THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Advances in Monte Carlo Methodology

Robert Salomone

B.Sc. (Hons)

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2018

School of Mathematics and Physics

Abstract

This thesis is comprised of two parts. The first presents several advances in Sequential Monte Carlo methods. A new class of sequential Monte Carlo methods called *Nested Sampling via Sequential Monte Carlo* (NS–SMC), which reframes the Nested Sampling method of Skilling in terms of sequential Monte Carlo techniques is introduced. In contrast to NS, the analysis of NS-SMC does not require the (unrealistic) assumption that the simulated samples be independent. This new framework allows one to obtain provably consistent and unbiased estimates of marginal likelihoods when Markov chain Monte Carlo (MCMC) is used to produce new samples. As the original NS algorithm is a special case of NS–SMC, this provides insights as to why NS seems to produce accurate estimates despite a typical violation of its assumptions. Novel calibration methods that apply generally to SMC Samplers are introduced, and applied in a numerical study where the performance of NS–SMC and temperature–annealed SMC is compared on several challenging and realistic statistical problems.

The second part of the dissertation presents several novel Monte Carlo methods for the estimation of distributional quantities relating sums of random variables. For the sum of dependent log–normal random variables, novel estimators for the left tail (cumulative distribution function), the right tail (or complementary distribution function), and the probability density function are introduced. Numerical experiments demonstrate that in all three settings, our proposed methodology delivers accurate estimators in settings for which existing methods have large variance and tend to underestimate the quantity of interest. Theoretical efficiency results are presented for the left and right tail estimators, and a method for efficiently sampling dependent log–normal random variables conditional on a left tail rare event exactly is also presented. Finally, a novel estimator for estimating the probability density function of a sum of random variables in a more general setting is studied, which allows estimation of marginal probability density functions in the context of approximate sampling with MCMC.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my higher degree by research candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Publications included in this thesis

No publications included.

Submitted manuscripts included in this thesis

1. **Salomone, R.**, South, L.F., Drovandi, C.C. and Kroese, D.P. (2018) — Unbiased and Consistent Nested Sampling via Sequential Monte Carlo. Submitted to *Journal of the American Statistical Association* on 13th August 2018.

2. Botev, Z.I., **Salomone, R.**, and Mackinlay, D. (2017). Fast and Accurate Computation of the Distribution of Sums of Dependent Log-Normals. Submitted to *Annals of Operations Research* on 6th June 2017, with subsequent revision submitted on 24th May 2018.

3. Laub, P.J., **Salomone, R.**, and Botev, Z.I. (2017). Monte Carlo Estimation of the Density of the Sum of Dependent Random Variables. Submitted to *Mathematics and Computers in Simulation* on 30th November 2017, with subsequent revision submitted on 30th June 2018.

Other publications during candidature

1. **Salomone, R.**, Vaisman, R., and Kroese, D.P. (2016). Estimating the Number of Vertices in Convex Polytopes. *Proceedings of the Annual International Conference on Operations Research and Statistics*, ORS 2016.

Contributions by others to the thesis

My supervisor, Professor Dirk P. Kroese helped me edit this thesis.

Statement of parts of the thesis submitted to qualify for the award of another degree

No works submitted towards another degree have been included in this thesis.

Research involving human or animal subjects

No animal or human subjects were involved in this research.

Acknowledgments

First and foremost, I thank my advisor, Professor Dirk P. Kroese, for his patience, support, and encouragement over all the time that we have worked together. He has been an exceptional advisor and I am fortunate to have been his student.

I am grateful to my collaborators: Zdravko Botev, Christopher Drovandi, Leah South, Slava Vaisman, and Patrick Laub. In particular, I would like to extend a special thanks to Zdravko Botev who has simultaneouly been an excellent collaborator, colleague, and mentor over the duration of my candiditure. I look back on my visits to the University of New South Wales to work together with great fondness, and hope the opportunity to do more work together will arise in future.

During the time of my candiditure, my experience was greatly enriched as a result of being a student member of the Australian Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS). It is through the centre that I met many people with similar research interests, and I am grateful for this opportunity. I was also fortunate to receive financial support in the form of an ACEMS top–up scholarship.

I thank Robert Kohn and Matias Quiroz for their invitation to visit their research group, and for their subsequent hospitality. I thoroughly enjoyed my visit and I look forward to the possibility of future collaboration.

A special thanks goes to my friends and colleagues at the University of Queensland who made my time there all the more enjoyable. In particular, I thank Liam Hodgkinson, Chris van der Heide, Pat Laub, and Fred Roosta–Khorasani for our many interesting discussions of both the scholarly and jovial kind.

Last, but certainly not least, I thank my parents for their support through all my studies (and life!) as well as my partner Jessie–Josephine, who has been exceptionally encouraging and supportive throughout my entire candiditure.

Robert Salomone
Brisbane, 2018

Financial support

This research was supported by an Australian Government Research Training Program Scholarship. Further financial assistance was provided by the Australian Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) under grant number CE140100049.

Keywords

Monte Carlo methods, Nested Sampling, sequential Monte Carlo, sums of lognormal random variables, density estimation

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 010404 Probability Theory, 40%
ANZSRC code: 010405 Statistical Theory, 40%
ANZSRC code: 010406 Stochastic Analysis and Modelling, 20%

Fields of Research (FoR) Classification

FoR code: 0104 Statistics 100%

Contents

List of figures

List of tables

List of abbreviations

Abbreviations	
AK	Asmussen–Kroese
ANS–SMC	Adaptive Nested Sampling via Sequential Monte Carlo
a.s.	almost surely
cdf	Cumulative distribution function
ESS	Effective sample size
FA	Factor analysis
GT	Gulisashvili and Tankov
iid	Independent and identically distributed
INS	Improved Nested Sampling
KKT	Karush–Kuhn–Tucker
LPM	Last Particle Method
MAK	Modified Asmussen–Kroese
MALA	Metropolis Adjusted Langevin Algorithm
MCMC	Markov Chain Monte Carlo
NS	Nested Sampling
NS–SMC	Nested Sampling via Sequential Monte Carlo
ODE	Ordinary differential equation
pdf	Probability density function
RE	Relative error
rv	Random variable
SLN	Sum of log–normals
SMC	Sequential Monte Carlo
TA–SMC	Temperature–Annealed Sequential Monte Carlo
WNRV	Work normalized relative variance
WNV	Work normalized variance

Chapter 1

Introduction

The evaluation of complicated integrals is a problem that is ubiquitous in the sciences, engineering, finance, and statistics. Often, exact evaluation of quantities of interest is not possible, so approximate methods must be used. A powerful and nowadays widely used approach in that of the *Monte Carlo* method. Here, difficult to compute numerical quantities, such as high-dimensional integrals, are estimated via the outcomes of random computational experiments. The aim of this thesis is the development of novel Monte Carlo methodology with improved theoretical properties and/or practical performance for problems of interest, as well as obtaining a better understanding of existing methods.

The first part of this thesis is concerned with Sequential Monte Carlo (SMC) methodology. SMC is a general class of methods that, broadly speaking, uses an interacting population of samples (called particles) to approximate a sequence of distributions and estimate their normalizing constants. Certain features of SMC are shared with a method called Nested Sampling (NS), that latter of which being a widely used tool in computational physics and amongst practitioners of Bayesian statistics in astronomy. However, NS lacks convergence results in the common practical setting where Markov Chain Monte Carlo is used to produce (dependent) samples. The primary contribution of the first part of this thesis is a new perspective through which to apply the Nested Sampling approach that resolves this issue and several others.

Chapter 2 introduces a novel class of Sequential Monte Carlo methods called *Nested Sampling via Sequential Monte Carlo* (NS–SMC), which reframes NS entirely in terms of Sequential Monte Carlo techniques. As a result, several new Sequential Monte Carlo methods are presented, including a variant

for which convergence results in the context of Markov Chain Monte Carlo samples are established. Moreover, the proposed methodology also allows for the unbiased estimation of normalizing constants, and provides an entirely new avenue through which to approach Nested Sampling based research. It is also demonstrated through the NS–SMC framework how an improved version of NS can be derived without requiring that the simulated samples be independent. As a result, insight is provided as to why the original Nested Sampling method seems to perform well in settings where there is a large amount of dependency in the simulated samples. The secondary contribution of Chapter 2 is the introduction of novel calibration methods for SMC samplers, and their subsequent use in a numerical study examining how NS–SMC and the popular temperature–annealed SMC approach compare when applied to challenging and realistic problems in Bayesian statistics.

In the second part of this dissertation, the area of study is Monte Carlo methods for estimating distributional quantities relating sums of random variables. Chapter 3 considers specifically sums of dependent log-normals; which are of interest in finance, risk management, and wireless communications. Novel methodology is proposed for many quantities of interest. Specifically, estimators for (i) the cumulative distribution function, (ii) the probability density function, and (iii) the complementary cumulative distribution function, are introduced. While, in cases (i) and (iii), the proposed estimators are shown to be theoretically efficient, the major contribution of the chapter is that the proposed methodology also exhibits excellent practical performance, particularly in cases where existing methods (some with *stronger* theoretical properties) perform poorly and, in the case of estimators for the complimentary cumulative distribution function, tend to underestimate the quantity of interest. An additional contribution of Chapter 3 is the introduction of the first method to efficiently generate *exact* samples of lognormal factors conditional on their sum being less than a quantity that is sufficiently small to make the event rare.

Finally, Chapter 4 considers probability density function estimation for the sum of more *general* dependent summands. Here, an unbiased estimator based on sensitivity analysis is proposed. We conduct a short numerical study that demonstrates it performs favourably in terms of variance when compared to other unbiased estimators, and examine its extensions to marginal density estimation in the context of Markov Chain Monte Carlo.

Part I

Advances in Sequential Monte Carlo Methods

Chapter 2: Authorship Statement

Citation

Salomone, R., South, L.F., Drovandi, C.C., and Kroese, D.P. (2018), Unbiased and Consistent Nested Sampling via Sequential Monte Carlo. Submitted to *Journal of the American Statistical Association* on 14th August, 2018.

The manuscript (with some minor additional editing) is included.

Contributions

The majority of the work for the paper was carried out by myself (55%) and L.F. South (35%) in different capacities. Specifically,

- I was responsible for the conception of the project, and the development of the main methodology (and associated theory) outlined in Sections 2.4 and 2.8.
- I was responsible for all of the writing, with the exception of Sections 2.5 and 2.6, which was carried out jointly with L.F. South.
- I was responsible for the phase transition example experiments in Section 2.4.3 and their interpretation.
- I assisted L.F. South (with additional input by C.C. Drovandi) to develop the calibration methodology in Section 2.5.
- L.F. South designed the experiments in Section 2.6, conducted the experiments in this section, and developed the associated MATLAB SMC library that is now available online. I interpreted the results of these experiments jointly with L.F. South.
- I contributed equally with L.F. South, C.C. Drovandi, and D.P. Kroese to the editing of the paper.

Chapter 2

Unbiased and Consistent Nested Sampling via Sequential Monte Carlo

2.1 Introduction

A canonical problem in the computational sciences is the estimation of integrals of the form

$$\pi(\varphi) = \mathbb{E}_\pi[\varphi(\boldsymbol{X})] = \int_E \varphi(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{2.1}$$

where π is a probability density on $E \subseteq \mathbb{R}^d$ and $\varphi : E \to \mathbb{R}$ is a π-integrable function. Note the "overloading" of notation for $\pi(\cdot)$, depending on whether the argument is a function φ or a vector $\boldsymbol{x}$. In Bayesian computation, which is the focus of this work, $\pi(\boldsymbol{x})$ is typically known only up to a normalizing constant, that is, $\pi(\boldsymbol{x}) = \gamma(\boldsymbol{x})/\mathcal{Z}$ for some known positive function γ which, in turn, decomposes into a product $\eta\,\mathcal{L}$, where η is another probability density function. In particular, in this setting, π is the *posterior* probability density, η is the *prior* probability density, $\mathcal{L}$ the *likelihood* function, and $\boldsymbol{x} \in E$ represents a *parameter*. For clarity, the correspondence to the usual Bayesian notation (which is typically written in terms of paramater $\boldsymbol{\theta}$ and data $\mathcal{D}$) is as follows:

$$\underbrace{p(\boldsymbol{\theta}\,|\,\mathcal{D})}_{\pi(\boldsymbol{x})} \propto \underbrace{p(\boldsymbol{\theta})}_{\eta(\boldsymbol{x})}\,\underbrace{p(\mathcal{D}\,|\,\boldsymbol{\theta})}_{\mathcal{L}(\boldsymbol{x})}. \tag{2.2}$$

Even though π, η and $\mathcal{L}$ are general functions, and do not have to be interpreted in terms of Bayesian computation, we henceforth refer to them as posterior, prior and likelihood function, respectively. Another quantity of interest is the normalizing constant

$$\mathcal{Z} = \int_E \eta(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{2.3}$$

which, in the Bayesian context, is called the *marginal likelihood* (or model evidence) and is often used in model selection.

The most popular methodology for estimating (2.1) is to use *Markov Chain Monte Carlo* (MCMC). Here, an ergodic Markov chain with π as its invariant density is simulated, yielding samples approximately from π after a suitably long duration known as the burn–in period. The empirical distribution of these samples can then be used to estimate (2.1). For more details, see [46, Chapters 6–12].

Nested Sampling (NS) [48] is a Monte Carlo/numerical quadrature method proposed initially for the estimation of marginal likelihoods, which also provides estimates of $\mathbb{E}_\pi[\varphi(\boldsymbol{X})]$ without requiring additional likelihood evaluations. The method is based on a sampling scheme that samples from progressively constrained (nested) versions of the prior. NS has achieved wide-spread acceptance as a tool for Bayesian computation in certain fields, being particularly popular in astronomy (see for example [50] and [51]) and more generally as a computational method in physics (examples here include [1] and [39]). However, NS has failed to achieve popularity more broadly in the statistical community, largely owing to a variety of theoretical problems, most notable of which is that the methodology assumes that one can obtain perfect and independent samples from constrained versions of the prior at each iteration, which is clearly unrealistic.

On the other hand, *Sequential Monte Carlo* (SMC) is a general methodology that involves using an interacting population of particles to approximate a sequence of distributions via a combination of mutation, correction, and selection steps. SMC has a rich theoretical basis, as it can be analyzed through interacting particle approximations to a flow of Feynman-Kac measures, see for example the technical monograph [15], or the tutorial [16]. The use of SMC methodology in a statistical setting began with the "Bootstrap Particle Filter" of [28] for online inference in hidden Markov models, and has been the topic of much research in the statistical community (see for example, the survey [20]). However, SMC methods in general date much further back to the *multilevel splitting* method of [32] for the estimation of rare–event probabilities. An overview of splitting techniques can be found in [47, Chapter 9], and such methods have continued to be active topic for research, see for example [4], [10], and [9].

The special case of SMC where all sampling distributions live on the same space E is discussed in [17]. In this setting, one can sample from an arbitrary density π by introducing an artificial sequence of densities bridging from an easy to sample distribution (say η) to π. This approach is often referred to as SMC in the *static* setting. While static SMC samplers often make use of MCMC moves, they possess advantages over the pure MCMC approach in that they are naturally parallelizable, can cope with complicated posterior landscapes such as those containing multimodality, and have the added benefit

of being able to produce consistent (and unbiased) estimates of the marginal likelihood as a byproduct.

The aim of this paper is to explore the connection between NS and SMC samplers, resolve some long–standing theoretical issues with NS by placing it in the SMC framework, and demonstrate not only how the resulting algorithm can be implemented effectively in practice, but also that it is able to obtain similar quality of results to existing SMC approaches under similar conditions on highly challenging examples.

To those ends, the contributions of this work are as follows:

1. We show that by implementing a special type of SMC sampler that takes two importance sampling paths at each iteration, one obtains an analogous SMC method to NS that resolves its main theoretical and practical issues. Most notably, the consistency of estimates of marginal likelihood and posterior inferences with our algorithm is easily established from the properties of SMC methods, and does not rely on obtaining perfect independent samples. Moreover, estimates of the marginal likelihood are unbiased.

2. We introduce an improved version of NS, of which the original NS method can be interpreted as a "rough" version. This gives insights as to why NS seems to work in practice when samples are dependent, despite the original formulation of the method requiring independent samples.

3. We provide recommendations on how to ensure robust performance of SMC samplers in practice, including how to tune MCMC kernels and determine an appropriate amount of MCMC repeats.

4. Using these techniques, we present the first extensive comparison between the popular temperature–annealed SMC approach and our NS–SMC approach, for both the purpose of marginal likelihood estimation and posterior sampling on difficult realistic statistical problems.

5. Having demonstrated that the ideas behind NS find their true home within SMC methodology, we conclude by discussing the variety of theoretical and methodological avenues of possible future research.

2.2 Nested Sampling

Nested Sampling (NS) [48] is based on the identity

$$\mathcal{Z} = \int_E \eta(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbb{E}_\eta[\mathcal{L}(\boldsymbol{X})] = \int_0^\infty \mathbb{P}(\mathcal{L}(\boldsymbol{X}) > l)\,\mathrm{d}\,l, \tag{2.4}$$

where $\mathcal{L}$ is a function mapping from some space E to $\mathbb{R}_+$, and $\boldsymbol{X} \sim \eta$. Note that $\mathbb{P}(\mathcal{L}(\boldsymbol{X}) > l)$ is simply the tail cdf (survival function) of the random variable $\mathcal{L}(\boldsymbol{X})$. We denote this survival function by $\overline{F}_{\mathcal{L}(\boldsymbol{X})}$. A simple inversion argument yields

$$\int_0^\infty \overline{F}_{\mathcal{L}(\boldsymbol{X})}(l)\,\mathrm{d}l = \int_0^1 \overline{F}^{-1}_{\mathcal{L}(\boldsymbol{X})}(p)\,\mathrm{d}p, \tag{2.5}$$

where $\overline{F}^{-1}_{\mathcal{L}(\boldsymbol{X})}(p)$ is the $(1-p)$–quantile function of the likelihood under η. This simple one-dimensional representation suggests that if one had access to the function $\overline{F}^{-1}_{\mathcal{L}(\boldsymbol{X})}$, the integral could then be approximated by numerical methods. For example, for a discrete set of values, $0 < p_T < \cdots < p_1 < p_0 = 1$, one could compute the Riemann sum

$$\sum_{t=1}^{T}(p_t - p_{t-1})\overline{F}^{-1}_{\mathcal{L}(\boldsymbol{X})}(p_t), \tag{2.6}$$

as a (deterministic) approximation of $\mathcal{Z}$. Unfortunately, the quantile function of interest is often intractable. NS provides an approximate way of performing quadrature such as (2.6). The core insight underlying NS is as follows. For N *independent* samples $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ from a density of the form

$$\eta(\boldsymbol{x}; l) := \frac{\eta(\boldsymbol{x})\mathbb{I}\{\mathcal{L}(\boldsymbol{x}) > l\}}{\mathbb{P}_\eta(\mathcal{L}(\boldsymbol{X}) > l)}, \quad \boldsymbol{x} \in E, \quad l \in \mathbb{R}_+, \tag{2.7}$$

we have that

$$\frac{\overline{F}_{\mathcal{L}(\boldsymbol{X})}\left(\min_k \mathcal{L}(\boldsymbol{X}^k)\right)}{\overline{F}_{\mathcal{L}(\boldsymbol{X})}(l)} \sim \mathsf{Beta}(N, 1). \tag{2.8}$$

Put simply, consider that one has N independent samples distributed according to the prior subject to a given likelihood constraint, and then introduces a new constraint determined by choosing the minimum likelihood value of the samples. This will define a region that has less (unconstrained) prior probability by a factor that has a $\mathsf{Beta}(N, 1)$ distribution. As samples from this new distribution will be compressed into a smaller region of the original prior, (2.8) is often referred to as a *compression factor*.

