence of MCMC samplers, see e.g. [49].

If the entire state space is small, then the Markov chain is uniformly ergodic. Uniform ergodicity does typically not hold for Metropolis samplers, see Mengersen and Tweedie in [40] Theorem 3.1. Therefore useful sufficient conditions for geometric ergodicity are often given in the form of drift conditions [12, 40]. Drift conditions, established through the construction of appropriate Lyapunov functions, are also useful for establishing central limit theorems for MCMC algorithms, see [34, 42] and the references therein.

1.2.4 Efficiency of the MCMC algorithm

Roughly speaking, the arguments given above lead to the following desired properties of the estimator.

1. *Rare event efficiency:* Construct an unbiased estimator \hat{q}_T of p^{-1} according to (1.3) by finding a function v which approximates the conditional density (1.2). The choice of v controls the normalised variance of the estimator.
2. *Large sample efficiency:* Design the MCMC sampler, by finding an appropriate Gibbs sampler or a proposal density in the Metropolis-Hastings algorithm, such that the resulting Markov chain is geometrically ergodic.

1.3 Outline and contribution of this thesis

The outline and contribution of the thesis are as follows.

- a. General formulation of the algorithm in Section 2. In this section we present the formal methodology in how to set up the MCMC simulation for efficient rare-event computation. The probabilistic assumptions made are mild and the setting is for instance not restricted to heavy-tails. The two essential design choices are highlighted. Corresponding to rare-event efficiency and large sample efficiency.
- b. Application to heavy-tailed random walks in Section 3. In this section the MCMC methodology is applied to the problem of computing

$$p_n = \mathbf{P}(Y_1 + \cdots + Y_n > a_n),$$

where $a_n \rightarrow \infty$ sufficiently fast so that the probability tends to zero. The increments Y are assumed to be heavy-tailed. We present a Gibbs sampler to produce a Markov chain whose invariant distribution is the conditional distribution

$$\mathbf{P}((Y_1, \dots, Y_n) \in \cdot \mid Y_1 + \cdots + Y_n > a_n).$$

The Markov chain is shown to preserve stationarity and uniformly ergodic, ensuring the large sample efficiency. In addition we design an estimator for $1/p_n$ having vanishing normalised variance. Numerical experiments performed and comparison made between MCMC and best-performing existing importance sampling estimators as well as standard Monte Carlo.

- c. Application to heavy-tailed random sums in Section 4. In this section the MCMC methodology is applied to the problem of computing

$$p_n = \mathbf{P}(Y_1 + \cdots + Y_{N_n} > a_{N_n}),$$

where N is a random variable and $a_N \rightarrow \infty$ sufficiently fast so that the probability tends to zero. The increments Y are assumed to be heavy-tailed. We present a Gibbs sampler to produce a Markov chain whose invariant distribution is the conditional distribution

$$\mathbf{P}((N, Y_1, \dots, Y_N) \in \cdot \mid Y_1 + \cdots + Y_N > a_N).$$

The Markov chain is shown to preserve stationarity and uniformly ergodic, ensuring the large sample efficiency. In addition we design an estimator for $1/p_n$ having vanishing normalised variance. Numerical experiments performed and comparison made between MCMC and best-performing existing importance sampling estimators as well as standard Monte Carlo.

- d. Application to stochastic recurrent equations in Section 5. In this section the MCMC methodology is applied to the problem of computing $p_n = \mathbf{P}(X_n > a_n)$, where

$$\begin{aligned} X_n &= A_n X_{n-1} + B_n, \\ X_0 &= 0, \end{aligned}$$

and $a_n \rightarrow \infty$ sufficiently fast so that the probability tends to zero. The increments B are assumed to be regularly varying of index α and $\mathbf{E}[A^{\alpha+\epsilon}] < \infty$ for some $\epsilon > 0$. We present a Gibbs sampler to produce a Markov chain whose invariant distribution is the conditional distribution

$$\mathbf{P}((A_2, \dots, A_n, B_1, \dots, B_n) \in \cdot \mid X_n > a_n).$$

The Markov chain is shown to preserve stationarity and uniformly ergodic, ensuring the large sample efficiency. In addition we design an estimator for $1/p_n$ having vanishing normalised variance. Numerical experiments performed and comparison made between MCMC and best-performing existing importance sampling estimators as well as standard Monte Carlo.

- e. Application to computing probability of ruin in an insurance model with risky investments in Section 6...

A paper titled *Markov chain Monte Carlo for computing rare-event probabilities for a heavy-tailed random walk* by Gudmundsson and Hult [25] based on Sections 2, 3, and 4 in the thesis has been accepted for publication in the Journal of Applied Probability in June 2014.

2 General Markov chain Monte Carlo formulation

In this section the Markov chain Monte Carlo ideas are applied to the problem of computing an expectation. Here the setting is general, for instance, there is no assumption that densities with respect to Lebesgue measure exist.

Let X be a random variable with distribution F and h be a non-negative F -integrable function. The problem is to compute the expectation

$$\theta = \mathbf{E}[h(X)] = \int h(x)dF(x).$$

In the special case when F has density f and $h(x) = I\{x \in A\}$ this problem reduces to the simpler problem of computing the probability in (1.1). illustrated in Section 1.2.

The analogue of the conditional distribution in (1.2) is the distribution F_h given by

$$F_h(B) = \frac{1}{\theta} \int_B h(x)dF(x), \quad \text{for measurable sets } B.$$

Consider a Markov chain $(X_t)_{t \geq 0}$ having F_h as its invariant distribution. To define an estimator of θ^{-1} , consider a probability distribution V with $V \ll F_h$. Then it follows that $V \ll F$ and it is assumed that the density dV/dF is known. Consider the estimator of $\zeta = \theta^{-1}$ given by

$$\hat{\zeta}_T = \frac{1}{T} \sum_{t=0}^{T-1} u(X_t), \quad \text{where } u(x) = \frac{1}{\theta} \frac{dV}{dF_h}(x). \quad (2.1)$$

Note that u does not depend on θ because $V \ll F_h$ and therefore

$$u(x) = \frac{1}{\theta} \frac{dV}{dF_h}(x) = \frac{1}{h(x)} \frac{dV}{dF}(x),$$

for x such that $h(x) > 0$. The estimator (2.1) is a generalisation of the estimator (1.3) where one can think of v as the density of V with respect to Lebesgue measure. An estimator of θ can then be constructed as $\hat{\theta}_T = \hat{\zeta}_T^{-1}$.

The variance analysis of $\hat{\zeta}_T$ follows precisely the steps outlined in Section 1.2. The normalised variance is

$$\theta^2 \text{Var}_{F_h}(\hat{\zeta}_T) = \frac{\theta^2}{T} \text{Var}_{F_h}(u(X_0)) + \frac{2\theta^2}{T^2} \sum_{t=0}^{T-1} \sum_{s=t+1}^{T-1} \text{Cov}_{F_h}(u(X_s), u(X_t)), \quad (2.2)$$

where the first term can be rewritten, similarly to the display (1.4), as

$$\frac{\theta^2}{T} \text{Var}_{F_h}(u(X_0)) = \frac{1}{T} \left(\mathbf{E}_V \left[\frac{dV}{dF_h} \right] - 1 \right).$$

The analysis above indicates that an appropriate choice of V is such that $\mathbf{E}_V[\frac{dV}{dF_h}]$ is close to 1. Again, the ideal choice would be taking $V = F_h$ leading to zero variance. This choice is not feasible but nevertheless suggests selecting V as an approximation of F_h . As already noted this is similar to the ideology behind choosing an efficient importance sampling estimator. The difference being that here $V \ll F$ is required whereas in importance sampling F needs to be absolutely continuous with respect to the sampling distribution. The crude upper bound for the covariance term in (2.2) is valid, just as in Section 1.2.

2.1 Asymptotic efficiency criteria

Asymptotic efficiency can be conveniently formulated in terms of a limit criteria as a large deviation parameter tends to infinity. As is customary in problems related to rare-event simulation the problem at hand is embedded in a sequence of problems, indexed by $n = 1, 2, \dots$. The general setup is formalised as follows.

Let $(X^{(n)})_{n \geq 1}$ be a sequence of random variables with $X^{(n)}$ having distribution $F^{(n)}$. Let h be a non-negative function, integrable with respect to $F^{(n)}$, for each n . Suppose

$$\theta^{(n)} = \mathbf{E}[h(X^{(n)})] = \int h(x) dF^{(n)}(x) \rightarrow 0,$$

as $n \rightarrow \infty$. The problem is to compute $\theta^{(n)}$ for some large n .

Denote by $F_h^{(n)}$ the distribution with $dF_h^{(n)}/dF^{(n)} = h/\theta^{(n)}$. For the n th problem, a Markov chain $(X_t^{(n)})_{t=0}^{T-1}$ with invariant distribution $F_h^{(n)}$ is generated by an MCMC algorithm. The estimator of $\zeta^{(n)} = (\theta^{(n)})^{-1}$ is based on a probability distribution $V^{(n)}$, such that $V^{(n)} \ll F_h^{(n)}$, with known density with respect to $F^{(n)}$. An estimator $\hat{\zeta}_T^{(n)}$ of ζ is given by

$$\hat{\zeta}_T^{(n)} = \frac{1}{T} \sum_{t=0}^{T-1} u^{(n)}(X_t^{(n)}),$$

where

$$u^{(n)}(x) = \frac{1}{h(x)} \frac{dV^{(n)}}{dF^{(n)}}(x).$$

The heuristic efficiency criteria in Sections 1.2 can now be rigorously formulated as follows:

1. *Rare-event efficiency:* Select the probability distributions $V^{(n)}$ such that

$$(\theta^{(n)})^2 \text{Var}_{F_h^{(n)}}(u^{(n)}(X)) \rightarrow 0, \text{ as } n \rightarrow \infty.$$

2. *Large sample size efficiency:* Design the MCMC sampler, by finding an appropriate Gibbs sampler or a proposal density for the Metropolis-Hastings algorithm, such that, for each $n \geq 1$, the Markov chain $(X_t^{(n)})_{t \geq 0}$ is geometrically ergodic.

Remark 2.1. The rare-event efficiency criteria is formulated in terms of the efficiency of estimating $(\theta^{(n)})^{-1}$ by $\hat{\zeta}_T^{(n)}$. If one insists on studying the mean and variance of $\hat{\theta}_T^{(n)} = (\hat{\zeta}_T^{(n)})^{-1}$, then the effects of the transformation $x \mapsto x^{-1}$ must be taken into account. For instance, the estimator $\hat{\theta}_T^{(n)}$ is biased and its variance could be infinite. The bias can be reduced for instance via the delta method illustrated in [3, p. 76]. We also remark that even in the estimation of $(\theta^{(n)})^{-1}$ by $\hat{\zeta}_T^{(n)}$ there is a bias coming from the fact that the Markov chain not being perfectly stationary.

3 Heavy-tailed Random Walk

The MCMC methodology presented in Section 2 is here applied to compute the probability that a random walk $S_n = Y_1 + \dots + Y_n$, where Y_1, \dots, Y_n are non-negative, independent and heavy-tailed, exceeds a high threshold a_n . This problem has received some attention in the context of conditional Monte Carlo algorithms [2, 4] and importance sampling algorithms [35, 16, 11, 8].

In this section a Gibbs sampler is presented for sampling from the conditional distribution $\mathbf{P}((Y_1, \dots, Y_n) \in \cdot \mid S_n > a_n)$. The resulting Markov chain is proved to be uniformly ergodic. An estimator for $(p^{(n)})^{-1}$ of the form (2.1) is suggested with $V^{(n)}$ as the conditional distribution of (Y_1, \dots, Y_n) given $\max\{Y_1, \dots, Y_n\} > a_n$. The estimator is proved to have vanishing normalised variance when the distribution of Y_1 belongs to the class of subexponential distributions. The proof is elementary and is completed in a few lines. This is in sharp contrast to efficiency proofs for importance sampling algorithms for the same problem, which require more restrictive assumptions on the tail of Y_1 and tend to be long and technical [16, 11, 9]. The section is concluded with numerical experiments to illustrate the comparativeness with existing importance sampling algorithm and standard Monte Carlo.

3.1 A Gibbs sampler for computing $\mathbf{P}(S_n > a_n)$

Let Y_1, \dots, Y_n be non-negative independent and identically distributed random variables with common distribution F_Y and density f_Y with respect to some reference measure μ . Consider the random walk $S_n = Y_1 + \dots + Y_n$ and the problem of computing the probability

$$p^{(n)} = \mathbf{P}(S_n > a_n),$$

where $a_n \rightarrow \infty$ sufficiently fast that $p^{(n)} \rightarrow 0$ as $n \rightarrow \infty$.