With this in mind, Skilling [48] proposes the NS procedure that proceeds as follows. Initially, a population of N independent samples (henceforth called particles) are drawn from η. Then, for each iteration $t = 1, \ldots, T$, the particle with the smallest value of $\mathcal{L}$ is identified. This "worst performing" particle at iteration t is denoted by $\check{\boldsymbol{X}}_t$ and its likelihood value by L_t. Finally, this particle is moved to a new position that is determined by drawing a sample according to $\eta(\cdot; L_t)$. By construction, this procedure results in a population of samples from η that is constrained to lie above higher values of $\mathcal{L}$ at each iteration.

After T iterations, we then have $\{L_t\}_{t=1}^T$. Each L_t corresponds to an unknown p_t satisfying $L_t = \overline{F}^{-1}_{\mathcal{L}(\boldsymbol{X})}(p_t)$. Skilling proposes to (deterministically) approximate the p_t values by assuming that at each iteration the

compression factor (2.8) is equal to its *geometric* mean, i.e., $\exp\big(\mathbb{E}\log(C)\big) = \exp(-1/N)$. Thus, we have the approximation $p_t = \exp(-t/N)$. This is the most popular implementation, and thus will be the version we consider for the remainder of this paper; however, it is worth noting that there exists another variant which randomly assigns $p_{t+1} = p_t B_t$ at each iteration, where $B_t \sim \mathsf{Beta}(N, 1)$. With the pairs $(L_t, p_t)_{t=1}^T$ in hand, the numerical integration is then of the form

$$\widehat{\mathcal{Z}} = \sum_{t=1}^{T} \underbrace{(p_t - p_{t-1})L_t}_{\widehat{\mathcal{Z}}_t}. \tag{2.9}$$

In practice, the number of iterations T is not set in advance, but rather the iterative sampling procedure is repeated until some termination criterion is satisfied. The standard approach is to continue until $p_t \cdot \max_{1\le j\le N} \mathcal{L}(\boldsymbol{X}^j) < \varepsilon \sum_{j=1}^{t} \widehat{\mathcal{Z}}_j$, where ϵ is some small value, say 10^{-8}. This choice attempts to ensure that the remaining integral is sufficiently small so that error arising from omission of the final $[0, p_T]$ in the quadrature is negligible.

In addition to estimates of the model evidence $\mathcal{Z}$, estimates of posterior expectations $\mathbb{E}_\pi[\varphi(\boldsymbol{X})]$, as in (2.1), can be obtained by assigning to each $\check{\boldsymbol{X}}_t$ the weight $w_t = \widehat{\mathcal{Z}}_t$, and using

$$\sum_{t=1}^{T} \varphi(\check{\boldsymbol{X}}_t) w_t \Big/ \sum_{t=1}^{T} w_t, \tag{2.10}$$

as an estimator. A formal justification for this is given in [13, Section 2.2], though in essence it is based on the fact that the numerator and denominator of (2.10) are (NS) estimators of their corresponding terms in the identity

$$\mathbb{E}_\pi[\varphi(\boldsymbol{X})] = \int_E \eta(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})\varphi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \Big/ \int_E \eta(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{2.11}$$

Pseudocode for NS is provided in Algorithm 1.

While the estimator (2.10) bears some resemblance to importance sampling (which is introduced in Section 2.3) in its use of a ratio estimator and weighted samples, it is not precisely the same.

2.2.1 Why isn't nested sampling more popular with statisticians?

There are several potential issues with NS that we speculate are the reasons why it has not achieved mainstream adoption in the statistics community. We outline what we believe to be the main four objections to NS below (in decreasing order of severity), as well as a discussion on relevant works that have attempted to address them.

Algorithm 1: Nested Sampling

input : population size N
$t \leftarrow 0$
for $k = 1$ **to** N **do** draw $\boldsymbol{X}^k \sim \eta$
while (not terminate) **do**
  $t \leftarrow t + 1$
  $m \leftarrow \operatorname{argmin}_{1 \le k \le N} \mathcal{L}(\boldsymbol{X}^k)$ `// identify worst-performing sample`
  $L_t \leftarrow \mathcal{L}(\boldsymbol{X}^m)$
  $w_t \leftarrow \big(\exp(-(t-1)/N) - \exp(-t/N)\big) L_t$
  $\breve{\boldsymbol{X}}_t \leftarrow \boldsymbol{X}^m$ `// save sample for inference`
  $\boldsymbol{X}^m \leftarrow$ a sample from $\eta(\cdot; L_t)$ `// move worst-performing particle`
end
$T \leftarrow t$
return estimator of the evidence $\widehat{\mathcal{Z}} = \sum_{t=1}^{T} w_t$ and weighted samples $\{\breve{\boldsymbol{X}}_t, w_t\}_{t=1}^{T}$.

1. **Assumption of Independent Samples**. The property (2.8) requires at each iteration independent samples with the correct distribution. This is a strong condition, as generating samples from constrained densities of the form (2.7) is in general difficult. The sampling method originally proposed is to move the worst performing particle at each iteration to the position of one of the other particles, and then run an MCMC algorithm for sufficiently many iterations to create an (approximately) independent sample. This procedure itself does not ensure the assumption of independence is satisfied, as it only produces independent samples asymptotically in the number of MCMC iterations. Moreover, for problems with likelihoods that have multiple well-separated modes, the constrained density will have increasingly isolated islands of support as the algorithm progresses, making it difficult for most samplers to cross between modes in any reasonable amount of time. Thus, even *approximate* independence may be difficult to achieve (and verify) in practice.

   Indeed, now over a decade after the introduction of the NS method, establishing consistency when MCMC transitions are used for sampling with NS remains a challenging open problem. Chopin and Robert [13] remark that "a reason why such a theoretical result seems difficult to establish is that each iteration involves both a different MCMC kernel and a different invariant distribution". In order to overcome the need for MCMC sampling, they propose a variant of NS for which the sampling can be performed exactly, and that demonstrate it can perform well in low dimensional problems for which π is approximately Gaussian.

   In a separate attempt to overcome dependency between samples, there is a class of approximate sampling methods called *region sampling* that attempts to generate independent samples by reparameterizing the problem so the constrained sampling problem is one of sampling

uniformly within constrained regions of a unit hypercube. The most popular of these methods is MultiNest [24], which uses the population of particles to construct a region that is a union of hyperellipsoids, sampling from this region, and accepting samples which satisfy the constraint. There is no way however to ensure the proposal region is a superset of the *actual* region. Buchner [7] proposes a method that is more robust (but still not immune) to this problem; however, the results show it can be an order of magnitude more inefficient and is more susceptible to the curse of dimensionality.

2. **Effect of Quadrature on Posterior Inferences**. As shown in (2.10), the ratio of two NS estimators (from a single run) can be used for posterior inferences. However, the precise effect of the use of quadrature in both estimators on estimates of $\pi(\varphi)$ is not well understood. The algorithm replaces the integral of $\mathcal{L}$ over a (random) *shell* $\{\boldsymbol{x} \in E : L_t < \mathcal{L}(\boldsymbol{x}) < L_{t+1}\}$ with a single value, and assigns a volume to that shell according to a geometric expectation. To our knowledge, the only work toward better understanding this unique form of error is [30], which quantifies it through bootstrapping techniques.

3. **Parallelization**. While NS can be parallelized across runs, NS does not allow one to make use of parallel computing architectures *within* runs without modifying the algorithm. The most natural way to parallelize NS, first proposed in [8] is as follows. If we generalize (2.8) to consider the K–th order statistic instead of simply the minimum ($K = 1$), then $1 - C$ has a $\mathsf{Beta}(K, N + 1 - K)$ distribution, with expectation $K/(N + 1)$. Thus, at each iteration, we can instead remove the K points with the lowest likelihood, set $p_t = (1 - K/(N + 1))^t$, and parallelize the sampling across K threads. The approach will not only increase the bias of the algorithm by introducing additional quadrature error, but will also compound the problem mentioned in the previous issue (as a single value will now be used to represent the mean of a larger shell).

4. **Truncation Error**. Finally, of lesser concern, yet still worth noting is that NS commits an $O(\exp(-T/N))$ truncation error [22] in the evidence estimate as a result of not performing quadrature on the entire $[0, 1]$ interval. A heuristic originally proposed by Skilling, which we call the *filling in* procedure is to simply add $\frac{1}{N}\sum_{k=1}^{N} \mathcal{L}(\boldsymbol{X}^k)$ after termination to the final evidence estimate. However, this is somewhat out of place with the rest of the quadrature. Using point process theory and techniques from the literature on unbiased estimation, Walter [52] proposes an unbiased version of NS. However, this unbiasedness relies on the assumption of independent sampling, and comes with a cost of additional variance.

As mentioned earlier, all of these potential issues stem from the use of quadrature in NS. Indeed, the

combined Monte Carlo/quadrature approach of NS seems to give somewhat of an overall awkwardness to the method. In the next section, we introduce SMC methodology, which we will soon discover allows us to retain the essence of NS, but allay the objections just discussed.

2.3 Sequential Monte Carlo

We begin with an introduction to importance sampling, which is the fundamental idea behind SMC. Recall that, in our setting, $\pi(\boldsymbol{x}) \propto \gamma(\boldsymbol{x})$, where γ is a known function. For any probability density ν such that $\nu(\boldsymbol{x}) = 0 \Rightarrow \pi(\boldsymbol{x}) = 0$, it holds that

$$\begin{aligned} \pi(\varphi) = \mathbb{E}_\pi[\varphi(\boldsymbol{X})] &= \int_E \varphi(\boldsymbol{x}) w(\boldsymbol{x}) \nu(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \Big/ \int_E w(\boldsymbol{x}) \nu(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &= \mathbb{E}_\nu[\varphi(\boldsymbol{X}) w(\boldsymbol{X})] / \mathbb{E}_\nu[w(\boldsymbol{X})], \end{aligned} \tag{2.12}$$

where $w(\boldsymbol{x}) = \gamma(\boldsymbol{x})/\nu(\boldsymbol{x})$ is called the *weight* function.

This suggests that one can draw $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^N \sim \nu$ and estimate (2.12) via the ratio

$$\sum_{k=1}^N \varphi(\boldsymbol{X}^k) w(\boldsymbol{X}^k) \Big/ \sum_{k=1}^N w(\boldsymbol{X}^k) = \sum_{k=1}^N \varphi(\boldsymbol{X}^k) \underbrace{\left(w(\boldsymbol{X}^k) \Big/ \sum_{k=1}^N w(\boldsymbol{X}^k) \right)}_{W^k},$$

where we call the $\{W^k\}_{k=1}^N$ the *normalized weights*.

A common measure of the quality of using ν with regard to approximating $\pi(\varphi)$ is the *effective sample size* (ESS),

$$\mathrm{ESS} := \mathbb{E}_\nu\big[w(\boldsymbol{X})\big]^2 / \mathbb{E}_\nu\big[w(\boldsymbol{X})^2\big].$$

In practice, this can be estimated via

$$\widehat{\mathrm{ESS}} = \left(\sum_{k=1}^N w(\boldsymbol{X}^k) \right)^2 \Big/ \sum_{k=1}^N w(\boldsymbol{X}^k)^2 = \left(\sum_{k=1}^N \left(W^k \right)^2 \right)^{-1}, \tag{2.13}$$

see [35, Chapter 2.5] for a full discussion. Unfortunately, in difficult high-dimensional settings, it is often hard to make a choice of importance sampling density to ensure that the ESS will be high (equivalently, that the variance of the normalized weights will be low).

SMC samplers [17] extend the idea of importance sampling to a general method for sampling from a sequence of probability densities $\{\pi_t\}_{t=1}^T$ defined on a common space E, as well as estimating their associated normalizing constants $\{\mathcal{Z}_t\}_{t=1}^T$ in a sequential manner. This is accomplished by obtaining at

each time step $t = 1, \ldots, T$ a collection of random samples (called *particles*) with associated (normalized) weights $\{\boldsymbol{X}_t^k, W_t^k\}_{t=1}^T$, for $k = 1, \ldots, N$, such that the weighted empirical measures of the cloud of particles,

$$\pi_t^N(\mathrm{d}\boldsymbol{x}) = \sum_{k=1}^{N} W_t^k \delta_{\boldsymbol{X}_t^k}(\mathrm{d}\boldsymbol{x}), \quad t = 1, \ldots, T, \tag{2.14}$$

converge to their corresponding *target* measures $\pi_t(\mathrm{d}\boldsymbol{x})$ as $N \longrightarrow \infty$.

SMC samplers have three main elements:

1. **Mutation**. For each iteration $t > 1$, the population of particles are moved from $\{\boldsymbol{X}_{t-1}^k\}_{k=1}^N$ to $\{\boldsymbol{X}_t^k\}_{k=1}^N$ according to a (forward in time) Markov kernel K_t, for which we denote the associated density $K_t(\boldsymbol{x}' \,|\, \boldsymbol{x})$. This implicitly defines a new importance sampling density at each iteration via the recursive formula
$$\nu_t(\boldsymbol{x}') = \int_E \nu_{t-1}(\boldsymbol{x}) K_t(\boldsymbol{x}' \,|\, \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{2.15}$$
2. **Correction.** The weights of the particles are updated via an *incremental importance weight* function $\tilde{w}_t$, to ensure the particle system is correctly reweighted with respect to the next target density. This update involves multiplying the current weight of each particle by a corresponding incremental weight.
3. **Selection.** The particles are resampled according to their weights, which are then reset to $1/N$. A variety of resampling schemes can be used (see for example [20, Section 3.4]. However, the simplest is multinomial resampling. Here, the resampled population contains C_k copies of $\boldsymbol{X}_t^k$ for each $k = 1, \ldots, N$, where $(C_1, \ldots, C_N) \sim \mathsf{Multinomial}(N, (W_t^k)_{k=1}^N))$.

Del Moral et al [17] show that one can use an arbitrary mutation kernel at each stage, without being required to compute the corresponding importance sampling density ν_t at each iteration. This is achieved by introducing an artificial backward (in time) kernel, which transforms the problem into one in the well–understood setting of filtering (for a comprehensive survey, see [20]). Here, sample paths of the particles' positions take values on the product space E^T, with the artificial joint distribution admitting each $\pi_t \propto \gamma_t$ (where γ_t is the unnormalized density) as a marginal. SMC samplers can be formulated in many different ways. For our purpose, we require SMC samplers for which K_t for $t > 1$ is a π_t-invariant MCMC kernel (or several iterations thereof). This approach is most straightforward and related directly to NS. For this case, a suitable choice of the *incremental weight* function at time t (i.e., one that will ensure the convergence (2.14)) is

$$\widetilde{w}_t(\boldsymbol{x}_{t-1}) = \gamma_t(\boldsymbol{x}_{t-1}) / \gamma_{t-1}(\boldsymbol{x}_{t-1}). \tag{2.16}$$

In this setting, the implicit backward kernel will be a good approximation to the *optimal* backward kernel, provided that π_t and π_{t-1} are sufficiently close.

SMC samplers give an approximation of $\pi_t(\varphi)$ at each iteration via

$$\pi_t^N(\varphi) := \sum_{k=1}^{N} W_t^k \varphi(\boldsymbol{X}_t^k). \tag{2.17}$$

Further to this, at each iteration SMC samplers give estimates of the ratios of normalizing constants

$$\widehat{\mathcal{Z}_t/\mathcal{Z}_{t-1}} = \underbrace{\sum_{k=1}^{N} W_{t-1}^k \widetilde{w}_t(\boldsymbol{X}_{t-1}^k)}_{\pi_t^N(\widetilde{w}_t)}, \text{ and } \widehat{\mathcal{Z}_t/\mathcal{Z}_1} = \prod_{k=2}^{t} \widehat{\mathcal{Z}_k/\mathcal{Z}_{k-1}}.$$

Somewhat remarkably, the estimators $\widehat{\mathcal{Z}_t/\mathcal{Z}_1}$ are unbiased, i.e, $\mathbb{E}\left[\widehat{\mathcal{Z}_t/\mathcal{Z}_1}\right] = \mathcal{Z}_t/\mathcal{Z}_1$. It follows readily that one can also obtain unbiased estimates of $\mathcal{Z}_t$ if $\mathcal{Z}_1$ is known, by including the $\mathcal{Z}_1$ term when γ_1 appears in the incremental weights.

Remark 1. (Adaptivity) Introducing any sort of adaptivity into the SMC algorithm, for example resampling only if some criteria is met, choosing the next distribution online, or setting the parameters of K_t according to the particle population, will not necessarily preserve the unbiasedness or convergence properties of the SMC estimators. The analysis of adaptive SMC methods is technically involved. However, there are consistency results for certain adaptive schemes, see for example [2], and [9], and [19]. Of course, one can always first run the algorithm adaptively, save the values of any adaptively chosen parameters, and then rerun the algorithm a second time and with these fixed.

SMC Samplers for Static Models

Del Moral et al [17] provide a strategy for using an SMC sampler to sample from a fixed density π by defining a sequence of densities $\pi_1, \pi_2, \ldots, \pi_T$ that transition from something that is easy to sample from (for example, the prior density) to π. This can be accomplished in a number of ways. We outline the two most common in the SMC literature. One approach ([11]) is to define the sequence of target distributions such that at each stage the effect of the likelihood is gradually introduced by considering more data than the last. The second method, first explored by Neal [40] is called *temperature annealing*. Here, we have the sequence of densities

$$\pi_t(\boldsymbol{x}) \propto \nu(\boldsymbol{x})^{1-l_t} \pi(\boldsymbol{x})^{l_t}, \quad t = 1, \ldots, T. \tag{2.18}$$

parametrized by some *temperature schedule*

$$l_1 = 0 < l_1 < \cdots < l_{T-1} < l_T = 1,$$

where ν is some initial importance sampling density. In the Bayesian setting, a natural choice is the gradual change from prior to the posterior:

$$\pi_t(\boldsymbol{x}) \propto \eta(\boldsymbol{x})\mathcal{L}(\boldsymbol{x})^{l_t}, \quad t = 1, \ldots, T.$$

In practice, it is often difficult to make a good choice for the temperature schedule. This can be achieved (approximately) by choosing the next temperature $l_{t+1} \in (l_t, 1]$ adaptively online according to the criterion of effective sample size (ESS), as proposed in [31]. This ensures successive distributions are sufficiently close. For some $\alpha \in (0, 1)$, one can approximately maintain an ESS of αN between successive distributions by choosing the next temperature online so that a given ESS is maintained. In other words, for a collection of particles, we choose L_{t+1} (and thus the next density) so that the ESS for the current importance sampling step is equal to some desired amount. Formally stated, for $\tilde{w}_{t+1}^k(l) := W_t^k \exp\left(-(l - L_t)\log \mathcal{L}(\boldsymbol{X}_t^k)\right)$, we solve

$$L_{t+1} = \inf_{l:L_t<l\leq 1} \left\{ \left(\sum_{k=1}^N \widetilde{w}_{t+1}(l) \right)^2 \Big/ \sum_{k=1}^N \widetilde{w}_{t+1}^k(l)^2 = \alpha N \right\}, \tag{2.19}$$

via the bisection method, for example. Pseudocode for adaptive temperature–annealed SMC (TA–SMC) is given in Algorithm 2.