It is convenient to denote by $\mathbf{Y}^{(n)}$ the n -dimensional random vector

$$\mathbf{Y}^{(n)} = (Y_1, \dots, Y_n)^\top,$$

and the set

$$A_n = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{1}^\top \mathbf{y} > a_n\},$$

where $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \dots, y_n)^\top$. With this notation

$$p^{(n)} = \mathbf{P}(S_n > a_n) = \mathbf{P}(\mathbf{1}^\top \mathbf{Y}^{(n)} > a_n) = \mathbf{P}(\mathbf{Y}^{(n)} \in A_n).$$

The conditional distribution

$$F_{A_n}^{(n)}(\cdot) = \mathbf{P}(\mathbf{Y}^{(n)} \in \cdot \mid \mathbf{Y}^{(n)} \in A_n),$$

has density

$$\frac{dF_{A_n}^{(n)}}{d\mu}(y_1, \dots, y_n) = \frac{\prod_{j=1}^n f_Y(y_j) I\{y_1 + \dots + y_n > a_n\}}{p^{(n)}}. \quad (3.1)$$

The first step towards defining the estimator of $p^{(n)}$ is to construct the Markov chain $(\mathbf{Y}_t^{(n)})_{t \geq 0}$ whose invariant density is given by (3.1) using a Gibbs sampler. In short, the Gibbs sampler updates one element of $\mathbf{Y}_t^{(n)}$ at a time keeping the other elements constant. Formally the algorithm proceeds as follows.

Algorithm 3.1. Start at an initial state $\mathbf{Y}_0^{(n)} = (Y_{0,1}, \dots, Y_{0,n})^\top$ where $Y_{0,1} + \dots + Y_{0,n} > a_n$. Given $\mathbf{Y}_t^{(n)} = (Y_{t,1}, \dots, Y_{t,n})^\top$, for some $t = 0, 1, \dots$, the next state $\mathbf{Y}_{t+1}^{(n)}$ is sampled as follows:

1. Draw j_1, \dots, j_n from $\{1, \dots, n\}$ without replacement and proceed by updating the components of $\mathbf{Y}_t^{(n)}$ in the order thus obtained.
2. For each $k = 1, \dots, n$, repeat the following.
 - (a) Let $j = j_k$ be the index to be updated and write

$$\mathbf{Y}_{t,-j} = (Y_{t,1}, \dots, Y_{t,j-1}, Y_{t,j+1}, \dots, Y_{t,n})^\top.$$

Sample $Y'_{t,j}$ from the conditional distribution of Y given that the sum exceeds the threshold. That is,

$$\mathbf{P}(Y'_{t,j} \in \cdot \mid \mathbf{Y}_{t,-j}) = \mathbf{P}\left(Y \in \cdot \mid Y + \sum_{k \neq j} Y_{t,k} > a_n\right).$$

- (b) Put $\mathbf{Y}'_t = (Y_{t,1}, \dots, Y_{t,j-1}, Y'_{t,j}, Y_{t,j+1}, \dots, Y_{t,n})^\top$.

3. Draw a random permutation π of the numbers $\{1, \dots, n\}$ from the uniform distribution and put $\mathbf{Y}_{t+1}^{(n)} = (Y'_{t,\pi(1)}, \dots, Y'_{t,\pi(n)})^\top$.

Iterate steps (1)-(3) until the entire Markov chain $(\mathbf{Y}_t^{(n)})_{t=0}^{T-1}$ is constructed.

Remark 3.2. (i) In the heavy-tailed setting the trajectories of the random walk leading to the rare event are likely to consist of one large increment (the big jump) while the other increments are average. The purpose of the permutation step is to force the Markov chain to mix faster by moving the big jump to different locations. However, the permutation step in Algorithm 3.1 is not really needed when considering the probability $\mathbf{P}(S_n > a_n)$. This is due to the fact that the summation is invariant of the ordering of the steps.

(ii) The algorithm requires sampling from the conditional distribution $\mathbf{P}(Y \in \cdot \mid Y > c)$ for arbitrary c . This is easy whenever inversion is feasible, see [3, p. 39], or acceptance/rejection sampling can be employed. There are, however, situations where sampling from the conditional distribution $\mathbf{P}(Y \in \cdot \mid Y > c)$ may be difficult, see [33, Section 2.2].

The following proposition confirms that the Markov chain $(\mathbf{Y}_t^{(n)})_{t \geq 0}$, generated by Algorithm 3.1, has $F_{A_n}^{(n)}$ as its invariant distribution.

Proposition 3.3. *The Markov chain $(\mathbf{Y}_t^{(n)})_{t \geq 0}$, generated by Algorithm 3.1, has the conditional distribution $F_{A_n}^{(n)}$ as its invariant distribution.*

Proof. The goal is to show that each updating step (Step 2 and 3) of the algorithm preserves stationarity. Since the conditional distribution $F_{A_n}^{(n)}$ is permutation invariant it is clear that Step 3 preserves stationarity. Therefore it is sufficient to consider Step 2 of the algorithm.

Let $P_j(\mathbf{y}, \cdot)$ denote the transition probability of the Markov chain $(\mathbf{Y}_t^{(n)})_{t \geq 0}$ corresponding to the j th component being updated. It is sufficient to show that,

for all $j = 1, \dots, m$ and all Borel sets of product form $B_1 \times \dots \times B_n \subset A_n$, the following equality holds:

$$F_{A_n}^{(n)}(B_1 \times \dots \times B_n) = \mathbf{E}_{F_{A_n}^{(n)}}[P_j(\mathbf{Y}, B_1 \times \dots \times B_n)].$$

Observe that, because $B_1 \times \dots \times B_n \subset A_n$,

$$\begin{aligned} F_{A_n}^{(n)}(B_1 \times \dots \times B_n) &= \mathbf{E}\left[\prod_{k=1}^n I\{Y_k \in B_k\} \mid S_n > a_n\right] \\ &= \frac{\mathbf{E}[I\{Y_j \in B_j\} I\{S_n > a_n\} \prod_{k \neq j} I\{Y_k \in B_k\}]}{\mathbf{P}(S_n > a_n)} \\ &= \frac{\mathbf{E}\left[\frac{\mathbf{E}[I\{Y_j \in B_j\} \mid Y_j > a_n - S_{n,-j}, \mathbf{Y}_{-j}^{(n)}] \prod_{k \neq j} I\{Y_k \in B_k\}}{\mathbf{P}(Y_j > a_n - S_{n,-j} \mid \mathbf{Y}_{-j}^{(n)})}\right]}{\mathbf{P}(S_n > a_n)} \\ &= \frac{\mathbf{E}[P_j(\mathbf{Y}^{(n)}, B_1 \times \dots \times B_n) \prod_{k \neq j} I\{Y_k \in B_k\}]}{P(S_n > a_n)} \\ &= \mathbf{E}[P_j(\mathbf{Y}^{(n)}, B_1 \times \dots \times B_n) \mid S_n > a_n] \\ &= \mathbf{E}_{F_{A_n}^{(n)}}[P_j(\mathbf{Y}, B_1 \times \dots \times B_n)], \end{aligned}$$

with the conventional notation of writing $\mathbf{Y}^{(n)} = (Y_1, \dots, Y_n)^\top$, $S_n = Y_1 + \dots + Y_n$, $\mathbf{Y}_{-j}^{(n)} = (Y_1, \dots, Y_{j-1}, Y_{j+1}, Y_n)^\top$ and $S_{n,-j} = Y_1 + \dots + Y_{j-1} + Y_{j+1} + \dots + Y_n$. \square

As for the ergodic properties, Algorithm 3.1 produces a Markov chain which is uniformly ergodic.

Proposition 3.4. *For each $n \geq 1$, the Markov chain $(\mathbf{Y}_t^{(n)})_{t \geq 0}$ is uniformly ergodic. In particular, it satisfies the following minorisation condition: there exists $\delta > 0$ such that*

$$\mathbf{P}(\mathbf{Y}_1^{(n)} \in B \mid \mathbf{Y}_0^{(n)} = \mathbf{y}) \geq \delta F_{A_n}^{(n)}(B),$$

for all $\mathbf{y} \in A_n$ and all Borel sets $B \subset A_n$.

Proof. Take an arbitrary $n \geq 1$. Uniform ergodicity can be deduced from the following minorisation condition (see [44]): there exists a probability measure ν , $\delta > 0$, and an integer t_0 such that

$$\mathbf{P}(\mathbf{Y}_{t_0}^{(n)} \in B \mid \mathbf{Y}_0^{(n)} = \mathbf{y}) \geq \delta \nu(B),$$

for every $\mathbf{y} \in A_n$ and Borel set $B \subset A_n$. Take $\mathbf{y} \in A_n$ and write $g(\cdot \mid \mathbf{y})$ for the density of $\mathbf{P}(\mathbf{Y}_1^{(n)} \in \cdot \mid \mathbf{Y}_0^{(n)} = \mathbf{y})$. The goal is to show that the minorisation condition holds with $t_0 = 1$, $\delta = p^{(n)}/n!$, and $\nu = F_{A_n}^{(n)}$.

For any $\mathbf{x} \in A_n$ there exists an ordering j_1, \dots, j_n of the numbers $\{1, \dots, n\}$ such that

$$y_{j_1} \leq x_{j_1}, \dots, y_{j_k} \leq x_{j_k}, y_{j_{k+1}} > x_{j_{k+1}}, \dots, y_{j_n} > x_{j_n},$$

for some $k \in \{0, \dots, n\}$. The probability to draw this particular ordering in Step 1 of the algorithm is at least $1/n!$. It follows that

$$\begin{aligned} g(\mathbf{x} \mid \mathbf{y}) &\geq \frac{1}{n!} \frac{f_Y(x_{j_1}) I\{x_{j_1} \geq a_n - \sum_{i \neq j_1} y_i\}}{\bar{F}_Y(a_n - \sum_{i \neq j_1} y_i)} \\ &\quad \times \frac{f_Y(x_{j_2}) I\{x_{j_2} \geq a_n - \sum_{i \neq j_1, j_2} y_i - x_{j_1}\}}{\bar{F}_Y(a_n - \sum_{i \neq j_1, j_2} y_i - x_{j_1})} \\ &\quad \vdots \\ &\quad \times \frac{f_Y(x_{j_n}) I\{x_{j_n} \geq a_n - x_{j_1} - \dots - x_{j_{n-1}}\}}{\bar{F}_Y(a_n - x_{j_1} - \dots - x_{j_{n-1}})}. \end{aligned}$$

By construction of the ordering j_1, \dots, j_n all the indicators are equal to 1 and the expression in the last display is bounded from below by

$$\frac{1}{n!} \prod_{j=1}^n f_Y(x_j) = \frac{p^{(n)}}{n!} \cdot \frac{\prod_{j=1}^n f_Y(x_j) I\{x_1 + \dots + x_n > a_n\}}{p^{(n)}}.$$

The proof is completed by integrating both sides of the inequality over any Borel set $B \subset A_n$. \square

Remark 3.5. To keep the proof of Proposition 3.4 simple, we have not used the permutation step of the algorithm in the proof and not tried to optimise δ . By taking advantage of the permutation step we believe that the constant δ could, with some additional effort, be increased by a factor $n!$.

3.2 Constructing an efficient estimator

Note that so far the distributional assumption of steps Y_1, \dots, Y_n of the random walk have been completely general. For the rare-event properties of the estimator the design of $V^{(n)}$ is essential and this is where the distributional assumptions become important. In this section a heavy-tailed random walk is considered. To be precise, assume that the variables Y_1, \dots, Y_n are nonnegative and that the tail of F_Y is heavy in the sense that there is a sequence (a_n) of real numbers such that

$$\lim_{n \rightarrow \infty} \frac{\mathbf{P}(S_n > a_n)}{\mathbf{P}(M_n > a_n)} = 1, \quad (3.2)$$

where M_n denotes the maximum of Y_1, \dots, Y_n . The class of distributions for which (3.2) holds is large and includes the subexponential distributions. General conditions on the sequence (a_n) for which (3.2) holds are given in [15], see also [13]. For instance, if \bar{F}_Y is regularly varying at ∞ with index $\beta > 1$ then (3.2) holds with $a_n = an$, for $a > 0$.

Next consider the choice of $V^{(n)}$. As observed in Section 2 a good approximation to the conditional distribution $F_{A_n}^{(n)}$ is a candidate for $V^{(n)}$. For a heavy-tailed random walk the “one big jump” heuristics says that the sum is large most likely because one of the steps is large. Based on the assumption (3.2) a good candidate for $V^{(n)}$ is the conditional distribution,

$$V^{(n)}(\cdot) = \mathbf{P}(\mathbf{Y}^{(n)} \in \cdot \mid M_n > a_n).$$

Then $V^{(n)}$ has a known density with respect to $F^{(n)}(\cdot) = \mathbf{P}(\mathbf{Y}^{(n)} \in \cdot)$ given by

$$\frac{dV^{(n)}}{dF^{(n)}}(\mathbf{y}) = \frac{1}{\mathbf{P}(M_n > a_n)} I\{\mathbf{y} : \bigvee_{j=1}^n y_j > a_n\} = \frac{I\{\mathbf{y} : \bigvee_{j=1}^n y_j > a_n\}}{1 - F_Y(a_n)^n}.$$

The estimator of $q^{(n)} = \mathbf{P}(S_n > a_n)^{-1}$ is then given by

$$\hat{q}_T^{(n)} = \frac{1}{T} \sum_{t=0}^{T-1} \frac{dV^{(n)}}{dF^{(n)}}(\mathbf{Y}_t^{(n)}) = \frac{1}{1 - F_Y(a_n)^n} \cdot \frac{1}{T} \sum_{t=0}^{T-1} I\{\bigvee_{j=1}^n Y_{t,j} > a_n\} \quad (3.3)$$

where $(\mathbf{Y}_t^{(n)})_{t \geq 0}$ is generated by Algorithm 3.1. Note that the estimator (3.3) can be viewed as the asymptotic approximation $(1 - F_Y(a_n)^n)^{-1}$ of $(p^{(n)})^{-1}$ multiplied by the random correction factor $\frac{1}{T} \sum_{t=0}^{T-1} I\{\bigvee_{j=1}^n Y_{t,j} > a_n\}$. The efficiency of this estimator is based on the fact that the random correction factor is likely to be close to 1 and has small variance.