Algorithm 2: Adaptive Temperature–Annealed SMC

input : population size N
$t \leftarrow 1$, $L_1 \leftarrow 0$, $\mathcal{Z} \leftarrow 1$
for $k = 1$ **to** N **do** draw $\boldsymbol{X}_1^k \sim \eta$ and set $W_1^k = 1/N$
while $\gamma_t \neq 1$ **do**
  $t \leftarrow t + 1$
  $L_t \leftarrow$ solution to (2.19) obtained via bisection
  for $k = 1$ **to** N **do** $w_t^k \leftarrow W_{t-1}^k \mathcal{L}(\boldsymbol{X})^{L_t - L_{t-1}}$
  $\widehat{\mathcal{Z}} \leftarrow \widehat{\mathcal{Z}} \left(\sum_{k=1}^N w_t^k \right)$
  $\{\widetilde{\boldsymbol{X}}_{t-1}^k\}_{k=1}^N \leftarrow$ resample $\{\boldsymbol{X}_{t-1}^k\}_{k=1}^N$ according to $\{w_t^k\}_{k=1}^N$
  for $k = 1$ **to** N **do** $W_t^k \leftarrow 1/N$
  $\{\boldsymbol{X}_t^k\}_{k=1}^N \leftarrow \text{move}\left(\{\widetilde{\boldsymbol{X}}_{t-1}^k\}_{k=1}^N, K_t\right)$
end
return samples $\{\boldsymbol{X}_t\}_{k=1}^N$ and estimator of the marginal likelihood, $\widehat{\mathcal{Z}}$.

2.4 Nested Sampling via Sequential Monte Carlo

The similarity between SMC and NS at this point is evident. Both methods draw from some initial distribution (in our case, the prior distribution), and involve traversing a population of particles through a sequence of distributions, which is of an adaptive nature in NS, but may be either adaptive or fixed in SMC. From the outset, it would *seem* that nested sampling is some sort of SMC algorithm, yet it is distinct in its use of a quadrature rule. Further, it has an interesting point of difference in that NS does not transition from the prior to the posterior.

It turns out, somewhat suprisingly, that this difference is largely a matter of interpretation. Nested Sampling *is* a special type of adaptive SMC algorithm, where weights are assigned in a suboptimal way. In order to demonstrate this in a straightforward manner, we proceed as follows. We first present a general class of SMC methods called Nested Sampling via Sequential Monte Carlo (NS-SMC) methods. Then, we will proceed to show the correspondence with the original NS method by introducing an adaptive version of NS–SMC, and finally modifying this adaptive version further so that it more closely resembles (and is equivalent as $N \to \infty$) to NS.

We begin by considering a set *threshold schedule*,

$$l_1 = -\infty < l_2 < \cdots < l_T < l_{T+1} = \infty, \tag{2.20}$$

which in turn parametrizes a sequence of nested sets

$$E_1 = E \supset E_2 \supset \cdots \supset E_{T-1} \supset E_T,$$

via

$$E_t := \{\boldsymbol{x} \in E : \mathcal{L}(\boldsymbol{x}) \geq l_t\}, \quad t = 1, \ldots, T.$$

Next, define the sequence of constrained densities:

$$\eta_t(\boldsymbol{x}) = \underbrace{\frac{\eta(\boldsymbol{x})\mathbb{I}\{\boldsymbol{x} \in E_t\}}{\mathbb{P}_\eta(\boldsymbol{X} \in E_t)}}_{\mathcal{P}_t}, \quad t = 1, \ldots, T. \tag{2.21}$$

We now consider directly shells of $\mathcal{L}$, via the sets,

$$\breve{E}_t = \{\boldsymbol{x} \in E : l_t < \mathcal{L}(\boldsymbol{x}) \leq l_{t+1}\}, \quad t = 1, \ldots, T.$$

Observe that sets $(\breve{E}_t)_{t=1}^T$ form a partition of E, $\breve{E}_t \subset E_t$ for $t = 1, \ldots T-1$, and that because $l_{T+1} = \infty$, we have that $\breve{E}_T = E_T$. Next, we define a second set of densities, corresponding to constrained versions

of π to these shells,

$$\pi_t(\boldsymbol{x}) = \underbrace{\frac{\gamma(\boldsymbol{x})\mathbb{I}\{\boldsymbol{x} \in \breve{E}_t\}}{\int_E \gamma(\boldsymbol{x})\mathbb{I}\{\boldsymbol{x} \in \breve{E}_t\}\,\mathrm{d}\boldsymbol{x}}}_{\mathcal{Z}_t}, \quad t = 1, \ldots, T.$$

With the above in mind, we define a class of SMC Samplers called NS-SMC samplers, that have the following two properties:

1. Given samples targeting η_{t-1}, the importance sampling branches into two paths. One path targets the next constrained prior η_t, while the second targets (and terminates at) the constrained posterior π_{t-1}. This branching of importance sampling paths occurs for all but η_T, which proceeds only to π_T. This is illustrated in Figure 2.1.

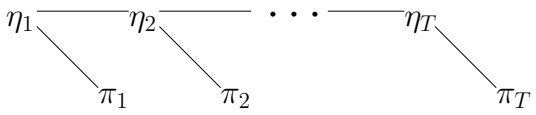


Figure 2.1: Importance sampling scheme for NS-SMC.

   The importance sampling procedure just described results in T (dependent) SMC samplers which output estimators of $\mathcal{Z}_t$, as well as samples that can be used to estimate $\pi_t(\varphi)$ for each $t = 1, \ldots, T$.

2. The resulting estimators for both $\mathcal{Z}_t$ and $\pi_t(\varphi)$ for $t = 1, \ldots, T$ are used together to estimate (2.1) via use of the identity

$$\pi(\varphi) = \sum_{t=1}^{T} \mathbb{P}_\pi(\boldsymbol{X} \in \breve{E}_t)\mathbb{E}_{\pi_t}[\varphi(\boldsymbol{X})] = \sum_{t=1}^{T} \frac{\mathcal{Z}_t}{\mathcal{Z}}\pi_t(\varphi). \tag{2.22}$$

For simplicity (and similarity to the original NS method), we consider the case where each η_t is used *directly* as an importance sampling density for π_t without any further resampling or moving. In such a case, we need only consider an SMC sampler that sequentially targets $\eta_1, \ldots, \eta_T$, because all terms in (2.22) can be rewritten in terms of expectations with respect to those densities. Thus, NS–SMC can be viewed as an extension to the rare–event SMC (multilevel splitting) method of Cérou et al [10], which uses density sequences of the form (2.21) in order to estimate the probability (normalizing constant) $\mathcal{P}_T$.

For ease of presentation, below we use shorthand notation analogously to (2.17). For example, instead of $\sum_{k=1}^{N} W_t^k \mathbb{I}\{\boldsymbol{X}_t^k \in E_t\}\mathcal{L}(\boldsymbol{X}_t^k)\varphi(\boldsymbol{X}_t^k)$, we write $\eta_t^N(\mathbb{I}_{E_t}\mathcal{L}\varphi)$.

Noting that $\pi_t/\eta_t = \mathcal{L}\mathbb{I}_{\breve{E}_t}$, we have

$$\pi(\varphi) = \sum_{t=1}^{T} \frac{\mathcal{Z}_t}{\mathcal{Z}} \pi_t(\varphi) = \sum_{t=1}^{T} \frac{\mathcal{Z}_t}{\mathcal{Z}} \frac{\eta_t(\mathcal{L}\mathbb{I}_{\breve{E}_t}\varphi)}{\eta_t(\mathcal{L}\mathbb{I}_{\breve{E}_t})}, \tag{2.23}$$

which is estimated via

$$\pi^N(\varphi) = \sum_{t=1}^{T} \frac{\widehat{\mathcal{Z}}_t}{\sum_{s=1}^{T} \widehat{\mathcal{Z}}_s} \cdot \frac{\eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t}\varphi)}{\eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t})}. \tag{2.24}$$

Note that

$$\begin{aligned} \widehat{\mathcal{Z}}_t \frac{\eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t}\varphi)}{\eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t})} &= \underbrace{\widehat{\mathcal{P}}_t \eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t})}_{\widehat{\mathcal{Z}}_t} \frac{\eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t}\varphi)}{\eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t})} = \widehat{\mathcal{P}}_t \eta_t^N(\mathcal{L}\mathbb{I}_{\breve{E}_t}\varphi) \\ &= \sum_{k=1}^{N} \widehat{\mathcal{P}}_t W_t^k \mathcal{L}(\boldsymbol{X}_t^k)\mathbb{I}\{\boldsymbol{X}_t^k \in \breve{E}_t\}\varphi(\boldsymbol{X}_t^k). \end{aligned} \tag{2.25}$$

The final equality above implies that reweighting with respect to the (full) posterior requires that each particle targeting η_t at iteration t is assigned the weight

$$\breve{w}_t^k = \widehat{\mathcal{P}}_t W_t^k \mathcal{L}(\boldsymbol{X}_t^k)\mathbb{I}\{\boldsymbol{X}_t^k \in \breve{E}_t\}.$$

In turn, we have that

$$\pi^N(\varphi) = \sum_{t=1}^{T}\sum_{k=1}^{N} \breve{W}_t^k \varphi(\boldsymbol{X}_t^k), \quad \breve{W}_t^k = \frac{\breve{w}_t^k}{\sum_{t=1}^{T}\sum_{k=1}^{N} \breve{w}_t^k}. \tag{2.26}$$

is an estimator of $\pi(\varphi)$.

Pseudocode for this version of NS–SMC is given in Algorithm 3. We call this version *Fixed* NS–SMC (as opposed to adaptive) as one specifies $\{l_t\}_{t=1}^{T}$ apriori. Note that resampling occurs at each iteration in order to avoid wasting computational effort moving particles with zero weight.

The issue of how to appropriately set $\{l_t\}_{t=1}^{T+1}$ will be addressed shortly. However, for now we return to the concerns with NS outlined earlier in Section 2.2.1, and note how they are addressed by NS–SMC:

1. **Assumption of Independent Samples.** NS-SMC has no requirement that the samples be independent. Moreover, the unbiasedness and consistency properties of Fixed NS–SMC are established in Appendix 2.8 via a straightforward application of Feynman–Kac formalism.

Algorithm 3: Fixed NS–SMC

input : population size N and thresholds $\{l_t\}_{t=1}^{T+1}$ satisfying (2.20).

$\widehat{\mathcal{P}}_1 \leftarrow 1$, $t \leftarrow 1$

for $k = 1$ **to** N **do** draw $\boldsymbol{X}_1^k \sim \eta$ and $W_1^k \leftarrow 1/N$

while true **do**

  $t \leftarrow t + 1$

  for $k = 1$ **to** N **do**

    $w_t^k \leftarrow W_{t-1}^k \mathbb{I}\{\mathcal{L}(\boldsymbol{X}_{t-1}^k) \geq l_t\}$ `// weight update for` $\eta_t \rightarrow \eta_{t+1}$

    $\breve{w}_{t-1}^k \leftarrow \widehat{\mathcal{P}}_{t-1} W_{t-1}^k \mathcal{L}(\boldsymbol{X}_{t-1}^k) \mathbb{I}\{\mathcal{L}(\boldsymbol{X}_{t-1}^k) < l_t\}$ `// weight for` π

  end

  $\widehat{\mathcal{P}}_t \leftarrow \widehat{\mathcal{P}}_{t-1} \left(\sum_{k=1}^N w_t^k \right)$ and $\widehat{\mathcal{Z}}_{t-1} \leftarrow \sum_{k=1}^N \breve{w}_{t-1}^k$

  if $\sum_{k=1}^N w_t^k = 0$ **then** $T \leftarrow t$ and **break**

  $\{\widetilde{\boldsymbol{X}}_{t-1}^k\}_{k=1}^N \leftarrow$ resample $\{\boldsymbol{X}_{t-1}^k\}_{k=1}^N$ according to $\{w_t^k\}_{k=1}^N$

  for $k = 1$ **to** N **do** $W_t^k \leftarrow 1/N$

  $\{\boldsymbol{X}_t^k\}_{k=1}^N \leftarrow \text{move}\left(\{\widetilde{\boldsymbol{X}}_{t-1}^k\}_{k=1}^N, K_t\right)$ — where K_t is an η_t–invariant MCMC kernel

end

$\widehat{\mathcal{Z}} = \sum_{t=1}^T \widehat{\mathcal{Z}}_t$

return weighted samples $\{\{\boldsymbol{X}_t^k, \breve{w}_t^k\}_{k=1}^N\}_{t=1}^{T+1}$ and estimator of the marginal likelihood, $\widehat{\mathcal{Z}}$.

2. **Effect of Quadrature on Posterior Inferences.** All errors in NS–SMC are solely Monte Carlo errors. The analogous error to that of NS in estimating $\pi(\varphi)$ is more natural and occurs as the result of the error in the ratios $\widehat{\mathcal{Z}}_t / \widehat{\mathcal{Z}}$ for $t = 1, \ldots, T$.

3. **Parallelization.** NS-SMC is easily parellizable without any further modification. After resampling, the move step, which is often the most computationally intensive, can be parallelized at the particle level.

4. **Truncation Error.** NS-SMC commits no truncation error as the final density π_T accounts for the interval $[0, p_T]$ which is omitted from the NS quadrature. However, it is important to note that the choice of the final threshold l_T will still have an effect on the variance of NS–SMC. Nevertheless, the absence of truncation error is a key factor in allowing NS–SMC to obtain unbiased estimates of $\mathcal{Z}$.

2.4.1 Adaptive NS-SMC

Generally one does not have a good idea of a choice of $\{l_t\}_{t=1}^T$ that will perform well. In a similar manner to adaptive TA–SMC, at each iteration t in Algorithm 3 we can replace l_t with a random threshold L_t that is chosen adaptively. Ensuring an estimated ESS for η_t that is at least $(1 - \rho)N$ simply

reduces to choosing L_t to be the $(1-\rho)$ quantile of the values $\left(\mathcal{L}(\boldsymbol{X}_{t-1}^k)\right)_{k=1}^N$. Such a choice in NS–SMC results in the online specification of both η_t and π_{t-1}. While $(1-\rho)$ is analogous to α, we use this notation as it is common in adaptive multilevel splitting algorithms [3, 4, 10], where ρ is interpreted as the proportion of particles that one desires to lie above each successive adaptively chosen threshold. For NS–SMC, $(1-\rho)$ can be interpreted as the desired proportion of samples with non–zero weight for π_{t-1}.

As with NS, Adaptive NS–SMC also requires that the iteration at which termination occurs is determined online in some manner. The termination criterion/procedure we use compares the evidence estimate after an iteration with an estimate that would be obtained by instead terminating at that iteration. At each iteration, after computing L_t, we compare the ratio of the two evidence estimates, and if it is greater than $1-\epsilon$, we instead set $L_t = \infty$, and declare $T = t-1$. In our examples, we found that the choice $\epsilon = 10^{-2}$ was suitable.

Remark 2. For a given adaptive choice of the next threshold L_t, experiments indicate that there is considerably less bias (particularly for small N) in the estimates of $\mathcal{Z}$ if one sets $\eta_t \propto \eta \mathbb{I}_{\{\mathcal{L}>L_t\}}$ and $\pi_{t-1} \propto \gamma \mathbb{I}_{\{L_{t-1}<\mathcal{L}\leq L_t\}}$ instead of $\eta_t \propto \eta \mathbb{I}_{\{\mathcal{L}\geq L_t\}}$ and $\pi_{t-1} \propto \gamma \mathbb{I}_{\{L_{t-1}\leq\mathcal{L}<L_t\}}$.

2.4.2 Improved NS

In this section, we follow the original NS sampling scheme more closely and derive an SMC estimator using a similar two-branched importance sampling scheme as illustrated in Figure 2.1. Specifically, we choose the sequence of distributions adaptively so only one particle lies below the next threshold and conduct our move step in a similar manner to NS. Just as Algorithm 3 can be viewed as an extension to rare-event SMC algorithm of Cérou et al [10], the more direct variant of NS we describe here can be viewed as an extension of the static *Last Particle Method* (LPM) for rare-event simulation [29]. Unfortunately, there is a lack of theoretical results for the LPM in the setting where MCMC is used (due mainly to the special type of move step, outlined shortly).

We call this method Improved Nested Sampling (INS). The sampling scheme is *identical* for NS and INS, and thus one can obtain both estimates from the same nested sampling run. Somewhat surprisingly, provided the filling in procedure is used, the NS and INS estimators of model evidence and posterior quantities *also* become identical as $N \to \infty$. This provides insight into why NS seems to perform well in practice despite a violation of the independence assumption that underlies its quadrature.

INS is a modified version of ANS–SMC with the following differences:

1. We enforce for iterations $t < T$ that a *single* particle has non-zero incremental weight for π_t. That is, like NS, we have only one particle that does not have support on the next constrained version of η.

2. We conduct the resampling and mutation step in a manner that ensures that MCMC is only required to replenish the "worst–performing particle".

Unfortunately, setting $\rho = (N-1)/N$ alone in ANS–SMC does not always ensure the first property above, which requires that all particles correspond to a *unique* value of $\mathcal{L}$. In discrete settings it is common for some particles to have the same value of $\mathcal{L}$. However, even if $\mathbb{P}_\eta(\mathcal{L}(\boldsymbol{X}) = l) = 0$ for all $l \in \mathbb{R}$, there may still be duplicate particles if there is a non–zero probability that the MCMC kernel will return the same point (as is the case in Metropolis–Hastings MCMC).

The solution is reasonably straightforward. We employ auxiliary variables in a similar manner to [38, pgs. 96–98], who proposes a variant of NS that can be applied to discrete spaces. A similar approach is used in [9] to break ties in the theoretical analysis of adaptive multilevel splitting.

For brevity, we assume the aforementioned condition that $\mathbb{P}_\eta(\mathcal{L}(\boldsymbol{X}) = l) = 0$ for all $l \in \mathbb{R}$, which is typically the case for continuous E. However, this condition excludes certain cases of what [48] refers to as "degenerate likelihoods". Under this assumption, the approach about to be described is entirely *implicit* if one does not consider any auxiliary variables, ignores any duplicate particles, and just moves a single particle at each iteration, yielding the same L_t for multiple iterations. However, in discrete cases, one must consider the extended space *explicity* and conduct the move step differently, see [39] and [38].