Theorem 3.6. *Suppose that (3.2) holds. Then the estimator $\hat{q}_T^{(n)}$ in (3.3) has vanishing normalised variance for estimating $(p^{(n)})^{-1}$. That is,*

$$\lim_{n \rightarrow \infty} (p^{(n)})^2 \text{Var}_{F_{A_n}^{(n)}}(\hat{q}_T^{(n)}) = 0.$$

Proof. With $u^{(n)}(\mathbf{y}) = \frac{1}{1 - F_Y(a_n)^n} I\{\bigvee_{j=1}^n y_j > a_n\}$ it follows from (3.2) that

$$\begin{aligned} & (p^{(n)})^2 \text{Var}_{F_{A_n}^{(n)}}(u^{(n)}(\mathbf{Y}^{(n)})) \\ &= \frac{\mathbf{P}(S_n > a_n)^2}{\mathbf{P}(M_n > a_n)^2} \text{Var}_{F_{A_n}^{(n)}}(I\{\mathbf{Y} : \bigvee_{j=1}^n Y_j > a_n\}) \\ &= \frac{\mathbf{P}(S_n > a_n)^2}{\mathbf{P}(M_n > a_n)^2} \mathbf{P}(M_n > a_n \mid S_n > a_n) \mathbf{P}(M_n \leq a_n \mid S_n > a_n) \\ &= \frac{\mathbf{P}(S_n > a_n)}{\mathbf{P}(M_n > a_n)} \left(1 - \frac{\mathbf{P}(M_n > a_n)}{\mathbf{P}(S_n > a_n)}\right) \rightarrow 0. \end{aligned}$$

This completes the proof. \square

Remark 3.7. Theorem 3.6 covers a wide range of heavy-tailed distributions and even allows the number of steps to increase with n . Its proof is elementary. This is in sharp contrast to the existing proofs of efficiency (bounded relative error, say) for importance sampling algorithms that cover less general models and tend to be long and technical, see e.g. [16, 11, 9]. It must be mentioned, though, that Theorem 3.6 proves efficiency for computing $(p^{(n)})^{-1}$, whereas the authors of [16, 11, 9] prove efficiency for a direct computation of $p^{(n)}$.

3.3 Numerical experiments

First a note which applies to all of the numerical results presented in this thesis. The theoretical results guarantee that $\hat{q}_T^{(n)}$ is an efficient estimator of $(p^{(n)})^{-1}$. However, for comparison of existing algorithms the numerical experiments are based on $\hat{p}_T^{(n)} = (\hat{q}_T^{(n)})^{-1}$ as an estimator for $p^{(n)}$. The literature includes numerical comparison for many of the existing algorithms. In particular, in the setting of random sums. Numerical results for the algorithms by Dupuis et

al. [16], the hazard rate twisting algorithm by Juneja and Shahabuddin [35], and the conditional Monte Carlo algorithm by Asmussen and Kroese [4] can be found in [16]. Additional numerical results for the algorithms by Blanchet and Li [9], Dupuis et al. [16], and Asmussen and Kroese [4] can be found in [9]. From the existing results it appears as if the algorithm by Dupuis et al. [16] has the best performance. Therefore, we only include numerical experiments of the MCMC estimator and the estimator in [16], which is labelled IS.

By construction each simulation run of the MCMC algorithm only generates a single random variable (one simulation step) while both importance sampling and standard Monte Carlo generate n number of random variables (n simulation steps). Therefore the number of runs for the MCMC is scaled up by a factor of n so that all of the algorithms (MCMC, Monte Carlo and importance sampling) generate essentially the same number of random numbers. Thus getting a fairer comparison of the computer runtime between the three approaches.

Consider estimating $\mathbf{P}(S_n > a_n)$ where $S_n = Y_1 + \dots + Y_n$ with Y_1 having a Pareto distribution with density $f_Y(x) = \beta(x+1)^{-\beta-1}$ for $x \geq 0$. Let $a_n = an$. Each estimate is calculated using b number of batches, each consisting of T simulations in the case of importance sampling and standard Monte Carlo and Tn in the case of MCMC. The batch sample mean and sample standard deviation is recorded as well as the average runtime per batch. The results are presented in Table 1. The convergence of the algorithms can also be visualised by considering the point estimate as a function of number of simulation steps. This is presented in Figure 1. The MCMC algorithm appears to perform comparably with the importance sampling algorithm for p up to order 10^{-4} which is a relevant range in, say, insurance and finance. However for smaller p the MCMC appears to perform better. The improvement over importance sampling appears to increase as the event becomes more rare. This is due to the fact that the asymptotic approximation becomes better and better as the event becomes more rare.

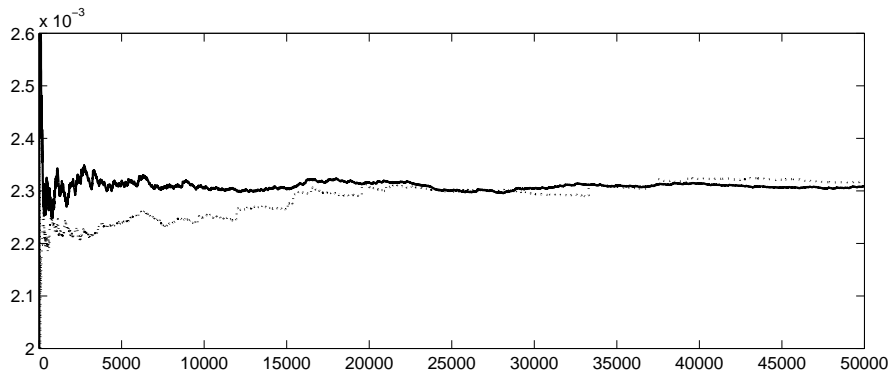


Figure 1: The figure illustrates the point estimate of $\mathbf{P}(S_n > a_n)$ as a function of the number of simulation steps, with $n = 5$, $a = 10$, $\beta = 2$. The estimate generated via the MCMC approach is drawn by a *solid line* and the estimate generated via IS is drawn by a *dotted line*.

Table 1: The table displays the batch mean and standard deviation of the estimates of $\mathbf{P}(S_n > a_n)$ as well as the average runtime per batch for time comparison. The number of batches run is b , each consisting of T simulations for importance sampling (IS) and standard Monte Carlo (MC) and Tn simulations for Markov chain Monte Carlo (MCMC). The asymptotic approximation is $p_{\max} = \mathbf{P}(\max\{Y_1, \dots, Y_n\} > a_n)$.

$b = 25, T = 10^5, \beta = 2, n = 5, a = 5, p_{\max} = 0.737\text{e-}2$			
	MCMC	IS	MC
Avg. est.	1.050e-2	1.048e-2	1.053e-2
Std. dev.	3e-5	9e-5	27e-5
Avg. time per batch(s)	12.8	12.7	1.4
$b = 25, T = 10^5, \beta = 2, n = 5, a = 20, p_{\max} = 4.901\text{e-}4$			
	MCMC	IS	MC
Avg. est.	5.340e-4	5.343e-4	5.380e-4
Std. dev.	6e-7	13e-7	770e-7
Avg. time per batch(s)	14.4	13.9	1.5
$b = 20, T = 10^5, \beta = 2, n = 5, a = 10^3, p_{\max} = 1.9992\text{e-}7$			
	MCMC	IS	
Avg. est.	2.0024e-7	2.0027e-7	
Std. dev.	3e-11	20e-11	
Avg. time per batch(s)	15.9	15.9	
$b = 20, T = 10^5, \beta = 2, n = 5, a = 10^4, p_{\max} = 1.99992\text{e-}9$			
	MCMC	IS	
Avg. est.	2.00025e-9	2.00091e-9	
Std. dev.	7e-14	215e-14	
Avg. time per batch(s)	15.9	15.9	
$b = 25, T = 10^5, \beta = 2, n = 20, a = 20, p_{\max} = 1.2437\text{e-}4$			
	MCMC	IS	MC
Avg. est.	1.375e-4	1.374e-4	1.444e-4
Std. dev.	2e-7	3e-7	492e-7
Avg. time per batch(s)	52.8	50.0	2.0
$b = 25, T = 10^5, \beta = 2, n = 20, a = 200, p_{\max} = 1.2494\text{e-}6$			
	MCMC	IS	MC
Avg. est.	1.2614e-6	1.2615e-6	1.2000e-6
Std. dev.	4e-10	12e-10	33,166e-10
Avg. time per batch(s)	49.4	48.4	1.9
$b = 20, T = 10^5, \beta = 2, n = 20, a = 10^3, p_{\max} = 4.9995\text{e-}8$			
	MCMC	IS	
Avg. est.	5.0091e-8	5.0079e-8	
Std. dev.	7e-12	66e-12	
Avg. time per batch(s)	53.0	50.6	
$b = 20, T = 10^5, \beta = 2, n = 20, a = 10^4, p_{\max} = 5.0000\text{e-}10$			
	MCMC	IS	
Avg. est.	5.0010e-10	5.0006e-10	
Std. dev.	2e-14	71e-14	
Avg. time per batch(s)	48.0	47.1	

4 Heavy-tailed Random Sum

The MCMC methodology presented in Section 2 and exemplified with a random walk in previous section, is here extended to compute the probability that a heavy-tailed random sum $S_N = Y_1 + \dots + Y_{N_n}$, where the number of steps N_n is random, and the Y 's are non-negative, independent and heavy-tailed, exceeds a high threshold a_n .

This is a relevant formulation in actuarial science, risk and queuing theory to name but a few. For instance, the stationary distribution of the waiting time and the workload of an $M/G/1$ queue can be represented as a random sum, see Amussen [1, Theorem 5.7, p. 237]. The classical Cramér-Lundberg model for the total claim amount faced by an insurance company is another standard example of a random sum.

This section follows the same structure as the previous one, a Gibbs sampler

is presented for sampling from the conditional distribution $\mathbf{P}((Y_1, \dots, Y_N) \in \cdot \mid S_N > a_n)$. The resulting Markov chain is proved to be uniformly ergodic. An estimator for $(p^{(N)})^{-1}$ of the form (2.1) is suggested with $V^{(n)}$ as the conditional distribution of (Y_1, \dots, Y_N) given $\max\{Y_1, \dots, Y_N\} > a_n$. The estimator is proved to have vanishing normalised variance when the distribution of Y_1 belongs to the class of subexponential distributions. The section is concluded with numerical experiments to illustrate the comparativeness with existing importance sampling algorithm and standard Monte Carlo.

4.1 A Gibbs sampler for computing $\mathbf{P}(S_{N_n} > a_n)$

Let Y_1, Y_2, \dots be non-negative independent random variables with common distribution F_Y and density f_Y . Let $(N^{(n)})_{n \geq 1}$ be integer valued random variables independent of Y_1, Y_2, \dots . Consider the random sum $S_{N^{(n)}} = Y_1 + \dots + Y_{N^{(n)}}$ and the problem of computing the probability

$$p^{(n)} = \mathbf{P}(S_{N^{(n)}} > a_n),$$

where $a_n \rightarrow \infty$ at an appropriate rate.

Denote by $\bar{\mathbf{Y}}^{(n)}$ the vector $(N^{(n)}, Y_1, \dots, Y_{N^{(n)}})^\top$. The conditional distribution of $\bar{\mathbf{Y}}^{(n)}$ given $S_{N^{(n)}} > a_n$ is given by

$$\begin{aligned} \mathbf{P}(N^{(n)} = k, (Y_1, \dots, Y_k) \in \cdot \mid S_{N^{(n)}} > a_n) \\ = \frac{\mathbf{P}((Y_1, \dots, Y_k) \in \cdot, S_k > a_n) \mathbf{P}(N^{(n)} = k)}{p^{(n)}}. \end{aligned} \quad (4.1)$$

A Gibbs sampler for sampling from the conditional distribution in (4.1) can be constructed essentially as in Algorithm 3.1. The only additional difficulty is to update the random number of steps in an appropriate way. In the following algorithm a particular distribution for updating the number of steps is proposed. To ease the notation the superscript n is suppressed in the description of the algorithm.

Algorithm 4.1. To initiate, draw N_0 from $\mathbf{P}(N \in \cdot)$ and $Y_{0,1}, \dots, Y_{0,N_0}$ such that $Y_{0,1} + \dots + Y_{0,N_0} > a_n$. Each iteration of the algorithm consists of the following steps. Suppose $\bar{\mathbf{Y}}_t = (k_t, y_{t,1}, \dots, y_{t,k_t})$ with $y_{t,1} + \dots + y_{t,k_t} > a_n$. Write $k_t^* = \min\{j : y_{t,1} + \dots + y_{t,j} > a_n\}$.

1. Sample the number of steps N_{t+1} from the distribution

$$p(k_{t+1} \mid k_t^*) = \frac{\mathbf{P}(N = k_{t+1}) I\{k_{t+1} \geq k_t^*\}}{P(N \geq k_t^*)}.$$

If $N_{t+1} > N_t$, sample $Y_{t+1,k_t+1}, \dots, Y_{t+1,N_{t+1}}$ independently from F_Y and put $\mathbf{Y}_t^{(1)} = (Y_{t,1}, \dots, Y_{t,k_t}, Y_{t+1,k_t+1}, \dots, Y_{t+1,N_{t+1}})$.

2. Proceed by updating all the individual steps as follows:

- (a) Draw $j_1, \dots, j_{N_{t+1}}$ from $\{1, \dots, N_{t+1}\}$ without replacement and proceed by updating the components of $\mathbf{Y}_t^{(1)}$ in the order thus obtained.
- (b) For each $k = 1, \dots, N_{t+1}$, repeat the following.

i. Let $j = j_k$ be the index to be updated and write

$$\mathbf{Y}_{t,-j}^{(1)} = (Y_{t,1}^{(1)}, \dots, Y_{t,j-1}^{(1)}, Y_{t,j+1}^{(1)}, \dots, Y_{t,N_{t+1}}^{(1)}).$$

Sample $Y_{t,j}^{(2)}$ from the conditional distribution of Y given that the sum exceeds the threshold. That is,

$$\mathbf{P}(Y_{t,j}^{(2)} \in \cdot \mid \mathbf{Y}_{t,-j}^{(1)}) = \mathbf{P}\left(Y \in \cdot \mid Y + \sum_{k \neq j} Y_{t,k}^{(1)} > a_n\right).$$

ii. Put $\mathbf{Y}_t^{(2)} = (Y_{t,1}^{(1)}, \dots, Y_{t,j-1}^{(1)}, Y_{t,j}^{(2)}, Y_{t,j+1}^{(1)}, \dots, Y_{t,N_{t+1}}^{(1)})^\top$.

(c) Draw a random permutation π of the numbers $\{1, \dots, N_{t+1}\}$ from the uniform distribution and put $\bar{\mathbf{Y}}_{t+1} = (N_{t+1}, Y_{t,\pi(1)}^{(2)}, \dots, Y_{t,\pi(N_{t+1})}^{(2)})$.

Iterate until the entire Markov Chain $(\bar{\mathbf{Y}}_t)_{t=0}^{T-1}$ is constructed.

Proposition 4.2. *The Markov chain $(\bar{\mathbf{Y}}_t)_{t \geq 0}$ generated by Algorithm 4.1 has the conditional distribution $\mathbf{P}((N, Y_1, \dots, Y_N) \in \cdot \mid Y_1 + \dots + Y_N > a_n)$ as its invariant distribution.*

Proof. The only essential difference from Algorithm 3.1 is the first step of the algorithm, where the number of steps and possibly the additional steps are updated. Therefore, it is sufficient to prove that the first step of the algorithm preserves stationarity. The transition probability of the first step, starting from a state $(k_t, y_{t,1}, \dots, y_{t,k_t})$ with $k_t^* = \min\{j : y_{t,1} + \dots + y_{t,j} > a_n\}$, can be written as follows.

$$\begin{aligned} & P^{(1)}(k_t, y_{t,1}, \dots, y_{t,k_t}; k_{t+1}, A_1 \times \dots \times A_{k_{t+1}}) \\ &= \mathbf{P}(N_{t+1} = k_{t+1}, (Y_{t,1}, \dots, Y_{t,k_{t+1}}) \in A_1 \times \dots \times A_{k_{t+1}} \\ &\quad \mid N_t = k_t, Y_{t,1} = y_{t,1}, \dots, Y_{t,k_t} = y_{t,k_t}) \\ &= \begin{cases} p(k_{t+1} \mid k_t^*) \prod_{k=1}^{k_{t+1}} I\{y_{t,k} \in A_k\}, & k_{t+1} \leq k_t, \\ p(k_{t+1} \mid k_t^*) \prod_{k=1}^{k_t} I\{y_{t,k} \in A_k\} \prod_{k=k_t+1}^{k_{t+1}} F_Y(A_k), & k_{t+1} > k_t. \end{cases} \end{aligned}$$

Consider the stationary probability of a set of the form $\{k_{t+1}\} \times A_1 \times \dots \times A_{k_{t+1}}$. With π denoting the conditional distribution $\mathbf{P}((N, Y_1, \dots, Y_N) \in \cdot \mid Y_1 + \dots + Y_N > a_n)$, it holds that

$$\begin{aligned} & \mathbf{E}_\pi[P^{(1)}(N_t, Y_{t,1}, \dots, Y_{t,N_t}; k_{t+1}, A_1 \times \dots \times A_{k_{t+1}})] \\ &= \frac{1}{\mathbf{P}(S_N > a_n)} \mathbf{E}[P^{(1)}(N, Y_1, \dots, Y_N; k_{t+1}, A_1 \times \dots \times A_{k_{t+1}}) I\{S_N > a_n\}] \end{aligned}$$

By conditioning on N and using independence of N and Y_1, Y_2, \dots the expression in the last display equals

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a_n)} \sum_{k_t=1}^{\infty} \mathbf{P}(N = k_t) \\ & \quad \times \mathbf{E}\left[P^{(1)}(k_t, Y_1, \dots, Y_{k_t}; k_{t+1}, A_1 \times \dots \times A_{k_{t+1}}) I\{S_{k_t} > a_n\}\right]. \end{aligned}$$

With $B_{k^*} = \{(y_1, y_2, \dots) \in \cup_{q=k^*}^{\infty} \mathbb{R}^q : \min\{j : y_1 + \dots + y_j > a\} = k^*\}$, $A_{k_t}^{\otimes} = A_1 \times \dots \times A_{k_t}$, and $A_{k_{t+1}}^{\otimes} = A_1 \times \dots \times A_{k_{t+1}}$ the expression in the last display can be written as

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a_n)} \left(\sum_{k_t=1}^{k_{t+1}} \mathbf{P}(N = k_t) \right. \\ & \quad \times \mathbf{E} \left[\sum_{k^*=1}^{k_t} I\{(Y_1, \dots, Y_{k_t}) \in B_{k^*}\} P^{(1)}(k_t, Y_1, \dots, Y_{k_t}; k_{t+1}, A_{k_{t+1}}^{\otimes}) \right] \\ & \quad + \sum_{k_t=k_{t+1}+1}^{\infty} \mathbf{P}(N = k_t) \\ & \quad \times \mathbf{E} \left[\sum_{k^*=1}^{k_{t+1}} I\{(Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*}\} P^{(1)}(k_t, Y_1, \dots, Y_{k_t}; k_{t+1}, A_{k_{t+1}}^{\otimes}) \right] \Bigg). \end{aligned}$$

Inserting the expression for $P^{(1)}$ the last expression equals

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a)} \left(\sum_{k_t=1}^{k_{t+1}} \mathbf{P}(N = k_t) \right. \\ & \quad \times \sum_{k^*=1}^{k_t} \mathbf{P}((Y_1, \dots, Y_{k_t}) \in B_{k^*} \cap A_{k_t}^{\otimes}) p(k_{t+1} | k^*) \prod_{j=k_t+1}^{k_{t+1}} F_Y(A_j) \\ & \quad + \sum_{k_t=k_{t+1}+1}^{\infty} \mathbf{P}(N = k_t) \sum_{k^*=1}^{k_{t+1}} \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) \Bigg). \end{aligned}$$

Changing the order of summation the last expression equals

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a_n)} \left(\sum_{k^*=1}^{k_{t+1}} \sum_{k_t=k^*}^{k_{t+1}} \mathbf{P}(N = k_t) \right. \\ & \quad \times \mathbf{P}((Y_1, \dots, Y_{k_t}) \in B_{k^*} \cap A_{k_t}^{\otimes}) p(k_{t+1} | k^*) \prod_{j=k_t+1}^{k_{t+1}} F_Y(A_j) \\ & \quad + \sum_{k^*=1}^{k_{t+1}} \sum_{k_t=k_{t+1}+1}^{\infty} \mathbf{P}(N = k_t) \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) \Bigg). \end{aligned}$$

Since $\mathbf{P}((Y_1, \dots, Y_{k_t}) \in B_{k^*} \cap A_{k_t}^{\otimes}) \prod_{j=k_t+1}^{k_{t+1}} F_Y(A_j) = \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes})$ the last expression equals

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a_n)} \left(\sum_{k^*=1}^{k_{t+1}} \sum_{k_t=k^*}^{k_{t+1}} \mathbf{P}(N = k_t) \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) \right. \\ & \quad + \sum_{k^*=1}^{k_{t+1}} \sum_{k_t=k_{t+1}+1}^{\infty} \mathbf{P}(N = k_t) \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) \Bigg). \end{aligned}$$

Summing over k_t the last expression equals

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a_n)} \left(\sum_{k^*=1}^{k_{t+1}} \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) \mathbf{P}(k^* \leq N \leq k_{t+1}) \right. \\ & \quad \left. + \sum_{k^*=1}^{k_{t+1}} \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) \mathbf{P}(N \geq k_{t+1} + 1) \right). \end{aligned}$$

From the definition of $p(k_{t+1} | k^*)$ it follows that the last expression equals

$$\begin{aligned} & \frac{1}{\mathbf{P}(S_N > a_n)} \sum_{k^*=1}^{k_{t+1}} \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) p(k_{t+1} | k^*) P(N \geq k^*) \\ &= \frac{1}{\mathbf{P}(S_N > a_n)} \sum_{k^*=1}^{k_{t+1}} \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in B_{k^*} \cap A_{k_{t+1}}^{\otimes}) P(N = k_{t+1}) \\ &= \frac{1}{\mathbf{P}(S_N > a_n)} \mathbf{P}((Y_1, \dots, Y_{k_{t+1}}) \in A_{k_{t+1}}^{\otimes}) P(N = k_{t+1}) \\ &= \mathbf{P}(N = k_{t+1}, (Y_1, \dots, Y_{k_{t+1}}) \in A_{k_{t+1}}^{\otimes} | Y_1 + \dots + Y_N > a_n), \end{aligned}$$

which is the desired invariant distribution. This completes the proof. \square

Proposition 4.3. *The Markov chain $(\bar{\mathbf{Y}}_t)_{t \geq 0}$ generated by Algorithm 4.1 is uniformly ergodic. In particular, it satisfies the following minorisation condition: there exists $\delta > 0$ such that*

$$\mathbf{P}(\bar{\mathbf{Y}}_1 \in B | \bar{\mathbf{Y}}_0 = \bar{\mathbf{y}}) \geq \delta \mathbf{P}((N, Y_1, \dots, Y_N) \in B | Y_1 + \dots + Y_N > a_n),$$

for all $\bar{\mathbf{y}} \in A = \cup_{k \geq 1} \{(k, y_1, \dots, y_k) : y_1 + \dots + y_k > a_n\}$ and all Borel sets $B \subset A$.

The proof requires only a minor modification from the non-random case, Proposition 3.4, and is therefore omitted.

4.2 Constructing an efficient estimator

Now consider the distributional assumptions and the design of $V^{(n)}$. The main focus is on the rare event properties of the estimator and therefore the large deviation parameter n will be suppressed to ease notation. Let the distribution of the number of steps $\mathbf{P}(N^{(n)} \in \cdot)$ to depend on n . By a similar reasoning as in the case of non-random number of steps the following assumption are imposed: the variables $N^{(n)}, Y_1, Y_2, \dots$ and the numbers a_n are such that

$$\lim_{n \rightarrow \infty} \frac{\mathbf{P}(Y_1 + \dots + Y_{N^{(n)}} > a_n)}{\mathbf{P}(M_{N^{(n)}} > a_n)} = 1, \quad (4.2)$$

where $M_k = \max\{Y_1, \dots, Y_k\}$. Note that the denominator can be expressed as

$$\begin{aligned} \mathbf{P}(M_{N^{(n)}} > a_n) &= \sum_{k=1}^{\infty} \mathbf{P}(M_k > a_n) \mathbf{P}(N^{(n)} = k) \\ &= \sum_{k=1}^{\infty} [1 - F_Y(a_n)^k] \mathbf{P}(N^{(n)} = k) \\ &= 1 - g_{N^{(n)}}(F_Y(a_n)), \end{aligned}$$

where $g_{N^{(n)}}(t) = \mathbf{E}[t^{N^{(n)}}]$ is the generating function of $N^{(n)}$. Sufficient conditions for (4.2) to hold are given in [37], Theorem 3.1. For instance, if \overline{F}_Y is regularly varying at ∞ with index $\beta > 1$ and $N^{(n)}$ has Poisson distribution with mean $\lambda_n \rightarrow \infty$, as $n \rightarrow \infty$, then (4.2) holds with $a_n = a\lambda_n$, for $a > 0$.