We extend the space from E to $E \times (0,1)$ via a uniformly distributed variable U. That is, we have

$$\eta(\boldsymbol{x}, u) = \eta(\boldsymbol{x})\mathbb{I}\{0 < u < 1\}, \text{ and } \pi(\boldsymbol{x}, u) \propto \gamma(\boldsymbol{x})\mathbb{I}\{0 < u < 1\}.$$

In this setting, define the *augmented* threshold schedule:

$$(l_1, v_1) = (-\infty, 0) < (l_2, v_2) < \cdots < (l_T, v_T) < (l_{T+1}, v_{T+1}) = (\infty, 1),$$

where $(l, v) < (l', v')$ is to be understood as either $l' > l$, or that both $l' = l$ and $v' > v$.

Applying a similar derivation of NS–SMC to that given earlier in this section, we have the sets

$$\begin{aligned} E_t &= \{(\boldsymbol{x}, u) \in E \times (0,1) : (\mathcal{L}(\boldsymbol{x}) > l_t, 0 < u < 1) \cup (\mathcal{L}(\boldsymbol{x}) = l_t, v_t < u < 1)\} \\ \breve{E}_t &= \{(\boldsymbol{x}, u) \in E \times (0,1) : (l_t < \mathcal{L}(\boldsymbol{x}) < l_{t+1}, 0 < u < 1) \cup (\mathcal{L}(\boldsymbol{x}) = l_t, v_t < u \leq v_{t+1})\}, \end{aligned} \tag{2.27}$$

and the densities

$$\eta_t(\boldsymbol{x},u) \propto \eta(\boldsymbol{x},u)\mathbb{I}\{(\boldsymbol{x},u) \in E_t\}, \text{ and } \pi_t(\boldsymbol{x},u) \propto \pi(\boldsymbol{x},u)\mathbb{I}\{(\boldsymbol{x},u) \in \breve{E}_t\}, \text{ for } t = 1,\ldots,T.$$

Note that this setup ensures that the E_t sets are nested and that the $\breve{E}_t$ sets form a partition.

Prior to demonstrating how INS relates to NS, we stress the following. Due to the special type of mutation step, the algorithm falls outside of the standard SMC sampler framework (which requires the same forward kernel for all particles).

Despite this, we continue to use incremental weight functions of the form (2.16). While this choice seems to be a natural one (and matches the approach used in the LPM), it implicitly assumes that the INS procedure produces a population of particles that are marginally distributed according to (the adaptively chosen) η_t at each time step. This may only hold approximately in practice, and even establishing that the property holds as $N \to \infty$ is difficult due to the complicated combination of adaptively chosen distributions and non–standard mutation step.

Nevertheless, we present the method for the purpose of making clear the connection of NS with the NS–SMC framework. Moreover, we point out that while our assumption on the marginal distribution of the particles at each iteration is a strong one, it is substantially weaker than that required in the original formulation of NS, which assumes not only the same condition on the marginal distributions of the particles, but *also* that the particles are independent. Recall that both of these conditions are required for the property (2.8) to hold.

With the above in mind, we sketch the key aspects of the INS below.

Adaptive Choice of Densities. At each (non–final) iteration, we determine π_{t-1} and η_t adaptively (via the choice of of the next threshold parameters L_t and V_t) as follows. First, we set $L_t = \min_{1\leq k\leq N}\mathcal{L}(\boldsymbol{X}_{t-1}^k)$. Next, denote the indices of the particles satisfying $\mathcal{L}(\boldsymbol{x}) = L_t$ by $\mathcal{I}$. We "break ties" by choosing $V_t = \min_{k\in\mathcal{I}}\{U_{t-1}^k\}$.

Reweighting. Importance sampling takes place for η_t and π_{t-1} with the incremental weight functions $\mathbb{I}_{E_t}$ and $\mathcal{L}\mathbb{I}_{\breve{E}_{t-1}}$, respectively.

By construction, we will have $N-1$ samples with non-zero incremental weight for η_t, giving

$$\widehat{\mathcal{P}}_t = \underbrace{\left(\frac{N-1}{N}\right)^{t-1}}_{\widehat{\mathcal{P}}_{t-1}} \underbrace{\frac{N-1}{N}}_{\eta_{t-1}^N(\mathbb{I}_{E_t})} = \left(\frac{N-1}{N}\right)^t.$$

Similarly, only one particle, denoted $(\check{\boldsymbol{X}}_{t-1}, \check{U}_{t-1})$, will have non–zero incremental weight for π_{t-1} (and thus non–zero weight for π), so we have

$$\widehat{\mathcal{Z}}_{t-1} = \underbrace{\left(\frac{N-1}{N}\right)^{t-1}}_{\widehat{\mathcal{P}}_{t-1}} \underbrace{\frac{1}{N}\mathcal{L}(\check{\boldsymbol{X}}_{t-1})}_{\eta_{t-1}^N(\mathcal{L}\mathbb{I}_{\check{E}_{t-1}})} . \tag{2.28}$$

Note that (2.28) is not only $\widehat{\mathcal{Z}}_{t-1}$ but also precisely the weight of $\check{\boldsymbol{X}}_{t-1}$ with respect to π as in (2.25). Recall that this is also the case with NS.

Resampling and Mutation. We resample according to a *residual* scheme, and reset all weights to $1/N$. As we have $N-1$ samples with equal (non-zero) weight, residual resampling will result in $N-1$ unique particle positions $\{\widetilde{\boldsymbol{X}}_{t-1}^k, \widetilde{U}_{t-1}^k\}_{k=1}^{N-1}$, as well as a final particle $(\widetilde{\boldsymbol{X}}_t^N, \widetilde{U}_t^N)$ that is a copy of one of the others.

The mutation step is as follows. We begin by applying the identity map all particles except the N–th one, moving $\{\widetilde{\boldsymbol{X}}_{t-1}^k, \widetilde{U}_{t-1}^k\}_{k=1}^{N-1}$ to $\{\boldsymbol{X}_t^k, U_t^k\}_{k=1}^{N-1}$. Then, we perform the following two η_t–invariant moves in sequence to move $(\widetilde{\boldsymbol{X}}_{t-1}^N, \widetilde{U}_{t-1}^N)$ to $(\boldsymbol{X}_t^N, U_t^N)$.

First, we move $\widetilde{\boldsymbol{X}}_{t-1}^N$ to $\boldsymbol{X}_t^N$ by applying some fixed number of iterations of an $\eta_t(\boldsymbol{x} \,|\, u)$–invariant MCMC kernel. Note that

$$\eta_t(\boldsymbol{x} \,|\, u) \propto \begin{cases} \eta(\boldsymbol{x})\mathbb{I}\{\mathcal{L}(\boldsymbol{x}) \geq L_t\} & u > V_t \\ \eta(\boldsymbol{x})\mathbb{I}\{\mathcal{L}(\boldsymbol{x}) > L_t\} & u \leq V_t \end{cases},$$

so this is simply sampling from a constrained version of η as in standard NS or NS–SMC. Next, we draw from a new u position according to

$$\eta_t(u \,|\, \boldsymbol{x}) \propto \begin{cases} \mathbb{I}\{0 < u < 1\} & \mathcal{L}(\boldsymbol{x}) > L_t \\ \mathbb{I}\{V_t < u < 1\} & \mathcal{L}(\boldsymbol{x}) = L_t \end{cases}.$$

Final Iteration. The reweighting and mutation steps continue up until a termination criteria is satisfied. At this point, we declare $T = t-1$, and set the final threshold parameters $L_t = \infty$, $U_t = 1$. Here, all samples will have non–zero incremental weight for π_T, and we have

$$\widehat{\mathcal{Z}}_T = \left(\frac{N-1}{N}\right)^T \frac{1}{N}\sum_{k=1}^{N} \mathcal{L}(\boldsymbol{X}_T^k). \tag{2.29}$$

Note that the above normalizing constant estimator bears similarity to the "filling in" heuristic in NS. However, here it arises *naturally* as a final step, and uses $\left(\frac{N-1}{N}\right)^T$ to estimate p_T, as opposed to $\exp(-T/N)$.

Despite this similarity, it still appears that (2.28) is distinct from its analogous term in NS. However, by means of some simple algebraic manipulation, we obtain the identity

$$\left(\frac{N-1}{N}\right)^t = \left(\frac{N-1}{N}\right)^{t-1}\left(1-\frac{1}{N}\right) = \left(\frac{N-1}{N}\right)^{t-1} - \left(\frac{N-1}{N}\right)^{t-1}\frac{1}{N},$$

which, after rearrangement, reveals that

$$\left(\frac{N-1}{N}\right)^{t-1}\frac{1}{N} = \left(\frac{N-1}{N}\right)^{t-1} - \left(\frac{N-1}{N}\right)^{t}, \tag{2.30}$$

resembling precisely the Riemann sum quadrature rule, with the choice $p_t = \left(\frac{N-1}{N}\right)^t$. The most notable aspect of this is that at no stage in the derivation of INS did we require the property given in (2.8), which would require samples to be independent.

We give this version of NS with the "improved" moniker as the alternative choices for p_t have been found to yield superior estimators of $\mathcal{P}_t$ when compared to those proposed originally by Skilling [48]. Guyader et al [29] show (under the same idealized sampling assumption as NS) that the LPM estimators $\widehat{\mathcal{P}}_t = \left(\frac{N-1}{N}\right)^t$ are *unbiased* estimators of $\mathbb{P}_\eta(\mathcal{L}(\boldsymbol{X}) \geq L_t)$. Further to this, [52, Remark 1] shows that this estimator results in superior estimates over $\exp(-t/N)$ in terms of variance so long as $p_t > \exp(-1)$. In light of this, Walter suggests using Riemann sum quadrature using these alternative values for p_t as it will result in a superior NS estimator.

The final piece of the puzzle connecting NS with INS and the overall NS–SMC framework, is that as $N \to \infty$ we have that $\left(\left(\frac{N-1}{N}\right)^t - e^{-t/N}\right) \to 0$, so NS's weights become equal to those of INS. We give a simple illustration of this convergence in Figure 2.2, where we plot the ratio of (2.30) to the standard NS Riemann sum / trapezoidal rule terms after $T/N = 10$ iterations of NS for different N (NS gives identical estimates for p_t for this choice, regardless of N). The convergence will be slower for larger T/N.

This provides some insight as to why NS seems to deliver correct results in practice, even when particles are far from independent (as will soon be demonstrated numerically). In essence, NS is an *adaptive* SMC algorithm on an extended space, where the choice of weights are, in a sense, sub–optimal.

2.4.3 Phase Transition Example

In order to compare the different variants of NS–SMC and NS, we shall use a phase transition example. Nested sampling is robust to models that contain phase transitions, i.e., models for which the graph of $\log p$ against $\log \mathcal{L}(F^{-1}_{\mathcal{L}(\boldsymbol{X})}(p))$ is not concave. For a full discussion of the challenges of phase transition

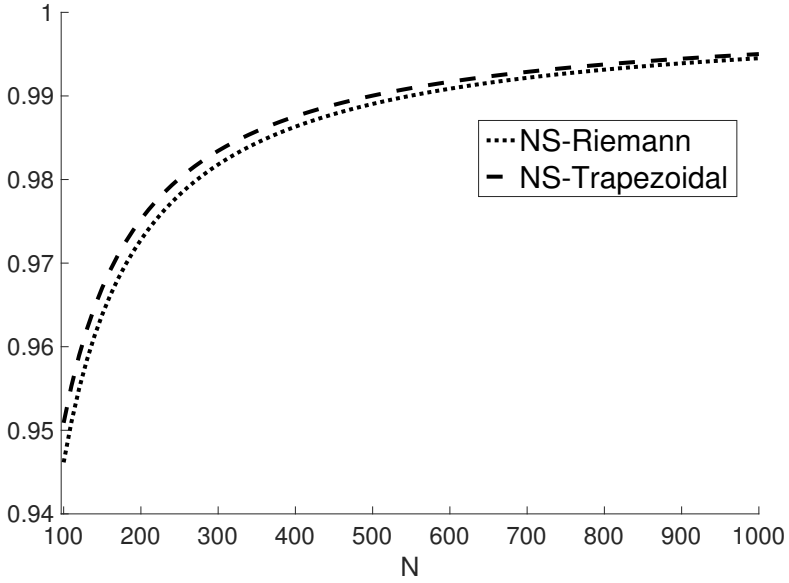


Figure 2.2: Ratio of weights for INS for $T/N = 10$, relative to NS estimators.

phenomena, including why they can be challenging for temperature based methods such as TA-SMC and the *power posteriors* method of [25], we refer back to the original NS paper [48]. In a Bayesian context, a phase transition can be understood intuitively as having a likelihood function that is spiked and changes rapidly in certain regions. While this would seem to be a pathological type of behaviour restricted to problems in physics, it is known to occur in statistical settings, see for example [5].

Similar to [48], we consider the estimation of

$$\mathcal{Z} = \int_{\mathbb{R}^n} \underbrace{\left(\sum_{k=1}^{2} a_k \phi_{\sigma_k}(\boldsymbol{x}) \right)}_{\mathcal{L}(\boldsymbol{x})} \underbrace{\frac{\mathbb{I}\{||\boldsymbol{x}|| < 1\}}{V(\mathcal{B}_n)}}_{\eta(\boldsymbol{x})} \mathrm{d}\boldsymbol{x},$$

where ϕ_σ denotes the pdf of a multivariate normal distribution with standard deviation σ for each component, centred at the origin, and $V(\mathcal{B}_n)$ denotes the volume of an n–dimension unit hypersphere. This problem can be viewed as estimating the model evidence of a model with uniform "prior" on the unit ball, and a mixture of two multivariate normals centered at the origin as a "likelihood function". Despite the conceptual simplicity of this problem, it is still difficult computationally, and we can introduce a phase transition by varying parameters appropriately. In our case, we introduce a phase transition by specifying $\boldsymbol{\sigma} = (0.1, 0.01)$ and $\boldsymbol{a} = (0.25, 0.75)$, which introduces a large "spike" in $\mathcal{L}$ due to the second mixture component. This particular example is also interesting as we are able to perform *exact sampling* from each η_t. This corresponds to using the *optimal* forward kernel.

In order to illustrate the effects of particle dependency, we also implement a version with MCMC. For an MCMC kernel, we perform ten iterations of a variant of the random walk sampler where we simply propose a movement along a randomly chosen coordinate axis. In order to ensure the sampler is well suited across progressively narrower densities, we choose h to be $1/10$ or $1/40$ with equal probability. We remark that this method strongly outperforms the obvious first choice of the standard random walk sampler. For NS and ANS-SMC, we use our knowledge of the problem to set the termination criterion to be $L_t/\mathcal{L}(\mathbf{0}) \geq 0.75$; i.e., we stop when the current threshold is higher than 75% of the maximum. While this ensures that the truncation error for NS is very small, we still use the filling–in procedure. For (fixed) NS–SMC, we use the thresholds obtained via a pilot run of ANS–SMC.

In terms of simulation effort, it is worth noting that a choice of $\rho = 0.37$ ($\approx \exp(-1)$) for ANS-SMC yields around the same number of likelihood evaluations as NS. This is because for one iteration of NS-SMC, we spend an effort proportional to N (each particle is moved/generated), whereas for NS the effort is proportional to 1. As $\exp(-1) \approx \left(\frac{N-1}{N}\right)^N$, we would expect roughly (discounting the effect of resampling and moving all the particles at once in the case of NS-SMC) that the two algorithms will have compressed a similar amount for prior mass for the same amount of likelihood evaluations. Thus, for purposes of a more direct comparison with NS / INS, we use this choice of ρ. We also implement adaptive TA–SMC for this example, where we we use the conservative choice of $\alpha = 0.95$ and 20 MCMC repeats. Note that TA–SMC with $\alpha = 0.95$ will attempt to maintain an ESS of $0.95N$ between successive distributions, and thus will progress slower and allow the particles to move around the space more.

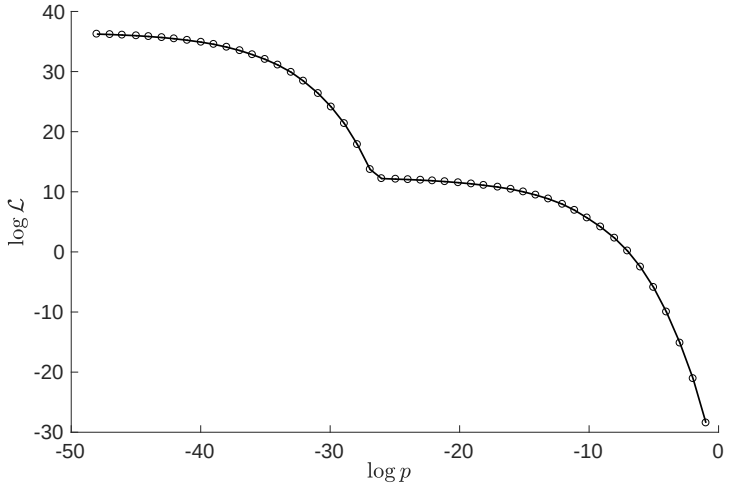


Figure 2.3: Phase transition diagnostic plot for the ten-dimensional sphere example. The phase transition appears around $\log p = -27$, corresponding to approximately 10^{-12} remaining prior mass.

The results given in Table 2.1 demonstrate several things. Firstly, we see that both the variance and bias in the integral estimate seems more pronounced for small N when MCMC is used. The observed (upward) bias for low N is a problem which seems to become more severe when samples are dependent.

Most notable is the exceptionally poor performance of TA–SMC, which fails on two accounts. Firstly, temperature–based methods are ill–suited to phase transitions. In the context of SMC, the nature of such problems means that the *actual* ESS will be low unless successive temperatures are very close. Secondly, at each iteration, TA–SMC vastly overestimates its ESS for any given l as a result of there being no particles in the spike. That is, TA–SMC is unable to identify when it has a poor approximation of the current target. This results in the adaptive choice of levels failing to achieve its goal, and producing very poor estimates as a result. For example, notice how for $N = 10^2$ and $N = 10^3$, it appears to have missed three quarters of the integral belonging to the spike completely. Interestingly, we see TA–SMC occasionally successfully find the spike in the $N = 10^4$ case, now giving a more accurate result, but with enormous variance. In contrast, NS–SMC's estimates of its own ESS will generally be better behaved, as the incremental weights along the η_t path will be either zero or one.

Table 2.1: Results for the 10-dimensional sphere example with phase transition. Results for $N = 10^2$ correspond to 1000 runs, while $N = 10^3$ and $N = 10^4$ correspond to 100 runs. We have that $\mathcal{Z} = 1$.

		$N = 10^2$		$N = 10^3$		$N = 10^4$	
sampler	method	$\widehat{\mathcal{Z}}$ (SE%)	evals	$\widehat{\mathcal{Z}}$ (SE%)	evals	$\widehat{\mathcal{Z}}$ (SE%)	evals
Exact	NS	1.14 (1.9)	5.0×10^3	1.01 (1.8)	5.0×10^4	1.000 (0.5)	5.0×10^5
	INS	0.99 (1.6)	5.0×10^3	1.00 (1.8)	5.0×10^4	0.999 (0.5)	5.0×10^5
	ANS–SMC	1.00 (2.2)	4.9×10^3	1.00 (2.1)	4.9×10^4	1.009 (0.7)	4.9×10^5
	NS–SMC	0.99 (2.4)	5.0×10^3	0.99 (2.1)	5.0×10^4	1.010 (0.6)	5.0×10^5
MCMC	NS	1.52 (9.8)	4.9×10^4	1.11 (5.2)	4.9×10^5	1.01 (1.6)	4.8×10^6
	INS	1.33 (8.6)	4.9×10^4	1.10 (5.1)	4.9×10^5	1.01 (1.5)	4.8×10^6
	ANS-SMC	1.19 (5.8)	4.8×10^4	1.06 (4.0)	4.8×10^5	1.02 (1.2)	4.8×10^6
	NS-SMC	1.01 (4.4)	4.8×10^4	0.94 (3.5)	4.8×10^5	1.00 (1.1)	4.8×10^6
	TA-SMC	0.24 (0.5)	4.7×10^4	0.25 (0.2)	4.8×10^5	1.03 (73)	4.8×10^6

While this example illustrates the similarity between NS and NS–SMC methods, showing how they can each handle phase transitions where TA–SMC has difficulty, the question arises as to how NS–SMC compares to TA–SMC on challenging and realistic problems. However, prior to conducting a comparative study, we first consider how one can attempt to ensure the best possible performance for both methods.

2.5 Calibration Methods

The implementation of SMC methods requires a specification of kernel parameters, and the number of MCMC iterations at *each* time step. As making a judicious choice of these parameters at each time step is a daunting task, it is common to use the same MCMC kernel parameters through the entire sequence. Likewise, it is common to use the same number of MCMC kernel iterations at each time step. Unfortunately, using a fixed scheme for kernel parameters and number of iterations does not take into account that the targets can become more (or less) difficult to sample from in later iterations. While SMC methods retain their convergence properties regardless of these factors, one would ideally like to choose them in a way that is in some sense optimal at each iteration, especially if we aim to make a fair comparison between different SMC methods. In this section we present some novel ways of approximately achieving this goal in practice.

2.5.1 Choice of Kernel Parameters

One of the major advantages of SMC samplers over MCMC and NS is the ability to use the population of particles at each time step to inform the choice of MCMC kernel parameters. For example, it is common (see, for example, [12]) to use the sample covariance matrix of the particles $\widehat{\Sigma}$ (an estimator of the global covariance Σ) in local proposals. However, when it comes to more general kernel parameter selection, it unfortunately remains common practice to use the same fixed kernel parameter across all time steps.

For MCMC samplers, Pasarica and Gelman [44] propose to select kernel parameters by maximizing the *expected square jump distance* (ESJD) for a single MCMC iteration, which is equivalent to minimizing the first order (lag–1) autocorrelation. For current state $\boldsymbol{X}_{\text{curr}}$ and state after an MCMC iteration $\boldsymbol{X}_{\text{new}}$, the ESJD is

$$\text{ESJD} := \mathbb{E}\,||\boldsymbol{X}_{\text{new}} - \boldsymbol{X}_{\text{curr}}||^2,$$

where $||\cdot||$ denotes some norm, and $\boldsymbol{X}_{\text{curr}}$ is distributed according to the target density. In the context of Metropolis–Hastings MCMC with proposal density $q(\boldsymbol{x}' \,|\, \boldsymbol{x})$, the ESJD is given by

$$\mathbb{E}[||\boldsymbol{X}_{\text{prop}} - \boldsymbol{X}_{\text{curr}}||^2\, \alpha_{\text{MH}}(\boldsymbol{X}_{\text{curr}}, \boldsymbol{X}_{\text{prop}})],$$

where $\boldsymbol{X}_{\text{prop}} \sim q(\boldsymbol{x}' \,|\, \boldsymbol{x}_{\text{curr}})$, and $\alpha_{\text{MH}}(\boldsymbol{X}_{\text{curr}}, \boldsymbol{X}_{\text{prop}})$ is the Metropolis Hastings acceptance probability of moving from $\boldsymbol{X}_{\text{curr}}$ to the proposal $\boldsymbol{X}_{\text{prop}}$. In this context, we can estimate the ESJD via

$$||\boldsymbol{X}_{\text{prop}} - \boldsymbol{X}_{\text{curr}}||^2\, \alpha_{\text{MH}}(\boldsymbol{X}_{\text{curr}}, \boldsymbol{X}_{\text{prop}}).$$

Fearnhead and Taylor [23] propose an adaptive SMC sampler that uses the estimated ESJD in its selection of MCMC kernel parameters. The method starts with an initial population of kernel parameters which is used in the first mutation step. After the first and all subsequent mutation steps, the population of kernel parameters is resampled according to ESJD and then jittered. Generally there will be many poor–performing kernel parameters in the early iterations, and this may lead to poor mixing that can affect later distributions. Moreover, the kernel parameters that were roughly optimal for the previous iteration are used as a basis for those in the next iteration. If the targets change in a way that significantly affects the optimal tuning parameter (for example, the separation of modes due to a new threshold in NS–SMC), then poor results can be expected. To avoid the use of many poor choices of parameters in early iterations and to make selection robust to changes in the optimal tuning parameter between iterations, we opt to select a single optimal tuning parameter per target based on a single MCMC iteration on all the particles.

Specifically, to automate the selection of a single optimal tuning parameter, we do the following. We specify a finite set of values for the tuning parameter and at each $t > 1$, each particle is randomly assigned one of these choices. We then perform a single MCMC iteration per particle and record the corresponding estimate of the ESJD. We follow both [44] and [23] in the choice of *Mahalanobis distance* as a norm, i.e., $||\boldsymbol{y}||_{\widehat{\Sigma}} := \sqrt{\boldsymbol{y}\,\widehat{\Sigma}^{-1}\,\boldsymbol{y}}$, where $\widehat{\Sigma}$ is an estimate of the global covariance matrix obtained from the particle positions.

The kernel parameter that produces the highest median estimated ESJD per target evaluation is selected and the remaining MCMC repeats are subsequently performed. Our method works well in combination with the method for tuning the number of MCMC repeats (which is explained shortly) and we illustrate how these methods work in an example in Figure 2.4.

Remark 3. When sampling from $\{\eta_t\}_{t=1}^{T}$ in NS–SMC using Metropolis–Hastings with proposal density $q(\boldsymbol{x}' \,|\, \boldsymbol{x})$, the acceptance probability becomes

$$\alpha_{\text{MH}} = \min\left\{1, \frac{\eta(\boldsymbol{x}')q(\boldsymbol{x}' \,|\, \boldsymbol{x})}{\eta(\boldsymbol{x})q(\boldsymbol{x} \,|\, \boldsymbol{x}')}\mathbb{I}\{\boldsymbol{x}' \in E_t\}\right\}.$$

While computing this quantity explicitly is required for estimating ESJD in the pilot run, we remark that when this is not required, it is more efficient to accept a proposal in two stages as follows. We *conditionally* accept with probability $\min\{1, \eta(\boldsymbol{x}')q(\boldsymbol{x}' \,|\, \boldsymbol{x})/\eta(\boldsymbol{x})q(\boldsymbol{x} \,|\, \boldsymbol{x}')\}$. Then, if a proposal has been conditionally accepted, we accept the proposal iff $\boldsymbol{x}' \in E_t$. This approach reduces the number of likelihood evaluations required for the same number of iterations, and is an additional benefit of NS–SMC samplers.

2.5.2 Choosing the Number of MCMC Iterations

Choosing the number of MCMC iterations per particle at each iteration in an efficient manner remains a challenging open problem. Computational effort aside, one would like the particles to be close to independent. However, in practice, we consider this too lofty a goal. For example, in the case of temperature annealing, at the final move step, achieving this is equivalent to ensuring burn-in for N standard MCMC samplers for π. Alternatively, one could focus less on attempting to ensure particle independence and instead try to ensure that there are N *unique* particles after the move step. For example, one could perform a single iteration of Slice Sampling to guarantee unique particles, but the average distance moved may be extremely small. In practice, a balance must be struck.

Drovandi and Pettitt [21] propose a formula to estimate the number of repeats required to move particles at least once with a specified probability in the context of a Metropolis–Hastings MCMC move step. The formula uses an average acceptance probability which can be estimated from the previous SMC iteration, or calculated with a single MCMC repeat for the current target as in [49]. Although this method is relatively simple to implement, it does not consider the quality of the proposed moves in terms of jumping distance. Large proposals that are accepted with small probability are given more repeats than small proposals that are accepted with high probability. In practice, this method is effective at ensuring a collection of unique particles, but the uniqueness of particles does not guarantee quality of the particle approximation. Furthermore, this method is not sensible in the context of moves with guaranteed acceptance, such as Slice Sampling. A second approach, given by Ridgway [45], is to check for convergence or stabilization of the moves. The sum (over the particles) of the absolute move distances at each MCMC is recorded and one should iterate until this quantity stabilizes. However, suggestions are not given as to precisely what defines stabilization of this quantity, or how to check for this in an automatic manner. Furthermore, we find that if the resampled particles already represent a reasonable approximation to the target, as they do in the context of NS–SMC and TA–SMC with a sufficiently large ρ/α, then stabilization becomes even more difficult to determine.

In light of this, we propose an approach that allows the particles to perform a reasonable level of exploration. Define the *expected jump distance* for a single MCMC repeat and particle to be $J := \mathbb{E}\,||\boldsymbol{X}_{\text{new}} - \boldsymbol{X}_{\text{curr}}||_{\widehat{\Sigma}}$. In Metropolis–Hastings (MH) MCMC, we can estimate J via

$$||\boldsymbol{X}_{\text{prop}} - \boldsymbol{X}_{\text{curr}}||_{\widehat{\Sigma}}\, \alpha_{\text{MH}}(\boldsymbol{X}_{\text{curr}}, \boldsymbol{X}_{\text{prop}}).$$

For R iterations of an MCMC kernel, we have

$$\widehat{J}(R) := \sum_{r=1}^{R} \widehat{J}_r, \tag{2.31}$$

where $\widehat{J}_r$ denotes the estimate of expected jump distance obtained from the r-th iteration. For some specified quantity J_{desired}, we propose to continue iterating the MCMC kernel over all particles until a specified proportion of the particles satisfies $\widehat{J}(R) > J_{\text{desired}}$. Note that the sum in (2.31) is over the MCMC iterations, rather than over the particles as was the case in [45].

Our proposed method requires a choice of the proportion of particles as well as a choice of J_{desired}. Both can be chosen based on how conservative the move step should be. In all of our examples, we choose J_{desired} online by using the (weighted) mean Mahalanobis distance between particles before resampling. We continue to perform repeats until 50% of particles satisfy $\widehat{J}(R) > J_{\text{desired}}$.

This method can be implemented to tune the repeats online and is well–suited to comparing/selecting different MCMC kernels that differ in terms of acceptance rate and jumping distance. Our experiments also find it to be more robust to sub–optimal tuning parameters than the acceptance probability based method of [21]. While the method does not account for possible back–and–forth behaviour of the sampler, we find that it works well for all samplers across both SMC methods. We note that the underlying method of iterating until a desired criteria is observed for a specified number of particles is quite general and can encompass a wide range of goals for the move step. For example, using other measures of distance is possible, as is considering the sum of *actual* distances moved.

It is important to note that choosing the number of MCMC repeats online is a form of adaptivity. Thus, we recommend this approach is only used in the pilot run (where the sequence of distributions and kernel parameters are also chosen adaptively), in order to determine the number of repeats for fixed SMC runs.

2.6 Comparison with TA–SMC

While we saw in Section 2.4.3 that variants of NS–SMC are capable of handling phase transitions as well as NS, with less bias, the question of how NS–SMC compares with TA–SMC arises. In this section, we compare the different SMC approaches on two challenging Bayesian statistical inference problems.

Our intention is not necessarily to demonstrate the superiority of our proposed method over TA-SMC. Given the variety of possible parameters for SMC (i.e., N and ρ/α) as well as many possible MCMC kernels, methods of tuning them, and choices for number of MCMC repeats at each iteration, there most likely exists an appropriate choice of these factors for any given problem that will allow one method to outperform the other. Thus, we instead aim to simply make our best efforts using our experience with

SMC to get the best out of both algorithms in an automated manner, and observe the results.

Since we have introduced NS–SMC as a way to overcome theoretical issues in NS, particularly when there is particle dependency, in our experiments we restrict ourselves to MCMC kernels. We point out that region samplers are only a valid η_t-invariant kernel at each t if they are able to make proposals on *all* of E_t. However, as discussed earlier, one cannot guarantee this.

To give a fair comparison, we considered three choices of MCMC kernels:

1. The classic Random Walk (RW) sampler, where proposals take the form $\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{X}, h^2\widehat{\Sigma})$. This sampler was selected as it a widely applicable and common sampler.

2. The Metropolis Adjusted Langevin Algorithm (MALA), with proposals

$$\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{X} + \nabla_{\boldsymbol{x}} \log \tilde{\pi}(\boldsymbol{X}), h^2\widehat{\Sigma}),$$

   where $\tilde{\pi}$ is the target distribution. MALA is applicable when the derivatives of the log target with respect to the parameters are available analytically or can be estimated unbiasedly. One of the strengths of TA-SMC is that there is rich literature of samplers that are straightforward to apply (see [34, Ch.6] for example). We feel it important to include MALA because it is more suited to unconstrained targets. The derivatives of the log likelihood are not used in NS–SMC MALA because the likelihood is only used in defining the constraints.

3. Slice Sampling [41], specifically the slicesample function from the Statistics and Machine Learning Toolbox in MATLAB . This implementation is based on the basic stepping out and shrinkage implementation described in [41]. The step out distance in each dimension is chosen to be $w\widehat{\sigma_i}$ where $\widehat{\sigma_i}$ is the standard deviation of the ith parameter estimated from the population of particles. Unlike RW and MALA, Slice Sampling is not disadvantaged by working in a constrained space, as it requires constrained sampling regardless of the underlying distribution.

In this section, we wish to compare the two SMC methods in a setting resembling what is typically used in practice. Thus, we use the stratified resampling scheme of Kitagawa [33] for both methods as this results in lower variance over that of the simpler multinomial scheme. For recent convergence results and justification for choosing this scheme over other alternatives, see [26].

We conduct an initial pilot run to determine the sequence of distributions, before executing 100 runs with fixed choice of distributions and MCMC parameters/repeats determined from the pilot run (chosen

as described previously in Section 2.5). In the pilot run, we used $\rho = 0.5$ for NS-SMC, and $\alpha = 0.5$ for TA-SMC, as this leads to the same proportion of ESS out of the total and in our experience these choices tend to generally perform well. As the same choice of ρ and α typically yields more iterations (i.e., target distributions) for NS-SMC, in the pilot run we used $N = 4 \cdot 10^4$ and $N = 10^4$ samples for TA-SMC and NS-SMC, respectively, in order to keep the number of likelihood evaluations roughly equivalent across methods. For 100 fixed runs, we used $N = 4 \cdot 10^3$ and $N = 10^3$ samples for TA-SMC and NS-SMC, respectively. We felt it was important to use a larger number of samples in the pilot run in order to tune the MCMC kernels better, and to ensure differences in performance were not simply due to poor selection of tuning parameters.

Figure 2.4 illustrates that the typical behaviour of our methods for the selection of kernel parameter and repeats is what one should expect for TA–SMC. Specifically, as the target becomes increasingly complex, the number of repeats increases, and smaller step sizes are made. Note how TA–SMC MALA makes larger steps and therefore uses fewer repeats. Further plots for our experiments in this section can be found in Section 2.10.

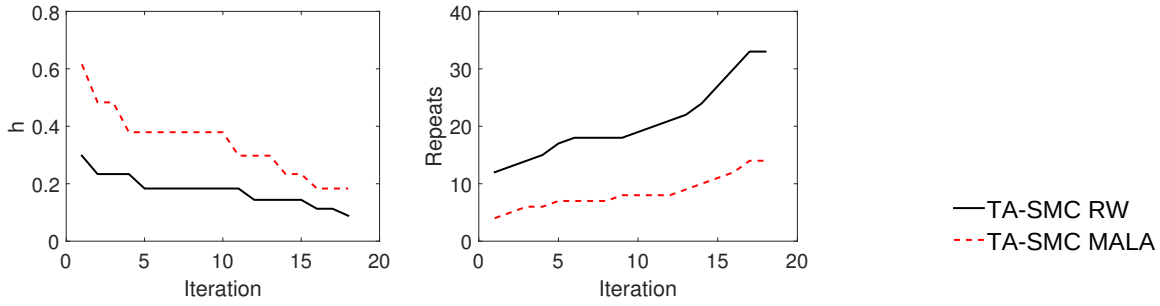


Figure 2.4: Tuning parameter and repeats selection for TA–SMC RW and TA–SMC MALA for the challenging three component factor analysis model considered in Section 2.6.

When comparing results, we examine estimates of posterior means, posterior lower (2.5%) and upper (97.5%) quantiles, and model evidence. We measure efficiency in terms of *work–normalized variance* (WNV), specifically the variance of the quantity of interest (a measure of statistical efficiency) multiplied by the number of likelihood evaluations (a measure of computational efficiency). The relative WNV measure shown in some tables is the WNV for that method divided by the WNV for TA–SMC RW. Thus, smaller values are considered evidence of superior performance.

2.6.1 Example 1: Factor Analysis

This model choice example demonstrates how NS–SMC and TA–SMC perform on three different posterior distributions of varying complexity. We consider the monthly exchange rate dataset used in [53], where exchange rates (relative to the British Pound) of six different currencies were collected from January 1975 to December 1986, for a total of $n = 143$ observations. As in [36], we model the covariance of the (standardized) monthly-differenced exchange rates, using a factor analysis model, i.e., for $k \leq d$ factors, our data is assumed to be drawn independently from a $\mathcal{N}(\mathbf{0}, \Omega)$ distribution, where Ω can be factorized as $\Omega = \beta\beta^\top + \Lambda$, for $\beta \in \mathbb{R}^{d\times k}$ lower triangular with positive diagonal elements, and Λ a diagonal matrix with diagonal given by $\boldsymbol{\lambda} \in \mathbb{R}^d_+$. Thus, we have that for each additional factor in the model, we introduce $6(k+1) - k(k-1)/2$ additional parameters, giving 12, 17, and 21 parameters for one, two and three factors, respectively. For priors, we follow [36] and specify

$$\begin{aligned}
\beta_{ij} &\sim \mathcal{N}(0,1), \quad i < j, i = 1,\ldots,k, j = 1,\ldots,d \\
\beta_{ii} &\sim \mathcal{TN}_{(0,\infty)}(0,1), \quad i = 1,\ldots,k \\
\lambda_i &\sim \mathcal{IG}(1.1, 0.05), \quad i = 1,\ldots,d,
\end{aligned}$$

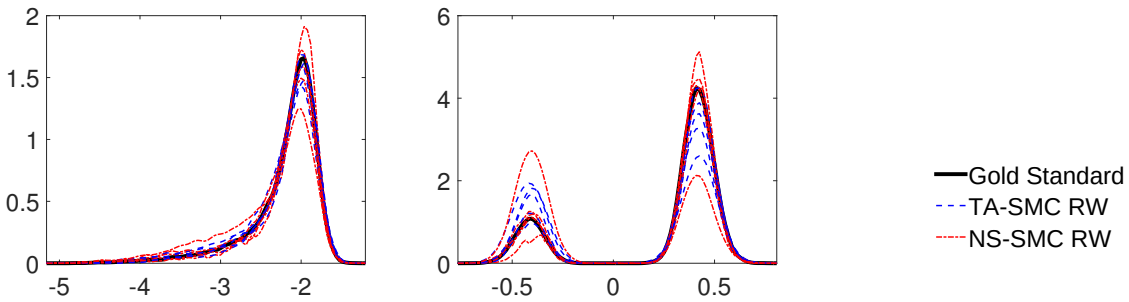
where $\mathcal{TN}_{(0,\infty)}(\boldsymbol{\mu}, \Sigma)$ denotes a $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ distribution truncated to the interval $(0, \infty)$, and $\mathcal{IG}(a, b)$ denotes the Inverse–Gamma distribution with probability density function

$$f(x) = \frac{b^a}{\Gamma(a)} x^{-a-1} \exp(-b/x), \quad x > 0.$$

In order to facilitate improved sampling, we take a log transform of β_{ii} for $i = 1,\ldots,k$, which obviates the need to deal with any constraints at all in TA–SMC.