Similarly to the non-random setting a good candidate for $V^{(n)}$ is the conditional distribution,

$$V^{(n)}(\cdot) = \mathbf{P}(\overline{\mathbf{Y}}^{(n)} \in \cdot \mid M_{N^{(n)}} > a_n).$$

Then $V^{(n)}$ has a known density with respect to $F^{(n)}(\cdot) = \mathbf{P}(\overline{\mathbf{Y}}^{(n)} \in \cdot)$ given by

$$\begin{aligned} \frac{dV^{(n)}}{dF^{(n)}}(k, y_1, \dots, y_k) &= \frac{1}{\mathbf{P}(M_{N^{(n)}} > a_n)} I\{(y_1, \dots, y_k) : \bigvee_{j=1}^k y_j > a_n\} \\ &= \frac{1}{1 - g_{N^{(n)}}(F_Y(a_n))} I\{(y_1, \dots, y_k) : \bigvee_{j=1}^k y_j > a_n\}. \end{aligned}$$

The estimator of $q^{(n)} = \mathbf{P}(S_n > a_n)^{-1}$ is given by

$$\hat{q}_T^{(n)} = \frac{1}{T} \sum_{t=0}^{T-1} \frac{dV^{(n)}}{dF^{(n)}}(\overline{\mathbf{Y}}_t^{(n)}) = \frac{1}{g_{N^{(n)}}(F_Y(a_n))} \cdot \frac{1}{T} \sum_{t=0}^{T-1} I\{\bigvee_{j=1}^{N_t} Y_{t,j} > a_n\}, \quad (4.3)$$

where $(\overline{\mathbf{Y}}_t^{(n)})_{t \geq 0}$ is generated by Algorithm 4.1.

Theorem 4.4. *Suppose (4.2) holds. The estimator $\hat{q}_T^{(n)}$ in (4.3) has vanishing normalised variance. That is,*

$$\lim_{n \rightarrow \infty} (p^{(n)})^2 \text{Var}_{\pi_n}(\hat{q}_T^{(n)}) = 0,$$

where π_n denotes the conditional distribution $\mathbf{P}(\overline{\mathbf{Y}}^{(n)} \in \cdot \mid S_{N^{(n)}} > a_n)$.

Remark 4.5. Because the distribution of $N^{(n)}$ may depend on n Theorem 4.4 covers a wider range of settings for random sums than those studied in [16, 9] where the authors present provably efficient importance sampling algorithms.

Proof. Since $p^{(n)} = \mathbf{P}(S_{N^{(n)}} > a_n)$ and

$$u^{(n)}(k, y_1, \dots, y_k) = \frac{I\{\bigvee_{j=1}^k y_j > a_n\}}{\mathbf{P}(M_{N^{(n)}} > a_n)},$$

it follows that

$$\begin{aligned} [p^{(n)}]^2 \text{Var}_{\pi_n}(u^{(n)}(\overline{\mathbf{Y}}^{(n)})) &= \frac{\mathbf{P}(S_{N^{(n)}} > a_n)^2}{\mathbf{P}(M_{N^{(n)}} > a_n)^2} \text{Var}_{\pi_n}(I\{\bigvee_{j=1}^{N^{(n)}} Y_j > a_n\}) \\ &= \frac{\mathbf{P}(S_{N^{(n)}} > a_n)^2}{\mathbf{P}(M_{N^{(n)}} > a_n)^2} \mathbf{P}(M_{N^{(n)}} > a_n \mid S_{N^{(n)}} > a_n) \mathbf{P}(M_{N^{(n)}} \leq a_n \mid S_{N^{(n)}} > a_n) \\ &= \frac{\mathbf{P}(S_{N^{(n)}} > a_n)}{\mathbf{P}(M_{N^{(n)}} > a_n)} \left(1 - \frac{\mathbf{P}(M_{N^{(n)}} > a_n)}{\mathbf{P}(S_{N^{(n)}} > a_n)}\right) \rightarrow 0, \end{aligned}$$

by (4.2). This completes the proof. \square

4.3 Numerical experiments

By construction each simulation run of the MCMC algorithm only generates a single random variable (one simulation step) while both importance sampling and standard Monte Carlo generate $N + 1$ number of random variables ($N + 1$ simulation steps). Therefore the number of runs for the MCMC is scaled up by a factor of $\mathbf{E}[N] + 1$ so that all of the algorithms (MCMC, Monte Carlo and importance sampling) generate essentially the same number of random numbers. Thus getting a fairer comparison of the computer runtime between the three approaches.

Consider estimating $\mathbf{P}(S_N > a_\rho)$ where $S_N = Y_1 + \dots + Y_N$ with N Geometrically distributed $\mathbf{P}(N = k) = (1 - \rho)^{k-1} \rho$ for $k = 1, 2, \dots$ and $a_\rho = a \mathbf{E}[N] = a/\rho$. The estimator considered here is $\hat{p}_T = (\hat{q}_T)^{-1}$ with \hat{q}_T as in (4.3). Each estimate is calculated using b number of batches, each consisting of T simulations in the case of importance sampling and standard Monte Carlo and $T \mathbf{E}[N]$ in the case of MCMC. The results are presented in Table 2. The MCMC algorithm appears to outperform the importance sampling algorithm consistently for different choices of the parameters.

We remark that in our simulation with $\rho = 0.2$, $a = 5 \cdot 10^9$ the sample standard deviation of the MCMC estimate is zero. This is because we did not observe any indicators $I\{\bigvee_{j=1}^n y_{t,j} > a_\rho\}$ being equal to 0 in this case.

Table 2: The table displays the batch mean and standard deviation of the estimates of $\mathbf{P}(S_N > a_\rho)$ as well as the average runtime per batch for time comparison. The number of batches run is b , each consisting of T simulations for importance sampling (IS) and standard Monte Carlo (MC) and $T \mathbf{E}[N]$ simulations for Markov chain Monte Carlo (MCMC). The asymptotic approximation is $p_{\max} = \mathbf{P}(\max\{Y_1, \dots, Y_N\} > a_\rho)$.

$b = 25, T = 10^5, \beta = 1, \rho = 0.2, a = 10^2, p_{\max} = 0.990\text{e-}2$			
	MCMC	IS	MC
Avg. est.	1.149e-2	1.087e-2	1.089e-2
Std. dev.	4e-5	6e-5	35e-5
Avg. time per batch(s)	25.0	11.0	1.2
$b = 25, T = 10^5, \beta = 1, \rho = 0.2, a = 10^3, p_{\max} = 0.999\text{e-}3$			
	MCMC	IS	MC
Avg. est.	1.019e-3	1.012e-3	1.037e-3
Std. dev.	1e-6	3e-6	76e-6
Avg. time per batch(s)	25.8	11.1	1.2
$b = 20, T = 10^6, \beta = 1, \rho = 0.2, a = 5 \cdot 10^4, p_{\max} = 2.000000\text{e-}8$			
	MCMC	IS	
Avg. est.	2.000003e-8	1.999325e-8	
Std. dev.	6e-14	1114e-14	
Avg. time per batch(s)	385.3	139.9	
$b = 20, T = 10^6, \beta = 1, \rho = 0.2, a = 5 \cdot 10^9, p_{\max} = 2.0000\text{e-}10$			
	MCMC	IS	
Avg. est.	2.0000e-10	1.9998e-10	
Std. dev.	0	13e-14	
Avg. time per batch(s)	358.7	130.9	
$b = 25, T = 10^5, \beta = 1, \rho = 0.05, a = 10^3, p_{\max} = 0.999\text{e-}3$			
	MCMC	IS	MC
Avg. est.	1.027e-3	1.017e-3	1.045e-3
Std. dev.	1e-6	4e-6	105e-6
Avg. time per batch(s)	61.5	44.8	1.3
$b = 25, T = 10^5, \beta = 1, \rho = 0.05, a = 5 \cdot 10^9, p_{\max} = 1.9999\text{e-}6$			
	MCMC	IS	MC
Avg. est.	2.0002e-6	2.0005e-6	3.2000e-6
Std. dev.	1e-10	53e-10	55,678e-10
Avg. time per batch(s)	60.7	45.0	1.3

5 Stochastic Recurrence Equations

The MCMC methodology presented in Section 2 is here applied to compute the probability that a solution X_m to a recurrence equation $X_m = A_m X_{m-1} + B_m$, where the innovations B are regularly varying with index α and $\mathbf{E}[A^{\alpha+\epsilon}] < \infty$ for some $\epsilon > 0$, exceeds a high threshold c_n . This problem has been considered using importance sampling scheme by Hult, Blanchet and Leder in [27].

In this section a Gibbs sampler is presented for sampling from the conditional distribution $\mathbf{P}(A_1, \dots, A_m, B_1, \dots, B_m \mid X_m > c_n)$. The resulting Markov chain is proved to be uniformly ergodic. An estimator for $(p^{(n)})^{-1}$ of the form (2.1) is suggested with $V^{(n)}$ as the conditional distribution of $(A_1, \dots, A_m, B_1, \dots, B_m)$ given $\{A_k > a, \forall k\} \cap \{\exists! j : B_j a^{m-j} > c_n\}$. The estimator is proved to have vanishing normalised variance under the probabilistic assumptions mentioned above. The proof is elementary and is completed in a few lines. The section is concluded with numerical experiments to illustrate the comparativeness with existing importance sampling algorithm and standard Monte Carlo.

5.1 A Gibbs sampler for computing $\mathbf{P}(X_m > c_n)$

Fix m and let $\mathbf{A} = (A_2, \dots, A_m)$ and $\mathbf{B} = (B_1, \dots, B_m)$ be independent sequences of independent and identically distributed random variables. Let A be a generic random variable for an element of the sequence \mathbf{A} and likewise B for an element of the sequence \mathbf{B} .

Consider the solution $(X_k)_{k=0}^m$ to the stochastic recurrence equation

$$\begin{aligned} X_k &= A_k X_{k-1} + B_k, \quad \text{for } k = 1, \dots, m, \\ X_0 &= 0. \end{aligned}$$

The solution $(X_k)_{k=0}^m$ can be written as a randomly weighted random walk

$$X_k = B_k + A_k B_{k-1} + \dots + A_k A_{k-1} \dots A_2 B_1 + A_k \dots A_1 x_0, \quad k = 1, \dots, m. \quad (5.1)$$

Our interest is in the problem of computing $p^{(n)} = \mathbf{P}(X_m > c_n)$, where $c_n \rightarrow \infty$. To this end we will propose a Gibbs sampler that produces a Markov chain with the conditional distribution

$$F_{c_n}^{(m)}(\cdot) = \mathbf{P}((\mathbf{A}, \mathbf{B}) \in \cdot \mid X_m > c_n) \quad (5.2)$$

as its invariant distribution. In addition we will suggest a choice of the probability distribution $V^{(n)}$ with good asymptotic properties.

The Markov chain $(\mathbf{A}_t, \mathbf{B}_t)_{t \geq 0}$ is constructed by the following algorithm, where the elements are updated sequentially in such a way that the weighted random walk exceeds the threshold after each individual update. Formally the algorithm is given as follows. An empty product, such as $\prod_{j=m+1}^m A_j$, is interpreted as 1.

Algorithm 5.1. Start with initial state $(\mathbf{A}_0^{(m)}, \mathbf{B}_0^{(m)}) = (A_{0,2}, \dots, A_{0,m}, B_{0,1}, \dots, B_{0,m})$ where $X_0^{(m)} = B_{0,m} + \sum_{i=1}^{m-1} B_{0,i} \prod_{j=i+1}^m A_{0,j} > c_n$. Given $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})$, for some $t = 0, 1, \dots$, the next state $(\mathbf{A}_{t+1}^{(m)}, \mathbf{B}_{t+1}^{(m)})$ is sampled as follows:

1. Draw a randomized ordering j_1, \dots, j_{2m} of $\{1, \dots, 2m\}$ and proceed updating $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})$ in the order thus obtained.
2. For $l = 1, \dots, 2m$, set $k = j_l$ and do the following:
 - i. If $k \in \{1, \dots, m\}$ then $A_{t,k}$ is to be updated. Sample A' from the conditional distribution

$$\mathbf{P}(A' \in \cdot \mid A' > s),$$

where

$$s = \max \left\{ \frac{c_n - \sum_{i=k}^m B_{t,i} \prod_{j=i+1}^m A_{t,j}}{\sum_{i=1}^{k-1} B_{t,i} \prod_{j=i+1, \neq k}^m A_{t,j}}, 0 \right\}.$$

Put $\mathbf{A}_{t+1}^{(m)} = (A_{t,1}, \dots, A_{t,k-1}, A', A_{t,k+1}, \dots, A_{t,m})$ and $\mathbf{B}_{t+1}^{(m)} = \mathbf{B}_t^{(m)}$.