The one factor posterior (FA1) is relatively easy to sample from in that the marginal densities are all unimodal. The two factor (FA2) posterior possesses highly separated modes that are challenging to capture for standard MCMC methods (for example, the reversible jump sampler of [36] failed to capture this). Finally, the three factor posterior (FA3) contains an exceptionally complex landscape, as shown by Figure 2.6.

We also include results from an extended "gold standard" run of TA-SMC, for which we used $N = 5 \times 10^4$, and the extremely conservative $\alpha = 0.999$ to ensure the particles adequately explored the space. We also note that $\log p$ vs. $\log \mathcal{L}$ plots do not indicate the presence of any phase transitions in any of the three cases. Results for evidence estimation are shown in Table 2.2 and Figure 2.8, and results for posterior inference are given in Appendix 2.9. It appears that RW and slice kernels are more efficient for NS–SMC than TA–SMC for both of the challenging models. Given the earlier discussion on using MALA in a constrained space, it is not surprising to see that NS–SMC has performed poorly.



(a) $\log \Lambda_{22}$ (b) β_{32}

Figure 2.5: FA2 posterior marginal estimates for the gold standard and for 5 runs of TA–SMC with a RW and NS–SMC with a RW sampler. Shown are parameters (a) $\log \Lambda_{22}$ which is highly skewed and (b) β_{32} which has well separated modes.

Table 2.2: Factor Analysis model evidence results for 100 runs. Efficiency factor is relative to TA–SMC RW.

Factors	Method	Sampler	$\log \widehat{\mathcal{Z}}$	avg. evals	relative WNV
One	TA–SMC	RW	-1014.28	7.3×10^5	1.0
		MALA	-1014.28	3.0×10^5	0.1
		Slice	-1014.27	9.5×10^5	2.6
	NS–SMC	RW	-1014.27	5.3×10^5	1.1
		MALA	-1014.24	8.2×10^5	2.2
		Slice	-1014.32	8.2×10^5	2.5
Two	TA–SMC	RW	-903.21	1.3×10^6	1.0
		MALA	-903.24	7.5×10^5	0.1
		SLICE	-903.38	1.3×10^6	2.5
	NS–SMC	RW	-903.23	1.2×10^6	0.3
		MALA	-903.02	1.9×10^6	1.9
		SLICE	-903.18	1.3×10^6	2.1
Three	TA–SMC	RW	-905.29	1.5×10^6	1.0
		MALA	-905.36	6.1×10^5	0.1
		SLICE	-905.02	1.8×10^6	11.8
	NS–SMC	RW	-905.39	1.7×10^6	0.4
		MALA	-905.40	1.4×10^6	0.7
		SLICE	-905.30	2.2×10^6	1.2

In the 2 and 3 component models, multimodality introduces additional difficulty. In SMC, the main issues with multimodality are that (a) resampling can change the proportion of particles in each mode and (b) many MCMC kernels do not correct for this by moving between modes. In NS–SMC, the constraints mean that modes become well separated quickly which compounds problem (b). On the other hand, TA–SMC suffers more from problem (a) because if resampling removes all samples from

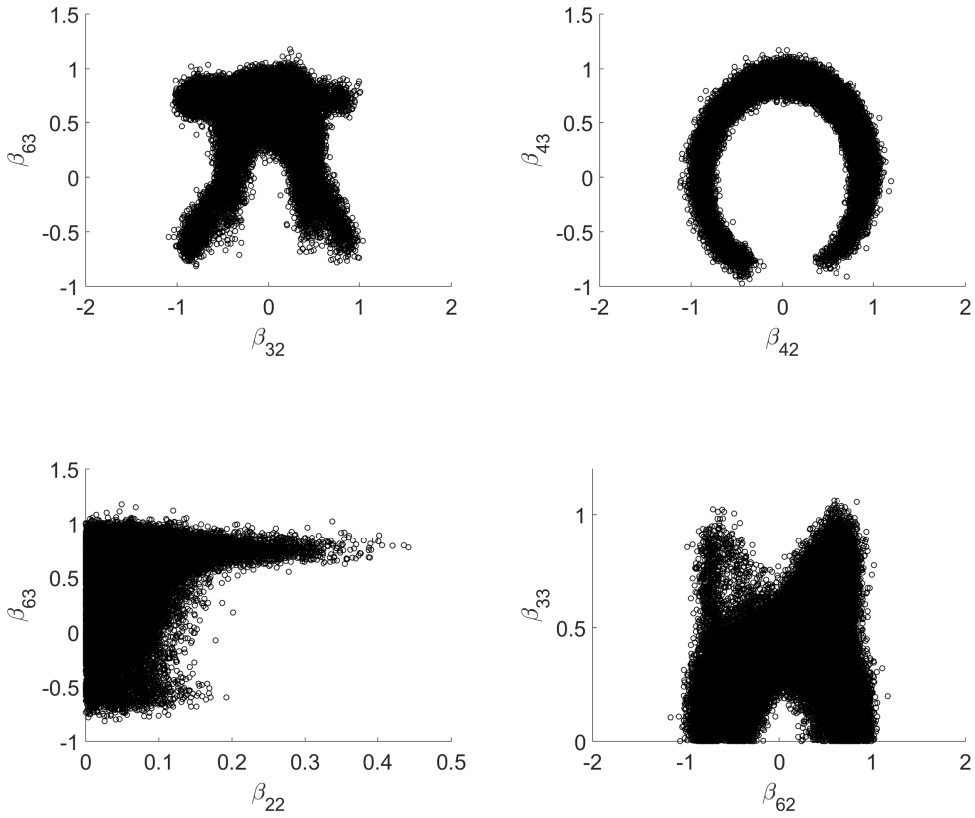
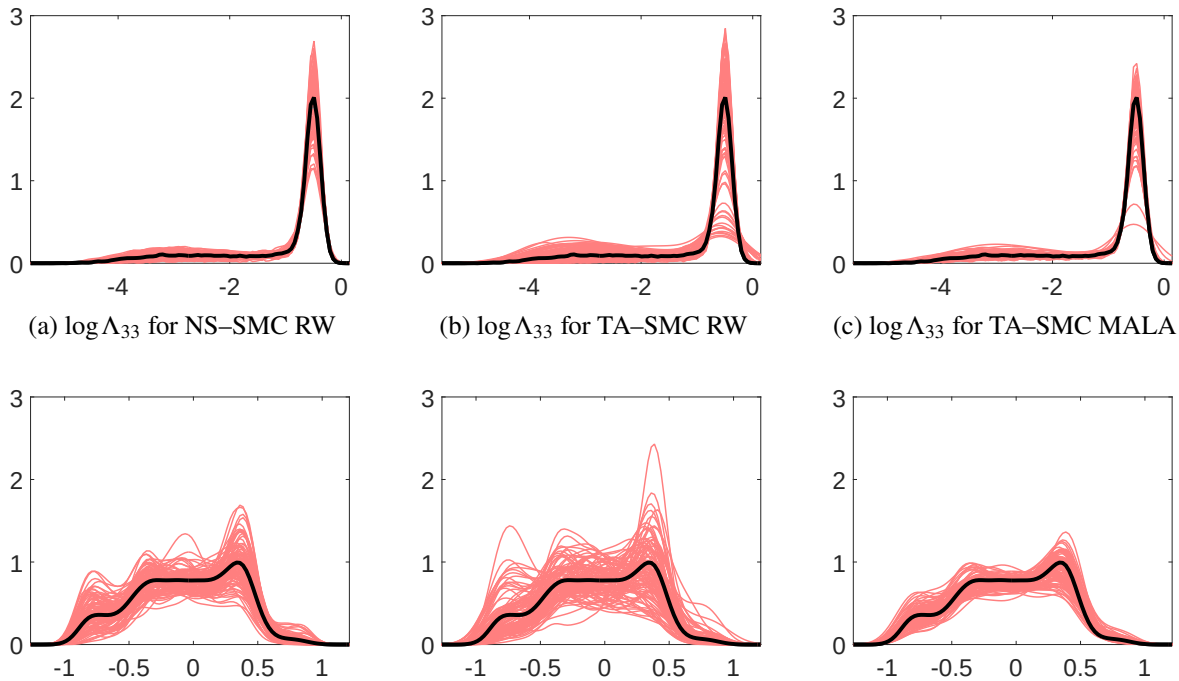


Figure 2.6: A selection of the most challenging bivariate distributions. Plots are FA3 bivariate posterior scatterplots from the gold standard run.

a given mode, then unless the unlikely event that the mode is rediscovered occurs, it will not be captured at all by the particles on the final target (even with recycling methods as described in [42], the highest weights will be on the final few targets). This may explain why one method does not seem to significantly outperform the other in the 2 and 3 component models.

2.6.2 Example 2: Ordinary Differential Equation

Models for which the posterior density exhibits strong and complicated tail dependencies present a unique challenge for samplers. Thus, it is natural to consider to what extent NS–SMC is robust to these issues by testing it on such an example.



(a) $\log \Lambda_{33}$ for NS–SMC RW (b) $\log \Lambda_{33}$ for TA–SMC RW (c) $\log \Lambda_{33}$ for TA–SMC MALA

(d) β_{32} for NS-SMC RW (e) β_{32} for TA-SMC RW (f) β_{32} for TA-SMC MALA

Figure 2.7: FA3 posterior marginal estimates for the gold standard (thick line) and for 100 runs of NS–SMC and TA–SMC (thin lines). Shown are parameters (a,b) $\log \Lambda_{33}$ which is highly skewed and (c,d) β_{32} which is multimodal.

We consider a system of ordinary differential equations for modelling biochemical pathways [27], specifically the following system of coupled ordinary differential equations (ODEs)

$$\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= -k_1 S \\
\frac{\mathrm{d}D}{\mathrm{d}t} &= k_1 S \\
\frac{\mathrm{d}R}{\mathrm{d}t} &= \frac{-V_1 RS}{Km_1 + R} + \frac{V_2 R_{pp}}{Km_2 + R_{pp}} \\
\frac{\mathrm{d}R_{pp}}{\mathrm{d}t} &= \frac{V_1 RS}{Km_1 + R} - \frac{V_2 R_{pp}}{Km_2 + R_{pp}}.
\end{aligned}$$

Following [27], Gamma priors are specified for all parameters,

$$\begin{aligned}
k_1, V_1, K_{m_1}, V_2, K_{m_2}, \sigma &\sim \mathcal{G}(1,1) \\
S(0), R(0) &\sim \mathcal{G}(5, 0.2) \\
D(0), R_{pp}(0) &\sim \mathcal{G}(1, 0.1).
\end{aligned}$$

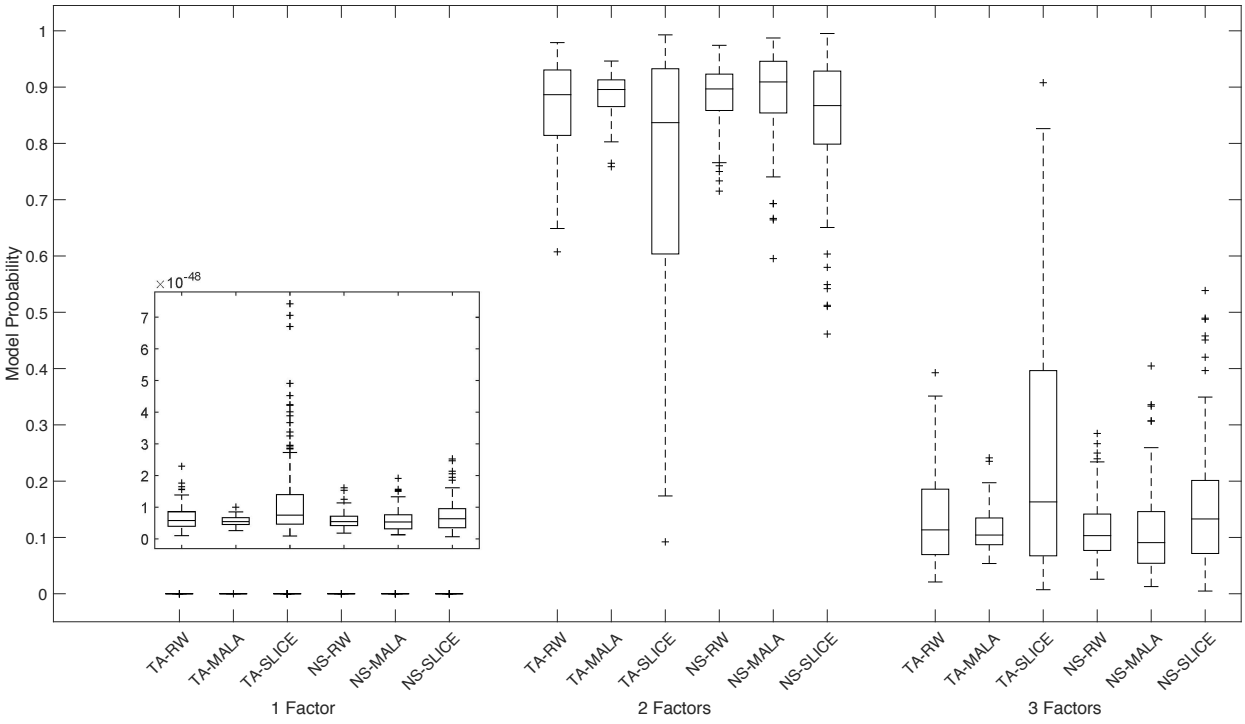


Figure 2.8: FA model probabilities based on 100 runs.

where $\mathcal{G}(\alpha, \beta)$ has the density

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x), \quad x \geq 0.$$

As in [27], we generate a synthetic dataset using

$$y(t) \sim \mathcal{N}\left(R_{pp}(t), \sigma^2\right), \quad t = 0, 3, 6, \ldots, 57,$$

where $\sigma = 0.02$, and $R_{pp}(t)$ is obtained via forward simulation of the model (this is a stiff system, so MATLAB's ODE15s solver is used) with

$$\begin{bmatrix} k_1 \\ V_1 \\ K_{m_1} \\ K_{m_2} \\ V_2 \\ S(0) \\ D(0) \\ R(0) \\ R_{pp}(0) \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.2 \\ 0.1 \\ 0.1 \\ 0.1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

We perform all sampling on a transformed space where the natural logarithm is applied element-wise to each parameter, in order to remove the need to sample on a constrained space. Despite being only a nine–dimensional parameter space, sampling from the posterior density in this example is challenging due to complex tail dependencies.

The "gold standard" for this example is a 10^7 iteration random walk MCMC run, with a burn-in of 10^5 iterations and thinning by taking every 10^3–th sample. This extended run uses roughly 5-10 times the number of likelihood evaluations as any of the SMC samplers considered here.

From Figure 2.10 and Tables 2.13 and 2.14 in Section 2.9, it can be seen that both TA–SMC and NS–SMC fall short, somewhat surprisingly, in a very similar manner with respect to tail coverage for parameters $\log k_1$, $\log Km_2$ and $\log V_2$. Observe in Figure 2.10 that the occasional run produces a disproportionate amount of samples in the tails, indicating that the failure to obtain representative samples in the tails is a largely a manifestation of high variance.

Table 2.3: ODE model evidence results for 100 runs.

Method	Sampler	$\log \widehat{\mathcal{Z}}$	avg. evals	relative WNV
TA–SMC	RW	21.98	1.3×10^6	1.0
	MALA	21.85	9.6×10^5	8.1
	SLICE	22.20	2.2×10^6	13.0
NS–SMC	RW	22.15	2.1×10^6	3.1
	MALA	22.00	8.8×10^5	1.2
	SLICE	21.97	2.0×10^6	3.8

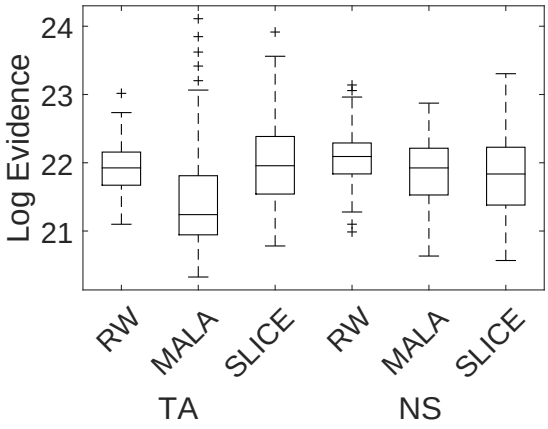
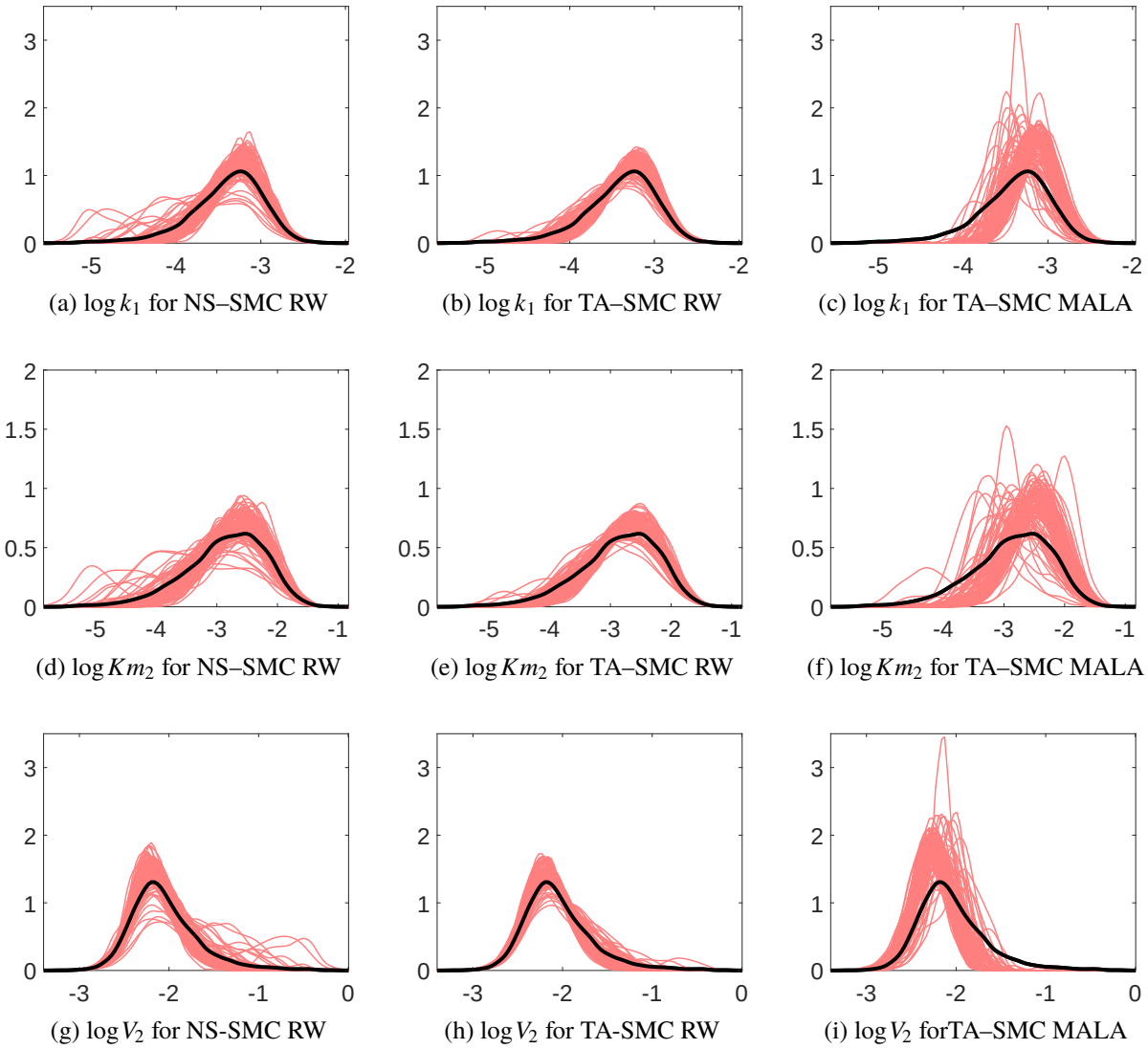


Figure 2.9: Boxplots of the log evidence for the ODE example based on 100 runs.

Once again, the choice of MCMC kernel has more of an impact on evidence and posterior estimation than the choice of SMC method. An interesting case here is TA–SMC MALA which performs poorly both in terms of evidence estimation and posterior approximation. TA–SMC MALA makes proposals which are guided by the (estimated) global covariance and the derivatives of the log target. As this does not take local dependencies into account, the use of derivative information here actually results in something that performs *worse* than the RW sampler by a significant factor. In general, one must keep in mind that the use of additional derivative information does not necessarily translate into superior performance.



(a) $\log k_1$ for NS–SMC RW (b) $\log k_1$ for TA–SMC RW (c) $\log k_1$ for TA–SMC MALA

(d) $\log Km_2$ for NS–SMC RW (e) $\log Km_2$ for TA–SMC RW (f) $\log Km_2$ for TA–SMC MALA

(g) $\log V_2$ for NS-SMC RW (h) $\log V_2$ for TA-SMC RW (i) $\log V_2$ forTA–SMC MALA

Figure 2.10: ODE posterior marginal estimates for the gold standard and for 100 runs of NS–SMC RW, TA–SMC RW and TA–SMC MALA. Shown are parameters (a,b,c) $\log k_1$ where lower tail coverage is an issue, (d,e,f) $\log Km_2$ where lower tail coverage is an issue, and (g,h,i) $\log V_2$ where upper tail coverage is an issue.

2.7 Discussion

The results of our numerical study demonstrate that the NS-SMC approach is capable of performing well on very difficult problems. The results in Section 2.6 indicate that the performance of the model evidence estimator is more a product of the performance of an MCMC kernel than of the overarching SMC method. However, as illustrated by the phase transition example in Section 2.4.3, there are problems for which NS–SMC is preferable. Such cases asides, the question whether SMC is preferable

using the TA or NS approach is really one of whether it is preferable to sample (relatively) easy distributions subject to a constraint or to sample potentially difficult distributions. Overall, our results provide evidence that the NS approach to SMC has its merits and deserves further attention.

In terms of extensions and variants to NS-SMC from the NS literature, we identify several promising directions. An analogous SMC method to the Diffusive Nested Sampling of Brewer [6] may be be possible through the use of specifying a sequence of mixtures of densities of the form (2.21). Such an approach may increase robustness to the tail coverage issues such as those in the ODE example, and would perhaps improve the performance of NS-SMC in multimodal settings. A NS-SMC version of the ellipsoidal nested importance sampling method of Chopin and Robert [13] is straightforward. We note however that as $\mathbb{P}_\eta(\boldsymbol{X} \in \breve{E}_t)$ is easily computed in this setting, and as exactly sampling from η constrained to the shells $\breve{E}_t$ is possible (as Nested Importance Sampling reformulates the problem so that η is Gaussian and $\breve{E}_t$ are ellipsoidal regions), the method reduces to a stratified form of importance sampling.

Conversely, improvements to NS-SMC may also be made by borrowing from the SMC literature. Again, there are several exciting possible directions in this regard. For example, use of the particle population at each stage to construct independence samplers as in [49], which not only are capable of providing highly effective MCMC kernels at each iteration, but have the added advantage of allowing one to recycle proposals to further improve estimates. Furthermore, with the absence of a deterministic quadrature rule, and Monte Carlo estimators in their place, NS-SMC may be improved further by control variate techniques such as zero–variance control variates [37] or control functionals [43].

In terms of theoretical developments, convergence results for ANS–SMC may be possible by extending the results of Cérou and Guyader [9] for adaptive multilevel splitting, and would require taking into account the dual importance sampling at each iteration, as well as the random termination condition. Convergence results for INS (and in turn NS with MCMC) remain difficult due to the combined adaptivity and special choice of move step, however the connection of NS to SMC provides a new way of looking at the problem.

Finally, as the performance of NS–SMC largely depends on the performance of the MCMC kernel used in the move step, further research on how to best sample from distributions subject to complicated constraints is also of interest. Such samplers are also of interest for SMC methods for Approximate Bayesian Computation (see for example, [18]).

2.8 Appendix: Theoretical Properties of Fixed NS–SMC

For the purposes of theoretical analysis, SMC algorithms can be interpreted as interacting particle approximations to a flow of associated Feynman–Kac measures. We proceed using the convention in the main text and [17] that $t \geq 1$, as opposed to $t \geq 0$. We note this point as the latter is typically used in the analysis of Feynman–Kac flows. However, the difference is simply one of presentation.

Consider the sequence of densities $\eta_1, \ldots, \eta_T$ defined in the nested manner described at the beginning of Section 2.4. Cérou et al [10, Proposition 2] show that the associated measures have the Feynman–Kac representation

$$\eta_t(f) = \mathbb{E}_{\eta_t}[f(\boldsymbol{X})] = \frac{\mathbb{E}[f(\boldsymbol{X}_t)\prod_{p=1}^{t-1}\mathbb{I}\{\boldsymbol{X}_p \in E_{p+1}\}]}{\mathbb{E}[\prod_{p=1}^{t-1}\mathbb{I}\{\boldsymbol{X}_p \in E_{p+1}\}]},$$

where f is any test function, and $(\boldsymbol{X}_p)_{p=1}^{t}$ is a Markov chain such that $\boldsymbol{X}_1 \sim \eta$. Precise details regarding the transition kernel of this time–inhomogeneous chain can be found in [10, Section 2]. However, the key aspect is that the kernel K_t that governs transitions from $\boldsymbol{X}_{t-1}$ to $\boldsymbol{X}_t$ is η_t–invariant.

For $t = 1, \ldots, T$, we thus have the *unnormalized* and *normalized Feyman–Kac measures*, given by

$$\gamma_t(\varphi) := \mathbb{E}\left[f(\boldsymbol{X}_t)\prod_{p=1}^{t-1}\mathbb{I}\{\boldsymbol{X}_p \in E_{p+1}\}\right]$$
$$\text{and } \eta_t(f) := \gamma_t(f)/\gamma_t(1),$$

respectively.

The population of particles in NS–SMC (equivalently, the fixed levels algorithm in [10]) approximate these measures with the *particle approximation* measures

$$\gamma_t^N(f) := \underbrace{\left(\prod_{p=1}^{t-1}\eta_p^N(\mathbb{I}_{E_{p+1}})\right)}_{\gamma_t^N(1)=\widehat{\mathcal{P}}_t}\underbrace{\left(\frac{1}{N}\sum_{k=1}^{N}f(\boldsymbol{X}_t^k)\right)}_{\eta_t^N(f)}, \tag{2.32}$$

and

$$\eta_t^N(\varphi) := \gamma_t^N(f)/\gamma_t^N(1).$$

Feynman–Kac particle approximation measures have the well–known properties (see for example, [15]) that for all bounded measurable f : (1) $\mathbb{E}[\gamma_t^N(f)] = \gamma_t(f)$, and (2) as $N \to \infty$, $\gamma_t^N(f) \xrightarrow{\text{a.s.}} \gamma_t(f)$ and $\eta_t^N(f) \xrightarrow{\text{a.s.}} \eta_t(f)$. These properties are often presented in the context of multinomial resampling.

However, they also hold for other resampling schemes that satisfy certain mild conditions; see Chapter 11.8 of [14].

We have that

$$\mathcal{P}_t = \gamma_t(1) = \prod_{p=1}^{t-1} \eta_p(\mathbb{I}_{E_{p+1}}) \quad \text{and} \quad \mathcal{Z}_t = \gamma_t(\mathcal{L}\mathbb{I}_{\check{E}_t}) = \mathcal{P}_t \underbrace{\eta_t(\mathcal{L}\mathbb{I}_{\check{E}_t})}_{\mathcal{Z}_t/\mathcal{P}_t}.$$

Henceforth, we proceed under the assumption that $\mathcal{L}$ is bounded. As a result of Property (1), it follows that the estimators

$$\widehat{\mathcal{Z}}_t = \gamma_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t}) = \underbrace{\left(\prod_{p=1}^{t-1} \eta_p^N(\mathbb{I}_{E_{p+1}})\right)}_{\widehat{\mathcal{P}}_t} \underbrace{\eta_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t})}_{\widehat{\mathcal{Z}_t/\mathcal{P}_t}}, \qquad \text{for } t = 1, \ldots, T,$$

are unbiased. By linearity of expectation, it follows that $\widehat{\mathcal{Z}} = \sum_{t=1}^T \widehat{\mathcal{Z}}_t$ is an unbiased estimator of $\mathcal{Z} = \sum_{t=1}^T \mathcal{Z}_t$. By Property (2), we have that $\widehat{\mathcal{Z}}_t \xrightarrow{\text{a.s.}} \mathcal{Z}_t$, for $t = 1, \ldots, T$, and thus $\widehat{\mathcal{Z}} \xrightarrow{\text{a.s.}} \mathcal{Z}$.

The NS-SMC estimator for $\pi(\varphi)$ is based on the simple identity:

$$\pi(\varphi) = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\mathcal{Z}} \pi_t(\varphi) = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\sum_{s=1}^T \mathcal{Z}_s} \cdot \frac{\eta_t(\mathcal{L}\mathbb{I}_{\check{E}_t}\varphi)}{\eta_t(\mathcal{L}\mathbb{I}_{\check{E}_t})},$$

which we approximate via

$$\pi^N(\varphi) = \sum_{t=1}^T \frac{\widehat{\mathcal{Z}}_t}{\sum_{s=1}^T \widehat{\mathcal{Z}}_s} \cdot \frac{\eta_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t}\varphi)}{\eta_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t})}.$$

Combining the almost–sure convergence of $\widehat{\mathcal{Z}}_1, \ldots, \widehat{\mathcal{Z}}_T$ with Property (2), as $N \to \infty$ we have that $\pi^N(\varphi) \xrightarrow{\text{a.s.}} \pi(\varphi)$ for any bounded measurable function φ.

Bibliography

[1] R. J. N. Baldock. *Classical Statistical Mechanics with Nested Sampling*. Springer theses. Springer, Cham, 2017.

[2] A. Beskos, A. Jasra, N. Kantas, and A. Thiery. On the convergence of adaptive sequential Monte Carlo methods. *The Annals of Applied Probability*, 26(2):1111–1146, 2016.

[3] Z. Botev and D. P. Kroese. An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. *Methodology and Computing in Applied Probability*, 10(4):471–505, December 2008.

[4] Z. I. Botev and D. P. Kroese. Efficient Monte Carlo simulation via the generalized splitting method. *Statistics and Computing*, 22(1):1–16, Jan 2012.

[5] B. J. Brewer. Inference for trans-dimensional Bayesian models with diffusive nested sampling. *arXiv:1411.3921*, 2014.

[6] B. J. Brewer, L. B. Pártay, and G. Csányi. Diffusive nested sampling. *Statistics and Computing*, 21(4):649–656, Oct 2011.

[7] J. Buchner. A statistical test for nested sampling algorithms. *Statistics and Computing*, 26(1):383–392, Jan 2016.

[8] N. Burkoff, C. Várnai, S. A. Wells, and D. L. Wild. Exploring the energy landscapes of protein folding simulations with Bayesian computation. *Biophysical Journal*, 102(4):878–886, February 2012.

[9] F. Cérou and A. Guyader. Fluctuation analysis of adaptive multilevel splitting. *Ann. Appl. Probab.*, 26(6):3319–3380, December 2016.

[10] F. Cérou, P. Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22(3):795–808, May 2012.

[11] N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, aug 2002.

[12] N. Chopin and J. Ridgway. Leave Pima indians alone: Binary regression as a benchmark for Bayesian computation. *Statist. Sci.*, 32(1):64–87, February 2017.

[13] N. Chopin and C. P. Robert. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

[14] P. Del Moral. *Feynman-Kac Formulae Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, New York, NY, 2004.

[15] P. Del Moral. *Mean field simulation for Monte Carlo integration*. Monographs on statistics & applied probability 126. Taylor & Francis, Boca Raton, 2013.

[16] P. Del Moral and A. Doucet. Particle methods: an introduction with applications. *ESAIM: Proceedings*, 44:1–46, January 2014.

[17] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[18] P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, September 2012.

[19] P. Del Moral, A. Doucet, and A. Jasra. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, February 2012.

[20] A. Doucet and A. M. Johansen. A Tutorial on Particle filtering and smoothing: Fiteen years later. *The Oxford handbook of nonlinear filtering*, (December 2008):656–705, 2011.

[21] C. C. Drovandi and A. N. Pettitt. Likelihood-free Bayesian estimation of multivariate quantile distributions. *Computational Statistics and Data Analysis*, 55(9):2541–2556, 2011.

[22] M. Evans. Bayesian statistics 8. chapter Discussion of Nested sampling for Bayesian computations by John Skilling, pages 491–524. Oxford University Press, New York, 2007.

[23] P. Fearnhead and B. M. Taylor. An adaptive sequential Monte Carlo sampler. *Bayesian analysis*, 8(2):411–438, 2013.

[24] F. Feroz and M. P. Hobson. Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses. *Monthly Notices of the Royal Astronomical Society*, 384(2):449–463, 2008.

[25] N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, July 2008.

[26] M. Gerber, N. Chopin, and N. Whiteley. Negative association, ordering and convergence of resampling methods. *arXiv:1707.01845*, pages 1–24, July 2017.

[27] M. Girolami. Bayesian inference for differential equations. *Theoretical Computer Science*, 408(1):4–16, 2008.

[28] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[29] A. Guyader, N. Hengartner, and E. Matzner-Løber. Simulation and estimation of extreme quantiles and extreme probabilities. *Applied Mathematics & Optimization*, 64(2):171–196, Oct 2011.

[30] E. Higson, W. Handley, M. Hobson, and A. Lasenby. Sampling errors in nested sampling parameter estimation. *Bayesian Analysis*, pages 1–24, 2018. (Advance Publication).

[31] A. Jasra, D. A. Stephens, A. Doucet, and T. Tsagaris. Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics*, 38(1):1–22, March 2011.

[32] H. Kahn and T. E. Harris. Estimation of particle Transmission by Random Sampling. *National Bureau of Standards Applied Mathematics Series*, 12:27–30, 1951.

[33] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[34] D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo methods*. John Wiley and Sons, New York, 2011.

[35] J. S. Liu. *Monte Carlo Strategies in Scientific Computing by Jun S. Liu.* Springer Series in Statistics. Springer New York : Imprint: Springer, New York, NY, 2001.

[36] H. F. Lopes and M. West. Bayesian model assessment in factor analysis. *Statistica Sinica*, 14(1):41–67, January 2004.

[37] A. Mira, R. Solgi, and D. Imparato. Zero variance Markov chain Monte Carlo for Bayesian estimators. *Statistics and Computing*, 23(5):653–662, September 2013.

[38] I. Murray. *Advances in Markov chain Monte Carlo methods*. PhD thesis, Gatsby computational neuroscience unit, University College London, 2007.

[39] I. Murray, D. J. C. MacKay, Z. Ghahramani, and J. Skilling. Nested sampling for Potts models. pages 947–954, 2005.

[40] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[41] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–741, June 2003.

[42] T. L. T. Nguyen, F. Septier, G. W. Peters, and Y. Delignon. Improving SMC sampler estimate by recycling all past simulated particles. In *Statistical Signal Processing (SSP), 2014 IEEE Workshop on*, pages 117–120. IEEE, 2014.

[43] C. J. Oates, M. Girolami, and N. Chopin. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, June 2017.

[44] C. Pasarica and A. Gelman. Adaptively scaling the Metropolis Hastings algorithm using expected squared jumped distance. *Statistica Sinica*, 20(1):343–364, 2010.

[45] J. Ridgway. Computation of Gaussian orthant probabilities in high dimension. *Statistics and Computing*, 26(4):899–916, Jul 2016.

[46] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 2 edition, 2004.

[47] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, 3 edition, 2017.

[48] J. Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–859, 12 2006.

[49] L. South, A. N. Pettitt, and C. Drovandi. Sequential Monte Carlo for Static Bayesian Models with Independent MCMC Proposals. *QUT ePrints*, pages 1–52, 2016.

[50] S. Vegetti and L. V. E. Koopmans. Bayesian strong gravitational-lens modelling on adaptive grids: objective detection of mass substructure in galaxies. *Monthly Notices of the Royal Astronomical Society*, 392(3):945–963, January 2009.

[51] J. Veitch. Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library. *Physical Review D (Particles, Fields, Gravitation and Cosmology)*, 91(4), February 2015.

[52] C. Walter. Point process-based Monte Carlo estimation. *Statistics and Computing*, 27(1):219–236, January 2017.