- ii. If $k \in \{m+1, \dots, 2m\}$ then $B_{t,(k-m)}$ is to be updated. Sample B' from the conditional distribution

$$\mathbf{P}(B' \in \cdot \mid B' > s),$$

where

$$s = \max \left\{ \frac{c_n - \sum_{i=1, \neq (k-m)}^m B_{t,i} \prod_{j=i+1}^m A_{t,j}}{A_{t,m} \cdots A_{t,(k-m)+1}}, 0 \right\}.$$

Put $\mathbf{A}_{t+1}^{(m)} = \mathbf{A}_t^{(m)}$ and $\mathbf{B}_{t+1}^{(m)} = (B_{t,1}, \dots, B_{t,(k-m)-1}, B', B_{t,(k-m)+1}, \dots, B_{t,m})$.

Iterate steps 1 and 2 until the entire Markov chain $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})_{t=0}^{T-1}$ is constructed.

The Markov chain $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})_{t \geq 0}$ constructed by Algorithm 5.1 has $F_{c_n}^{(m)}$ as its invariant probability distribution.

Proposition 5.2. *The Markov chain $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})_{t \geq 0}$ generated by Algorithm 5.1, has the conditional distribution $F_{c_n}^{(m)}$ as its invariant distribution.*

Proof. Note that it is sufficient to show that each updating step (Step 2i and 2ii in the Algorithm) preserves stationarity.

Consider the updating steps (Step 2i and 2ii). Let m be given and set $P_k^A(\mathbf{a}^{(m)}, \mathbf{b}^{(m)}, \cdot)$ and $P_k^B(\mathbf{a}^{(m)}, \mathbf{b}^{(m)}, \cdot)$ to be the transition probability of the Markov chain $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})_{t \geq 0}$ where the k th element of $\mathbf{A}_t^{(m)}$ and $\mathbf{B}_t^{(m)}$ is updated, respectively. Let

$$R = \{(A_1, \dots, A_m, B_1, \dots, B_m) \mid X_m > c_n\},$$

and observe that if A_k is to be updated conditioned on $X_m > c_n$ then

$$A_k > \frac{c_n - \sum_{i=k}^m B_{t,i} \prod_{j=i+1}^m A_{t,j}}{\sum_{i=1}^{k-1} B_{t,i} \prod_{j=i+1, \neq k}^m A_{t,j}} =: s_{A_k},$$

and similarly, if B_k is to be updated conditioned on $X_m > c_n$ then

$$B_k > \frac{c_n - \sum_{i=1, \neq (k-m)}^m B_{t,i} \prod_{j=i+1}^m A_{t,j}}{A_{t,m} \cdots A_{t,(k-m)+1}} =: s_{B_k}.$$

To prove that stationarity is preserved under updating via Step 2i it is sufficient to show that for arbitrary $k \in \{1, \dots, m\}$ and $D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m \subset R$ then it holds that

$$\begin{aligned} F_{c_n}^{(m)}(D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m) \\ = \mathbf{E}_{F_{c_n}^{(m)}}[P_k^A(A_1, \dots, A_m, B_1, \dots, B_m, D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m)]. \end{aligned} \quad (5.3)$$

Similarly to prove that stationarity is preserved under updating via Step 2ii it is sufficient to show

$$\begin{aligned} F_{c_n}^{(m)}(D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m) \\ = \mathbf{E}_{F_{c_n}^{(m)}}[P_k^B(A_1, \dots, A_m, B_1, \dots, B_m, D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m)]. \end{aligned} \quad (5.4)$$

The following computation shows that (5.3) holds.

$$\begin{aligned} F_{c_n}^{(m)}(D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m) \\ = \mathbf{E}_{F_{c_n}^{(m)}}\left[\prod_{j=1}^m I\{A_j \in D_j\} \prod_{i=1}^m I\{B_i \in E_i\}\right] \\ = \frac{\mathbf{E}[I\{A_k \in D_k\} I\{X_m > c_n\} \cdot \prod_{j=1, \neq k}^m I\{A_j \in D_j\} \prod_{i=1}^m I\{B_i \in E_i\}]}{\mathbf{P}(X_m > c_n)} \\ = \frac{\mathbf{E}\left[\frac{\mathbf{E}[I\{A_k \in D_k\} | A_k > s_{A_k}, \mathbf{A}_{-k}, \mathbf{B}]}{\mathbf{P}(A_k > s_{A_k})} \cdot \prod_{j=1, \neq k}^m I\{A_j \in D_j\} \prod_{i=1}^m I\{B_i \in E_i\}\right]}{\mathbf{P}(X_m > c_n)} \\ = \frac{\mathbf{E}[P_k^A(\mathbf{A}, \mathbf{B}, D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m) \cdot \prod_{j=1, \neq k}^m I\{A_j \in D_j\} \prod_{i=1}^m I\{B_i \in E_i\}]}{\mathbf{P}(X_m > c_n)} \\ = \mathbf{E}[P_k^A(\mathbf{A}, \mathbf{B}, D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m) | X_m > c_n] \\ = \mathbf{E}_{F_{c_n}^{(m)}}[P_k^A(\mathbf{A}, \mathbf{B}, D_1 \times \dots \times D_m \times E_1 \times \dots \times E_m)], \end{aligned}$$

with the conventional notation $\mathbf{A}_{-k} = (A_1, \dots, A_{k-1}, A_{k+1}, \dots, A_m)$.

The proof is completed by showing that (5.4) holds with similar computation as above. \square

The Markov chain $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})_{t \geq 0}$ constructed by Algorithm 5.1 is uniformly ergodic, thus ensuring large-sample efficiency.

Proposition 5.3. *For any $m \geq 1$, the Markov chain $(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)})_{t \geq 0}$ is uniformly ergodic.*

Proof. Let $m \geq 1$ be given and set

$$R = \{(A_1, \dots, A_m, B_1, \dots, B_m) \mid X_m > c_n\}.$$

Uniform ergodicity follows from the minorization condition (see Nummelin [44]) : there exists a probability measure ν , $\delta > 0$ and $t_0 \in \mathbb{N}$ such that

$$\mathbf{P}((\mathbf{A}_{t_0}^{(m)}, \mathbf{B}_{t_0}^{(m)}) \in D \times E \mid (\mathbf{A}_0^{(m)}, \mathbf{B}_0^{(m)}) = (\mathbf{a}, \mathbf{b})) \geq \delta \nu(D \times E),$$

for any (\mathbf{a}, \mathbf{b}) and $D \times E \subset R$. The goal is to prove this inequality for $t_0 = 1$, $\delta = p^{(n)}/(2m)!$ and $\nu = F_{c_n}^{(m)}$.

Take $\mathbf{c} = (\mathbf{a}, \mathbf{b})$ and let $g(\cdot \mid \mathbf{a}, \mathbf{b})$ be the density of $\mathbf{P}(\mathbf{A}_1, \mathbf{B}_1 \in \cdot \mid \mathbf{A}_0, \mathbf{B}_0 = \mathbf{a}, \mathbf{b})$.

Observe that for any $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in R$ there exists an ordering j_1, \dots, j_{2m} of $\{1, \dots, 2m\}$ such that

$$\begin{aligned} c_{j_1} &\leq z_{j_1}, \dots, c_{j_k} \leq z_{j_k} \\ c_{j_{k+1}} &\geq z_{j_{k+1}}, \dots, c_{j_{2m}} \geq z_{j_{2m}}, \end{aligned}$$

for some k . When updating from \mathbf{c} to \mathbf{z} using this particular ordering, then first all of elements in \mathbf{z} which are larger than their counterparts in \mathbf{c} are updated, and then all of the elements in \mathbf{z} which are smaller are updated. This guarantees that after every updating step, the updated vector belongs to R .

The probability for this particular ordering is $1/(2m)!$. To simplify notation, introduce

$$Z_k = \begin{cases} A_i & \text{if update } j_k \text{ corresponds to updating } A_i \text{ for some } i \\ B_i & \text{if update } j_k \text{ corresponds to updating } B_i \text{ for some } i \end{cases}$$

and

$$s_{Z_k} = \begin{cases} s_{A_i} & \text{if update } j_k \text{ corresponds to updating } A_i \text{ for some } i \\ s_{B_i} & \text{if update } j_k \text{ corresponds to updating } B_i \text{ for some } i \end{cases}$$

Therefore

$$\begin{aligned} g(\mathbf{x}, \mathbf{y}) &= \frac{1}{(2m)!} \frac{f_{Z_1}(z_{j_1})I\{Z_1 > s_{Z_1}\}}{\mathbf{P}(Z > s_{Z_1})} \\ &\quad \times \frac{f_{Z_2}(z_{j_2})I\{Z_2 > s_{Z_2}\}}{\mathbf{P}(Z > s_{Z_2})} \\ &\quad \vdots \\ &\quad \times \frac{f_{Z_{2m}}(z_{j_{2m}})I\{Z_{2m} > s_{Z_{2m}}\}}{\mathbf{P}(Z > s_{Z_{2m}})}. \end{aligned}$$

By construction all of the indicator functions are equal to 1 and the normalizing probabilities are bounded by 1 so the last display is bounded from below by

$$\frac{1}{(2m)!} \prod_{k=1}^{2m} f_{Z_k}(z_k) = \frac{p^{(n)}}{(2m)!} \cdot \frac{\prod_{k=1}^{2m} f_{Z_k}(z_k)I\{\mathbf{z} \in R\}}{p^{(m)}}.$$

The proof is completed by integrating both sides. \square

Remark 5.4. The lower bound δ in the proof of Proposition 5.3 can be chosen to be larger, but that would complicate and lengthen the proof.

5.2 Constructing an efficient estimator

As mentioned in Section 2 a good candidate for $V^{(n)}$ is a probability distribution

$$\mathbf{P}((\mathbf{A}, \mathbf{B}) \in \cdot \mid (\mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \in R^{(n)}),$$

where $r^{(n)} = \mathbf{P}((\mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \in R^{(n)})$ is asymptotically close to $p^{(n)} = \mathbf{P}(X_m > c_n)$ in the sense that $r^{(n)}/p^{(n)} \rightarrow 1$ as $n \rightarrow \infty$.

Observe that so far no limitation have been set on the probabilistic properties of \mathbf{A} and \mathbf{B} . The distributional assumptions have been very general. For the design of $V^{(n)}$ the probabilistic properties of \mathbf{A} and \mathbf{B} are of central importance and here they come into play. This paper considers the setting where the innovations B are most likely responsible for extreme values of the solution to the stochastic recurrence equation. We make the following assumptions.

1. The generic random variables A and B are nonnegative.
2. The generic random variable B has a regularly varying tail, with index $-\alpha < 0$. Formally,

$$\lim_{t \rightarrow \infty} \frac{\mathbf{P}(B > xt)}{\mathbf{P}(B > t)} = x^{-\alpha}, \text{ for all } x > 0.$$

3. The Breiman condition holds for the generic random variable A . That is, there exists $\epsilon > 0$ such that

$$\mathbf{E}[A^{\alpha+\epsilon}] < \infty.$$

Under the assumptions (1)-(3) it is possible to derive the asymptotic decay of $p^{(n)}$. Indeed, it follows from the representation (5.1) as a weighted random walk and Theorem 3.1 in [31] that

$$\frac{\mathbf{P}(X_m > c_n)}{\mathbf{P}(B > c_n)} \rightarrow \sum_{k=0}^{m-1} E[A^\alpha]^k.$$

Now consider the choice of $V^{(n)}$. Let $V^{(n)}$ be defined as the probability distribution

$$V^{(n)}(\cdot) = \mathbf{P}((\mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \in \cdot \mid (\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in R^{(n)}),$$

with

$$R^{(n)} = \{A_k > a, \text{ for all } k = 1, \dots, m-1\} \cap \{\exists! j : a^{m-j} B_j > c_n\}.$$

The probability of this conditioning event can be computed explicitly as

$$\begin{aligned} r^{(n)} &= \mathbf{P}(\{A_k > a, \text{ for all } k = 1, \dots, m-1\} \cap \{\exists! j : a^{m-j} B_j > c_n\}) \\ &= \mathbf{P}(A > a)^{m-1} \\ &\quad \times \left(\mathbf{P}(B_m > x) \mathbf{P}(B_{m-1} < x/a) \cdots \mathbf{P}(B_1 < x/a^{m-1}) \right. \\ &\quad \left. + \mathbf{P}(B_m < x) \mathbf{P}(B_{m-1} > x/a) \mathbf{P}(B_{m-2} < x/a^2) \cdots \mathbf{P}(B_1 < x/a^{m-1}) \right. \\ &\quad \left. + \cdots + \mathbf{P}(B_m < x) \cdots \mathbf{P}(B_2 < x/a^{m-2}) \mathbf{P}(B_1 > x/a^{m-1}) \right) \\ &= \bar{F}_A(a)^{m-1} \sum_{i=1}^m \bar{F}_B(x/a^{m-i}) \prod_{j=1, \neq i}^m F_B(x/a^{m-j}). \end{aligned}$$

From the regular variation property of the distribution of B , assumption (2), it follows that

$$r^{(n)} \sim \bar{F}_A(a)^{m-1} \bar{F}_B(c_n) \{1 + a^\alpha + (a^\alpha)^2 + \dots + (a^\alpha)^{m-1}\} \quad \text{as } n \rightarrow \infty.$$

A convenient choice of the level $a = a_n$ is such that $r^{(n)}/p^{(n)} \rightarrow 1$, as $n \rightarrow \infty$. That is, a may be chosen as the solution to

$$\bar{F}_A(a)^{m-1} \sum_{k=0}^{m-1} a^{k\alpha} = \sum_{k=0}^{m-1} E[A^\alpha]^k.$$

The distribution $V^{(n)}$ has a known density with respect to $F(\cdot) = \mathbf{P}((\mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \in \cdot)$ given by

$$\frac{dV^{(n)}}{dF(\cdot)}(\mathbf{a}, \mathbf{b}) = \frac{1}{r^{(n)}} I\{(\mathbf{a}, \mathbf{b}) \in R^m\}.$$

Thus the MCMC estimator $\hat{q}_T^{(n)}$ of $1/p^{(n)}$ is given by

$$\hat{q}_T^{(n)} = \frac{1}{r^{(n)}} \frac{1}{T} \sum_{t=0}^{T-1} I\{(\mathbf{A}_t^{(m)}, \mathbf{B}_t^{(m)}) \in R^{(n)}\}, \quad (5.5)$$

where $(\mathbf{A}_t, \mathbf{B}_t)_{t=0}^{T-1}$ is generated via Algorithm 5.1. Observe that the estimator first factor of the estimator $\hat{q}_T^{(n)}$ may be interpreted as the asymptotic approximation $1/r^{(n)}$ multiplied by a stochastic correction factor.

Theorem 5.5. *The estimator $\hat{q}_T^{(n)}$ given by 5.5 has vanishing normalized variance for estimating $1/p^{(n)}$,*

$$\lim_{n \rightarrow \infty} (p^{(n)})^2 \text{Var}_{F_{c_n}}(\hat{q}_T^{(n)}) \rightarrow 0.$$

Proof. With $u^{(n)}(\mathbf{a}, \mathbf{b}) = \frac{1}{r^{(n)}} I\{(\mathbf{a}, \mathbf{b}) \in R\}$ it follows from assumptions 1-3 made above that

$$\begin{aligned} & (p^{(n)})^2 \text{Var}_{F_{c_n}^{(m)}} \left(\frac{1}{r^{(n)}} I\{(\mathbf{a}, \mathbf{b}) \in R\} \right) \\ &= \frac{(p^{(n)})^2}{(r^{(n)})^2} \text{Var}_{F_{c_n}^{(m)}} (I\{(\mathbf{a}, \mathbf{b}) \in R\}) \\ &= \frac{(p^{(n)})^2}{(r^{(n)})^2} \mathbf{P}(I\{(\mathbf{a}, \mathbf{b}) \in R\} \mid X_m > c_n) \mathbf{P}(I\{(\mathbf{a}, \mathbf{b}) \notin R\} \mid X_m > c_n) \\ &= \frac{p^{(n)}}{r^{(n)}} \mathbf{P}\left(1 - \frac{r^{(n)}}{p^{(n)}}\right) \rightarrow 0. \end{aligned}$$

This completes the proof. □

5.3 Numerical experiments

Theorem 5.5 of this paper proves that $\hat{q}_T^{(n)}$ is an efficient estimator of $1/p^{(n)}$. Most existing algorithms however design an efficient estimator $\hat{p}_T^{(n)}$ of $p^{(n)}$, so for comparison reasons the numerical experiments are based on $(\hat{q}_T^{(n)})^{-1}$.

By construction each simulation run of the MCMC algorithm only generates a single random variable (one simulation step) while both importance sampling and standard Monte Carlo generate $2m$ number of random variables ($2m$ simulation steps). Therefore the number of runs for the MCMC is scaled up by a factor of $2m$ so that all of the algorithms (MCMC, Monte Carlo and importance sampling) generate essentially the same number of random numbers. Thus getting a fairer comparison of the computer runtime between the three approaches.

Consider estimating $\mathbf{P}(X_n > c_n)$ where X_n is a solution to the recurrence equation $X_n = A_n X_{n-1} + B_n$ with $X_0 = 0$. The innovation B is a Pareto distributed variable with index α while the A is exponentially distributed with intensity λ . Each estimate is calculated using b number of batches, each consisting of T simulations in the case of importance sampling and standard Monte Carlo and $2nT$ in the case of MCMC. The results are presented in Table 3.

Table 3: The table displays the batch mean and standard deviation of the estimates of $\mathbf{P}(X_n > c)$ as well as the average runtime per batch for time comparison. The number of batches run is b , each consisting of T simulations for importance sampling (IS) and standard Monte Carlo (MC) and $T 2n$ simulations for Markov chain Monte Carlo (MCMC).

$b = 25, T = 10^5, n = 4, c = 10, \alpha = 2, \lambda = 3$			
	MCMC	IS	MC
Avg. est.	1.233e-2	1.223e-2	1.221e-2
Std. dev.	43e-5	9e-5	43e-5
Avg. time per batch(s)	35	36	2
$b = 25, T = 10^5, n = 4, c = 10^2, \alpha = 2, \lambda = 3$			
	MCMC	IS	MC
Avg. est.	1.298e-4	1.278e-4	1.360e-4
Std. dev.	7e-6	1e-6	35e-6
$b = 25, T = 10^5, n = 4, c = 10^3, \alpha = 2, \lambda = 3$			
	MCMC	IS	MC
Avg. est.	1.149e-6	1.284e-6	2.000e-6
Std. dev.	36e-8	7e-8	408e-8
$b = 25, T = 10^5, n = 5, c = 2, \alpha = 5, \lambda = 3$			

6 Ruin probability in an Insurance Model with Risky Investments

In this section the Markov chain Monte Carlo approach to rare-event simulation is applied to compute the ruin probability in an insurance model with risky investments.

The ruin problem with investment is reasonably well studied. A recent overview is given by Paulsen [46]. In the infinite horizon setting there are two asymptotic regimes. Power tail asymptotics can arise either as the cumulative effect of negative returns on the investment asset or because of power tails of the claim size distribution. In the first case the power tail asymptotics can be derived by expressing the risk reserve as the solution to a stochastic recurrence equation whose stationary solution has a power tail. See e.g. [45, 36, 20, 47, 38]. In the second case the power asymptotics of the ruin probability is more directly inferred from the power tail of the claim size distribution, see [21, 50, 38].

The following model, in discrete time, for the risk reserve of an insurance company is considered here. Denote by B_k the net loss, claims minus premiums, over the k th period. Suppose the insurance company invests the risk reserve in a risky asset and denote by R_k the stochastic return on the risky asset over the k th period. It is assumed that $\{B_k\}$ and $\{R_k\}$ are independent sequences, each consisting of independent and identically distributed random variables. The risk reserve U_k at the end of the k th period is modeled as

$$\begin{aligned} U_k &= R_k(Z_{k-1} - B_k), \quad \text{for } k \geq 1, \\ U_0 &= u. \end{aligned}$$

Iterating the relation above yields

$$U_n = R_n \cdots R_1 u - (R_n \cdots R_1 B_1 + R_n \cdots R_2 B_2 + \cdots + R_n B_n).$$

Assume that $R_k > 0$ a.s. for all k and put $A_k = 1/R_k$. The last display is equivalent to

$$A_1 \cdots A_n U_n = u - W_n,$$

where

$$W_n = B_1 + A_1 B_2 + \cdots + A_1 \cdots A_{n-1} B_n.$$

Observe that W_n represents the discounted losses that have accumulated up until time n . The event of ruin up until time n is equivalent to

$$\left\{ \inf_{0 \leq k \leq n} U_k < 0 \right\} = \left\{ \sup_{0 \leq k \leq n} W_k > u \right\}.$$

Our objective is to construct an efficient algorithm to compute the ruin probability

$$p^{(n)} = \mathbf{P} \left(\sup_{0 \leq k \leq n} W_k > u_n \right).$$

As in the previous section we denote by $\mathbf{A}^{(n)} = (A_1, \dots, A_{n-1})$ and $\mathbf{B}^{(n)} = (B_1, \dots, B_n)$. To compute $p^{(n)}$ with the MCMC approach a Gibbs sampler is proposed with the conditional distribution

$$F_{u_n}^n(\cdot) = \mathbf{P}((\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in \cdot \mid \sup_{0 \leq k \leq n} W_k > u_n).$$

6.1 A Gibbs sampler for computing the ruin probability

The Gibbs sampler is constructed similarly as in Section 5 with the difference that the conditioning event is $\{\sup_{0 \leq k \leq n} W_k > u_n\}$ instead of $\{X_m > c_n\}$.

Algorithm 6.1. Start with initial state $(\mathbf{A}_0^{(n)}, \mathbf{B}_0^{(n)}) = (A_{0,1}, \dots, A_{0,n}, B_{0,1}, \dots, B_{0,n})$ where $X_0^{(n)} > u_n$. Given $(\mathbf{A}_t^{(n)}, \mathbf{B}_t^{(n)})$, for some $t = 0, 1, \dots$, the next state $(\mathbf{A}_{t+1}^{(n)}, \mathbf{B}_{t+1}^{(n)})$ is sampled as follows:

1. Draw a randomized ordering j_1, \dots, j_{2n} of $\{1, \dots, 2n\}$ and proceed updating $(\mathbf{A}_t^{(n)}, \mathbf{B}_t^{(n)})$ in the order thus obtained.
2. For $m = 1, \dots, 2n$, set $k = j_m$ and do the following:
 - i. If $k \in \{1, \dots, n\}$ then $A_{t,k}$ is to be updated. Sample A' from the conditional distribution

$$\mathbf{P}(A' \in \cdot \mid A' > s),$$

where

$$s = \min_{1 \leq k \leq n} \left\{ \frac{u_n - \sum_{i=1}^k B_{t,i} \prod_{j=1}^{k-1} A_{t,j}}{\sum_{i=k+1}^n B_{t,i} \prod_{j=1, \neq k}^{i-1} A_{t,j}} \right\}.$$

Put $\mathbf{A}_{t+1}^{(n)} = (A_{t,1}, \dots, A_{t,k-1}, A', A_{t,k+1}, \dots, A_{t,n})$ and $\mathbf{B}_{t+1}^{(n)} = \mathbf{B}_t^{(n)}$.

- ii. If $k \in \{n+1, \dots, 2n\}$ then $B_{t,(k-n)}$ is to be updated. Sample B' from the conditional distribution

$$\mathbf{P}(B' \in \cdot \mid B' > s),$$

where

$$s = \min_{1 \leq (k-n) \leq n} \left\{ \frac{u_n - \sum_{i=1, \neq (k-n)}^n B_{t,i} \prod_{j=1}^{i-1} A_{t,j}}{\prod_{j=1}^{(k-n)-1} A_{t,j}} \right\}.$$

Put $\mathbf{A}_{t+1}^{(n)} = \mathbf{A}_t^{(n)}$ and $\mathbf{B}_{t+1}^{(n)} = (B_{t,1}, \dots, B_{t,(k-n)-1}, B', B_{t,(k-n)+1}, \dots, B_{t,n})$.

Iterate steps 1 and 2 until the entire Markov chain $(\mathbf{A}_t^{(n)}, \mathbf{B}_t^{(n)})_{t=0}^{T-1}$ is constructed.

Proposition 6.2. *The Markov chain $(\mathbf{A}_t^{(n)}, \mathbf{B}_t^{(n)})_{t \geq 0}$ generated by Algorithm 6.1, has the conditional distribution $F_{u_n}^{(n)}$ as its invariant distribution and is uniformly ergodic.*

The proof of the above result is essentially identical to the proofs of Proposition 5.2 and 5.3 is therefore omitted.

6.2 Constructing an efficient estimator of the reciprocal ruin probability

As mentioned in Section 2 a good candidate for $V^{(n)}$ is a probability distribution

$$\mathbf{P}((\mathbf{A}, \mathbf{B}) \in \cdot \mid (\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in R^{(n)}),$$

where $r^{(n)} = \mathbf{P}((\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in R^{(n)})$ is asymptotically close to $p^{(n)}$ in the sense that $r^{(n)}/p^{(n)} \rightarrow 1$ as $n \rightarrow \infty$.

Observe that, apart from the independence assumptions, the distributional assumptions on B and R have been completely general. For the design of $V^{(n)}$ the probabilistic properties of B and R are of central importance and here they come into play. This paper considers the setting where large claims are most likely responsible for ruin. We make the following assumptions.

1. The distribution of B has a regularly varying right tail, with index $-\alpha < 0$:

$$\lim_{t \rightarrow \infty} \frac{\mathbf{P}(B > xt)}{\mathbf{P}(B > t)} = x^{-\alpha}, \text{ for all } x > 0.$$

2. The stochastic returns R are almost surely strictly positive. In addition, there exists $\epsilon > 0$ such that $\mathbf{E}[R^{-\alpha-\epsilon}] < 1$.

Under the assumptions (1)-(2) it is possible to derive the asymptotic decay of $p^{(n)}$. Note first that (2) translates into the conditions that the generic random variables A is strictly positive and $\mathbf{E}[A^{\alpha+\epsilon}] < 1$ for some $\epsilon > 0$. It follows from the representation of W_n as a weighted random walk and by combining Example 2.2 and Corollary 5.1 in [32] that

$$\lim_{n \rightarrow \infty} \frac{\mathbf{P}(\sup_{0 \leq k \leq n} W_k > u_n)}{n \mathbf{P}(B > u_n)} = \mathbf{E} \left[\left(\sup_{k \geq 1} \prod_{j=1}^k A_j \right)^\alpha \right].$$

Now consider the choice of $V^{(n)}$. Let $V^{(n)}$ be defined as the probability distribution

$$V^{(n)}(\cdot) = \mathbf{P}((\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in \cdot \mid (\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in R^{(n)}),$$

with

$$R^{(n)} = \{A_k > a_n, \text{ for all } k = 1, \dots, n-1\} \cap \{\exists! j : a_n^{n-j} B_j > u_n\}.$$

The probability of this conditioning event can be computed explicitly as

$$\begin{aligned} r^{(n)} &= \mathbf{P}(\{A_k > a_n, \text{ for all } k = 1, \dots, n-1\} \cap \{\exists! j : a_n^{n-j} B_j > u_n\}) \\ &= \mathbf{P}(A > a_n)^{n-1} \left(\mathbf{P}(B_n > u_n) \mathbf{P}(B_{n-1} < u_n/a_n) \cdots \mathbf{P}(B_1 < u_n/a_n^{n-1}) \right. \\ &\quad + \mathbf{P}(B_n < u_n) \mathbf{P}(B_{n-1} > u_n/a_n) \mathbf{P}(B_{n-2} < u_n/a_n^2) \cdots \mathbf{P}(B_1 < u_n/a_n^{n-1}) \\ &\quad \left. + \cdots + \mathbf{P}(B_n < u_n) \cdots \mathbf{P}(B_2 < u_n/a_n^{n-2}) \mathbf{P}(B_1 > u_n/a_n^{n-1}) \right) \\ &= \bar{F}_A(a_n)^{n-1} \sum_{i=1}^n \bar{F}_B(u_n/a_n^{n-i}) \prod_{j=1, \neq i}^n F_B(u_n/a_n^{n-j}). \end{aligned}$$

From the regular variation property of the distribution of B , assumption (2), it follows that if $u_n/a_n^{n-1} \rightarrow \infty$, then

$$r^{(n)} \sim \bar{F}_A(a_n)^{n-1} \bar{F}_B(u_n) \{1 + a_n^\alpha + (a_n^\alpha)^2 + \cdots + (a_n^\alpha)^{n-1}\} \quad \text{as } n \rightarrow \infty.$$

A convenient choice of the level a_n is such that $r^{(n)}/p^{(n)} \rightarrow 1$, as $n \rightarrow \infty$. That is, a_n may be chosen as the solution to

$$\bar{F}_A(a_n)^{n-1} \sum_{k=0}^{n-1} a_n^{k\alpha} = n\mathbf{E}\left[\left(\sup_{k \geq 1} \prod_{j=1}^k A_j\right)^\alpha\right].$$

The distribution $V^{(n)}$ has a known density with respect to $F(\cdot) = \mathbf{P}((\mathbf{A}^{(n)}, \mathbf{B}^{(n)}) \in \cdot)$ given by

$$\frac{dV^{(n)}}{dF(\cdot)}(\mathbf{a}, \mathbf{b}) = \frac{1}{r^{(n)}} I\{(\mathbf{a}, \mathbf{b}) \in R^{(n)}\}.$$

Thus the MCMC estimator $\hat{q}_T^{(n)}$ of $1/p^{(n)}$ is given by

$$\hat{q}_T^{(n)} = \frac{1}{r^{(n)}} \frac{1}{T} \sum_{t=0}^{T-1} I\{(\mathbf{A}_t^{(n)}, \mathbf{B}_t^{(n)}) \in R^{(n)}\}, \quad (6.1)$$

where $(\mathbf{A}_t, \mathbf{B}_t)_{t=0}^{T-1}$ is generated via Algorithm 6.1. Observe that the estimator first factor of the estimator $\hat{q}_T^{(n)}$ may be interpreted as the asymptotic approximation $1/r^{(n)}$ multiplied by a stochastic correction factor.

Theorem 6.3. *The estimator $\hat{q}_T^{(n)}$ given by (6.1) has vanishing normalized variance for estimating $1/p^{(n)}$,*

$$\lim_{n \rightarrow \infty} (p^{(n)})^2 \text{Var}_{F_{a_n}^{(n)}}(\hat{q}_T^{(n)}) \rightarrow 0.$$

References

- [1] S. Asmussen. *Applied Probability and Queues*, volume 51 of *Stochastic Modelling and Applied Probability*. Springer, New York, 2003.
- [2] S. Asmussen and K. Binswanger. Simulation of ruin probabilities for subexponential claims. *Astin Bulletin*, 27(2):297–318, 1997.
- [3] S. Asmussen and P. W. Glynn. *Stochastic Simulation*, volume 57 of *Stochastic Modelling and Applied Probability*. Springer, New York, 2007.
- [4] S. Asmussen and D. P. Kroese. Improved algorithms for rare event simulation with heavy tails. *Adv. Appl. Probab.*, 38:545–558, 2006.
- [5] J. Blanchet and P. W. Glynn. Efficient rare-event simulation for the maximum of a heavy-tailed random walk. *Ann. of Appl. Prob.*, 18(4):1351–1378, 2008.
- [6] J. Blanchet and P.W. Glynn. Efficient rare event simulation for the single server queue with heavy tailed distributions. *Ann. Appl. Prob.*, 18(4):1351–1378, 2008.
- [7] J. Blanchet, P.W. Glynn, and J. Liu. Fluid heuristics, lyapunov bounds and efficient importance sampling for a heavy-tailed g/g/1 queue. *Queueing Syst.*, 56(3):99–113, 2007.

- [8] J. Blanchet and C. Li. Efficient rare-event simulation for heavy-tailed compound sums. *ACM T. Model Comput. S.*, 21:1–10, 2011.
- [9] J. Blanchet and C. Li. Efficient rare-event simulation for heavy-tailed compound sums. *ACM T. Model Comput. S.*, 21:1–10, 2011.
- [10] J. Blanchet and J. Liu. State-dependent importance sampling for regularly varying random walks. *Adv. Appl. Probab.*, 40:1104–1128, 2008.
- [11] J. Blanchet and J. C. Liu. State-dependent importance sampling for regularly varying random walks. *Adv. Appl. Probab.*, 40:1104–1128, 2008.
- [12] K. J. Chan. Asymptotic behavior of the gibbs sampler. *J. Am. Stat. Assoc.*, 88:320–326, 1993.
- [13] D. B. H. Cline and T. Hsing. Large deviation probabilities for sums of random variables with heavy or subexponential tails. Technical report, Texas A&M University, 1994.
- [14] P. Del Moral and J. Garnier. Genealogical particle analysis of rare events. *Ann. Appl. Probab.*, 15(4):2496–2534, 2005.
- [15] D. Denisov, A. B. Dieker, and V. Shneer. Large deviations for random walks under subexponentiality: The big-jump domain. *Ann. Probab.*, 36:1946–1991, 2008.
- [16] P. Dupuis, K. Leder, and H. Wang. Importance sampling for sums of random variables with regularly varying tails. *ACM T. Model Comput. S.*, 17(3), 2007.
- [17] P. Dupuis, A. Sezer, and H. Wang. Dynamic importance sampling for queueing networks. *Ann. Appl. Probab.*, 17(4):1306–1346, 2007.
- [18] P. Dupuis and H. Wang. Importance sampling, large deviations, and differential games. *Stoch. Stoch. Rep.*, 76(6):481–508, 2004.
- [19] P. Dupuis and H. Wang. Subsolutions of an isaacs equation and efficient schemes for importance sampling. *Math. Oper. Res.*, 32(3):1–35, 2007.
- [20] A. Frolova, Y. Kabanov, and S. Pergaminshchikov. In the insurance business risky investments are dangerous. *Finance Stoch.*, 6:227–235, 2002.
- [21] J. Gaier and P. Grandits. Ruin probabilities in the presence of regularly varying tails and optimal investment. *Insur. Math. Econ.*, 30:211–217, 2002.
- [22] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741, 1984.
- [23] W. R. Gilks, S. Richardsson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall, 1996.
- [24] P. Glasserman and Y. Wang. Counterexamples in importance sampling for large deviations probabilities. *Ann. Appl. Probab.*, 7(3):731–746, 1995.

- [25] T. Gudmundsson and H. Hult. Markov chain monte carlo for computing rare-event probabilities for a heavy-tailed random walk. *Journal of Applied Probability*, 51(2), June 2014.
- [26] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [27] H. Hult, Blanchet. J., and K. Leder. Rare-event simulation for stochastic recurrence equations with heavy-tailed innovations. *ACM T. Model. Comput. S.*, 2012.
- [28] H. Hult and F. Lindskog. Extremal behavior of regularly varying stochastic processes. *Stoch. Proc. Appl.*, 115(2):249–274, 2005.
- [29] H. Hult, F. Lindskog, O. Hammarlid, and C.J. Rehn. *Risk and Portfolio Analysis*. Springer, New York, 2012.
- [30] H. Hult, F. Lindskog, T. Mikosch, and G. Samorodnitsky. Functional large deviations for multivariate regularly varying random walks. *Ann. Appl. Probab.*, 14(4):2651–2680, 2005.
- [31] H. Hult and G. Samorodnitsky. Tail probabilities for infinite series of regularly varying random vectors. *Bernoulli*, 14:838–864, 2008.
- [32] H. Hult and G. Samorodnitsky. Large deviations for point processes based on stationary sequences with heavy tails. *J. Appl. Probab.*, 47:1–40, 2010.
- [33] H. Hult and J. Svensson. On importance sampling with mixtures for random walks with heavy tails. Working paper, Royal Institute of Technology, October 2010.
- [34] G. L. Jones. On the markov chain central limit theorem. *Probability Surveys*, 1:299–320, 2004.
- [35] S. Juneja and P. Shahabuddin. Simulation heavy tailed processes using delayed hazard rate twisting. *ACM T. Model Comput. S.*, 12(2):94–118, April 2002.
- [36] V. Kalashnikov and R. Norberg. Power tailed ruin probabilities in the presence of risky investments. *Stoch. Process. Appl.*, 98:211–228, 2002.
- [37] C. Klüppelberg and T. Mikosch. Large deviations for heavy-tailed random sums with applications to insurance and finance. *J. Appl. Probab.*, 37(2):293–308, 1 1997.
- [38] C. Klüppelberg and R. Kostadinova. Integrated risk models with exponential lévy investment. *Insur. Math. Econ.*, 42:560–577, 2008.
- [39] P. L’Ecuyer and C. Landim. Splitting for rare event simulation. *Proc. 2006 Winter Simulation Conf.*, pages 137–148, 2006.
- [40] K. L. Mengersen and R. L. Tweedie. Rates of convergence of the hastings and metropolis algorithms. *Ann. Stat.*, 24:101–121, 1996.

- [41] N. Metropolis, A.W. Rosebluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1953.
- [42] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer, New York, 1993.
- [43] T. Mikosch. *Non-life Insurance Mathematics*. Springer, New York, 2009.
- [44] E. Nummelin. *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge University Press, 1984.
- [45] H. Nyrhinen. On the ruin probabilities in a general economic environment. *Stoch. Process. Appl.*, 83:319–330, 1999.
- [46] J. Paulsen. Ruin models with investment income. *Probab. Surv.*, 5:416–434, 2008.
- [47] S. Pergamenschikov and O. Zeitouni. Ruin probability in the presence of risky investments. *Stoch. Process. Appl.*, 116:267–278, 2006.
- [48] B. D. Ripley. *Stochastic Simulation*. Wiley Series in Probability and Statistics, New Jersey, 1987.
- [49] J. S. Rosenthal. Minorization conditions and convergence rates for markov chain monte carlo. *J. Am. Stat. Assoc.*, 90(430):558–566, June 1995.
- [50] H. Schmidli. On optimal investment and sub exponential claims. *Insur. Math. Econ.*, 36:25–35, 2005.
- [51] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: a sampling-resampling perspective. *Am. Stat.*, 46:84–88, 1992.
- [52] L. Tierney. Markov chains for exploring posterior distributions (with discussion). *Ann. Stat.*, 22:1701–1762, 1994.