[53] M. West and J. Harrison. *Bayesian forecasting and dynamic models*. Springer series in statistics. Springer, New York, 2nd edition edition, 1997.

Supplementary Material for Part I

2.9 Inference Results

We present posterior expectation and quantile results from 100 runs, $N = 4 \cdot 10^3$ for TA–SMC and $N = 10^3$ for NS–SMC for the Factor Analysis and ODE examples. In this section and the next, we refer to the one, two, and three component factor analysis models as FA1, FA2, and FA3, respectively.

In brackets we report the ratio of (sample variance $\times$ average number of evaluations of $\mathcal{L}$) for the associated method to that of TA–SMC with the Random Walk Sampler. Thus, lower values indicate lower work–normalized variance.

Table 2.4: Inference Results for FA1 — Part 1 of 2

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \Lambda_{11}$	Gold standard		−0.23	−0.46	0.01
	TA–SMC	RW	−0.23(1.0)	−0.47(1.0)	0.01(1.0)
		MALA	−0.23(0.2)	−0.46(0.3)	0.01(0.4)
		SLICE	−0.23(4.5)	−0.47(1.4)	0.01(2.1)
	NS–SMC	RW	−0.23(1.3)	−0.47(0.5)	0.01(1.1)
		MALA	−0.24(3.7)	−0.47(1.3)	0.01(2.2)
		SLICE	−0.23(3.2)	−0.47(1.2)	0.01(1.7)
$\log \Lambda_{22}$	Gold standard		−0.25	−0.48	−0.00
	TA–SMC	RW	−0.25(1.0)	−0.48(1.0)	−0.00(1.0)
		MALA	−0.25(0.2)	−0.48(0.3)	−0.00(0.2)
		SLICE	−0.25(3.4)	−0.48(1.3)	−0.00(1.3)
	NS–SMC	RW	−0.24(1.3)	−0.48(0.7)	−0.00(0.7)
		MALA	−0.25(1.7)	−0.48(1.0)	−0.01(1.0)
		SLICE	−0.25(4.6)	−0.48(1.2)	−0.00(1.5)
$\log \Lambda_{33}$	Gold standard		−0.43	−0.66	−0.19
	TA–SMC	RW	−0.43(1.0)	−0.66(1.0)	−0.18(1.0)
		MALA	−0.43(0.2)	−0.66(0.3)	−0.18(0.2)
		SLICE	−0.43(3.0)	−0.66(1.6)	−0.18(1.5)
	NS–SMC	RW	−0.43(1.0)	−0.66(0.6)	−0.19(0.6)
		MALA	−0.43(3.7)	−0.66(1.7)	−0.19(1.2)
		SLICE	−0.43(4.0)	−0.66(1.4)	−0.18(1.7)
$\log \Lambda_{44}$	Gold standard		−2.65	−3.51	−2.05
	TA–SMC	RW	−2.65(1.0)	−3.51(1.0)	−2.04(1.0)
		MALA	−2.66(0.1)	−3.52(0.2)	−2.05(0.3)
		SLICE	−2.65(4.4)	−3.51(2.2)	−2.05(2.2)
	NS–SMC	RW	−2.66(1.2)	−3.52(0.9)	−2.05(0.7)
		MALA	−2.66(3.5)	−3.52(2.4)	−2.05(1.3)
		SLICE	−2.65(3.6)	−3.51(2.4)	−2.05(1.4)
$\log \Lambda_{55}$	Gold standard		−1.45	−1.73	−1.17
	TA–SMC	RW	−1.45(1.0)	−1.73(1.0)	−1.16(1.0)
		MALA	−1.45(0.2)	−1.73(0.4)	−1.16(0.3)
		SLICE	−1.45(3.1)	−1.73(1.8)	−1.16(1.9)
	NS–SMC	RW	−1.45(1.2)	−1.73(0.9)	−1.17(0.8)
		MALA	−1.45(2.6)	−1.73(1.7)	−1.16(1.1)
		SLICE	−1.45(3.4)	−1.73(1.9)	−1.16(1.4)
$\log \Lambda_{66}$	Gold standard		−1.44	−1.73	−1.16
	TA–SMC	RW	−1.43(1.0)	−1.72(1.0)	−1.15(1.0)
		MALA	−1.44(0.2)	−1.73(0.4)	−1.15(0.3)
		SLICE	−1.44(4.5)	−1.73(1.5)	−1.15(1.7)
	NS–SMC	RW	−1.44(1.5)	−1.73(1.0)	−1.15(0.8)
		MALA	−1.44(3.6)	−1.73(1.5)	−1.16(1.5)
		SLICE	−1.44(5.3)	−1.72(1.5)	−1.15(1.7)

Table 2.5: Inference Results for FA1 — Part 2 of 2

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \beta_{11}$	Gold standard		−0.81	−1.24	−0.48
	TA–SMC	RW	−0.81(1.0)	−1.24(1.0)	−0.48(1.0)
		MALA	−0.81(0.2)	−1.23(0.3)	−0.48(0.3)
		SLICE	−0.81(4.1)	−1.23(2.5)	−0.48(2.0)
	NS–SMC	RW	−0.82(1.1)	−1.24(1.2)	−0.48(0.7)
		MALA	−0.81(3.2)	−1.23(2.3)	−0.48(1.3)
		SLICE	−0.81(4.2)	−1.24(2.5)	−0.48(1.3)
β_{21}	Gold standard		0.46	0.30	0.63
	TA–SMC	RW	0.46(1.0)	0.30(1.0)	0.63(1.0)
		MALA	0.46(0.3)	0.30(0.3)	0.63(0.4)
		SLICE	0.46(3.4)	0.30(1.6)	0.63(2.1)
	NS–SMC	RW	0.46(1.8)	0.30(0.8)	0.63(1.0)
		MALA	0.46(3.6)	0.30(1.1)	0.63(1.5)
		SLICE	0.46(4.5)	0.30(1.3)	0.63(2.3)
β_{31}	Gold standard		0.59	0.44	0.75
	TA–SMC	RW	0.59(1.0)	0.44(1.0)	0.75(1.0)
		MALA	0.59(0.2)	0.44(0.3)	0.75(0.3)
		SLICE	0.59(4.9)	0.44(2.2)	0.75(2.2)
	NS–SMC	RW	0.59(1.6)	0.44(0.7)	0.75(1.2)
		MALA	0.59(4.1)	0.44(1.2)	0.75(2.0)
		SLICE	0.59(7.3)	0.44(2.0)	0.75(3.2)
β_{41}	Gold standard		0.97	0.86	1.10
	TA–SMC	RW	0.97(1.0)	0.85(1.0)	1.10(1.0)
		MALA	0.97(0.2)	0.86(0.4)	1.11(0.2)
		SLICE	0.97(5.7)	0.86(2.4)	1.10(1.9)
	NS–SMC	RW	0.97(1.3)	0.86(0.8)	1.10(0.7)
		MALA	0.97(3.2)	0.86(1.4)	1.11(1.3)
		SLICE	0.97(8.4)	0.86(2.9)	1.10(2.5)
β_{51}	Gold standard		0.88	0.76	1.02
	TA–SMC	RW	0.88(1.0)	0.76(1.0)	1.02(1.0)
		MALA	0.88(0.2)	0.76(0.3)	1.02(0.3)
		SLICE	0.88(6.3)	0.76(2.5)	1.02(1.8)
	NS–SMC	RW	0.88(1.7)	0.76(0.8)	1.02(0.9)
		MALA	0.88(2.7)	0.76(1.2)	1.02(1.6)
		SLICE	0.88(8.4)	0.76(2.4)	1.02(2.7)
β_{61}	Gold standard		0.88	0.76	1.02
	TA–SMC	RW	0.88(1.0)	0.75(1.0)	1.02(1.0)
		MALA	0.88(0.2)	0.75(0.4)	1.02(0.4)
		SLICE	0.88(5.2)	0.75(2.5)	1.02(2.3)
	NS–SMC	RW	0.88(1.2)	0.75(0.9)	1.02(0.8)
		MALA	0.88(3.3)	0.76(2.0)	1.02(1.4)
		SLICE	0.88(7.6)	0.75(2.7)	1.02(3.3)

Table 2.6: Inference Results for FA2 — Part 1 of 3

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \Lambda_{11}$	Gold standard		−3.08	−4.41	−1.91
	TA–SMC	RW	−3.05(1.0)	−4.38(1.0)	−1.91(1.0)
		MALA	−3.07(0.3)	−4.39(0.3)	−1.91(0.3)
		SLICE	−2.94(7.4)	−4.10(6.6)	−1.97(13.0)
	NS–SMC	RW	−3.07(0.4)	−4.41(0.4)	−1.90(0.5)
		MALA	−3.05(4.1)	−4.30(3.9)	−1.93(5.6)
		SLICE	−3.03(3.6)	−4.25(3.2)	−1.95(6.6)
$\log \Lambda_{22}$	Gold standard		−2.20	−3.50	−1.70
	TA–SMC	RW	−2.22(1.0)	−3.49(1.0)	−1.70(1.0)
		MALA	−2.21(0.2)	−3.48(0.4)	−1.70(0.3)
		SLICE	−2.29(10.8)	−3.31(5.5)	−1.73(22.6)
	NS–SMC	RW	−2.22(0.4)	−3.53(0.6)	−1.70(0.6)
		MALA	−2.23(3.5)	−3.48(5.1)	−1.71(9.4)
		SLICE	−2.23(3.5)	−3.38(3.9)	−1.72(8.0)
$\log \Lambda_{33}$	Gold standard		−0.48	−0.70	−0.24
	TA–SMC	RW	−0.48(1.0)	−0.71(1.0)	−0.24(1.0)
		MALA	−0.48(0.6)	−0.71(0.4)	−0.24(0.6)
		SLICE	−0.48(56.6)	−0.71(25.3)	−0.23(34.0)
	NS–SMC	RW	−0.48(1.3)	−0.71(1.1)	−0.24(1.4)
		MALA	−0.47(11.7)	−0.70(8.4)	−0.23(7.8)
		SLICE	−0.48(23.0)	−0.70(18.9)	−0.24(18.1)
$\log \Lambda_{44}$	Gold standard		−3.45	−4.49	−2.54
	TA–SMC	RW	−3.44(1.0)	−4.49(1.0)	−2.53(1.0)
		MALA	−3.44(0.3)	−4.47(0.4)	−2.53(0.4)
		SLICE	−3.36(16.6)	−4.26(14.8)	−2.52(22.2)
	NS–SMC	RW	−3.44(0.3)	−4.49(0.5)	−2.54(0.6)
		MALA	−3.45(7.6)	−4.43(9.7)	−2.56(7.3)
		SLICE	−3.41(7.5)	−4.38(8.7)	−2.53(9.3)
$\log \Lambda_{55}$	Gold standard		−1.39	−1.65	−1.13
	TA–SMC	RW	−1.39(1.0)	−1.65(1.0)	−1.12(1.0)
		MALA	−1.39(0.6)	−1.65(0.5)	−1.12(0.6)
		SLICE	−1.40(39.1)	−1.66(23.8)	−1.13(21.2)
	NS–SMC	RW	−1.38(1.4)	−1.64(1.3)	−1.12(1.8)
		MALA	−1.39(8.9)	−1.65(5.9)	−1.13(7.1)
		SLICE	−1.39(16.6)	−1.65(11.2)	−1.13(13.4)
$\log \Lambda_{66}$	Gold standard		−1.37	−1.63	−1.10
	TA–SMC	RW	−1.37(1.0)	−1.63(1.0)	−1.10(1.0)
		MALA	−1.37(0.4)	−1.63(0.3)	−1.11(0.6)
		SLICE	−1.37(28.0)	−1.64(18.6)	−1.11(18.3)
	NS–SMC	RW	−1.37(1.0)	−1.63(1.2)	−1.11(1.3)
		MALA	−1.37(7.1)	−1.63(4.6)	−1.10(7.0)
		SLICE	−1.37(15.2)	−1.63(10.6)	−1.11(14.4)

Table 2.7: Inference Results for FA2 — Part 2 of 3

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \beta_{11}$	Gold standard		−0.02	−0.15	0.11
	TA–SMC	RW	−0.02(1.0)	−0.15(1.0)	0.11(1.0)
		MALA	−0.02(0.3)	−0.15(0.3)	0.11(0.5)
		SLICE	−0.02(29.2)	−0.14(16.4)	0.11(32.8)
	NS–SMC	RW	−0.02(0.7)	−0.15(0.8)	0.11(1.6)
		MALA	−0.02(6.0)	−0.15(7.3)	0.11(7.8)
		SLICE	−0.02(14.0)	−0.14(6.8)	0.11(22.1)
β_{21}	Gold standard		0.95	0.83	1.09
	TA–SMC	RW	0.95(1.0)	0.83(1.0)	1.09(1.0)
		MALA	0.95(0.4)	0.83(0.4)	1.09(0.4)
		SLICE	0.96(25.3)	0.84(27.1)	1.09(24.3)
	NS–SMC	RW	0.95(0.8)	0.83(0.9)	1.09(1.2)
		MALA	0.95(4.9)	0.83(6.1)	1.09(6.3)
		SLICE	0.95(15.6)	0.83(12.5)	1.09(13.5)
β_{31}	Gold standard		0.45	0.30	0.62
	TA–SMC	RW	0.45(1.0)	0.30(1.0)	0.62(1.0)
		MALA	0.45(0.4)	0.30(0.4)	0.62(0.4)
		SLICE	0.46(28.7)	0.30(26.3)	0.62(24.6)
	NS–SMC	RW	0.46(0.8)	0.30(1.0)	0.62(1.2)
		MALA	0.45(8.3)	0.30(6.5)	0.62(7.2)
		SLICE	0.46(20.9)	0.30(14.1)	0.62(18.8)
β_{41}	Gold standard		0.39	0.23	0.56
	TA–SMC	RW	0.39(1.0)	0.23(1.0)	0.56(1.0)
		MALA	0.39(0.4)	0.23(0.4)	0.56(0.3)
		SLICE	0.40(22.9)	0.23(23.0)	0.56(22.4)
	NS–SMC	RW	0.40(0.7)	0.23(1.0)	0.56(0.9)
		MALA	0.39(5.7)	0.23(6.1)	0.56(4.9)
		SLICE	0.39(18.6)	0.24(15.3)	0.56(15.3)
β_{51}	Gold standard		0.41	0.25	0.58
	TA–SMC	RW	0.41(1.0)	0.25(1.0)	0.58(1.0)
		MALA	0.41(0.4)	0.25(0.5)	0.58(0.3)
		SLICE	0.42(22.6)	0.25(22.3)	0.58(17.6)
	NS–SMC	RW	0.41(0.8)	0.25(1.1)	0.58(0.8)
		MALA	0.41(5.3)	0.25(5.3)	0.58(3.2)
		SLICE	0.41(18.6)	0.26(14.4)	0.58(13.0)
β_{61}	Gold standard		0.41	0.25	0.57
	TA–SMC	RW	0.41(1.0)	0.25(1.0)	0.58(1.0)
		MALA	0.41(0.4)	0.25(0.3)	0.57(0.4)
		SLICE	0.41(25.6)	0.25(22.4)	0.57(27.1)
	NS–SMC	RW	0.41(0.6)	0.25(1.0)	0.58(1.0)
		MALA	0.41(5.9)	0.25(6.2)	0.57(5.3)
		SLICE	0.41(19.3)	0.25(14.9)	0.57(19.0)

Table 2.8: Inference Results for FA2 — Part 3 of 3

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \beta_{22}$	Gold standard		−3.54	−6.34	−2.21
	TA–SMC	RW	−3.57(1.0)	−6.06(1.0)	−2.23(1.0)
		MALA	−3.53(0.3)	−6.17(0.6)	−2.22(0.3)
		SLICE	−3.46(6.0)	−5.15(2.7)	−2.26(11.4)
	NS–SMC	RW	−3.57(0.4)	−6.34(1.0)	−2.22(0.4)
		MALA	−3.50(4.6)	−5.49(2.5)	−2.26(11.3)
		SLICE	−3.49(3.8)	−5.39(1.9)	−2.25(7.5)
β_{32}	Gold standard		0.25	−0.50	0.56
	TA–SMC	RW	0.22(1.0)	−0.47(1.0)	0.56(1.0)
		MALA	0.25(0.2)	−0.49(0.0)	0.56(0.2)
		SLICE	0.16(3.1)	−0.37(6.5)	0.54(75.5)
	NS–SMC	RW	0.22(0.5)	−0.49(0.4)	0.56(0.5)
		MALA	0.23(2.2)	−0.42(5.6)	0.56(4.1)
		SLICE	0.22(1.5)	−0.41(4.6)	0.56(1.7)
β_{42}	Gold standard		0.55	−0.97	1.03
	TA–SMC	RW	0.47(1.0)	−0.93(1.0)	1.03(1.0)
		MALA	0.54(0.2)	−0.96(0.0)	1.03(0.2)
		SLICE	0.35(3.0)	−0.68(7.1)	0.99(658.8)
	NS–SMC	RW	0.47(0.5)	−0.95(0.5)	1.03(0.3)
		MALA	0.49(2.2)	−0.80(6.1)	1.03(4.3)
		SLICE	0.46(1.5)	−0.77(5.1)	1.03(2.4)
β_{52}	Gold standard		0.46	−0.84	0.90
	TA–SMC	RW	0.40(1.0)	−0.80(1.0)	0.90(1.0)
		MALA	0.46(0.2)	−0.83(0.0)	0.90(0.2)
		SLICE	0.29(3.1)	−0.60(7.0)	0.86(482.9)
	NS–SMC	RW	0.40(0.5)	−0.82(0.5)	0.90(0.7)
		MALA	0.42(2.2)	−0.69(6.0)	0.90(4.9)
		SLICE	0.39(1.5)	−0.67(5.0)	0.90(2.4)
β_{62}	Gold standard		0.46	−0.84	0.90
	TA–SMC	RW	0.40(1.0)	−0.80(1.0)	0.90(1.0)
		MALA	0.46(0.2)	−0.83(0.0)	0.90(0.2)
		SLICE	0.29(3.1)	−0.60(7.0)	0.86(405.4)
	NS–SMC	RW	0.39(0.5)	−0.82(0.5)	0.89(0.5)
		MALA	0.41(2.2)	−0.69(6.0)	0.89(3.5)
		SLICE	0.39(1.5)	−0.67(5.0)	0.90(2.0)

Table 2.9: Inference Results for FA3 — Part 1 of 4

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \Lambda_{11}$	Gold standard		−2.89	−4.31	−1.86
	TA–SMC	RW	−2.89(1.0)	−4.25(1.0)	−1.86(1.0)
		MALA	−2.88(0.1)	−4.28(0.1)	−1.86(0.1)
		SLICE	−2.84(5.0)	−4.14(3.4)	−1.89(11.0)
	NS–SMC	RW	−2.88(0.4)	−4.30(0.5)	−1.86(0.6)
		MALA	−2.87(0.9)	−4.28(0.8)	−1.86(1.1)
		SLICE	−2.87(2.6)	−4.23(2.0)	−1.86(4.0)
$\log \Lambda_{22}$	Gold standard		−2.38	−3.88	−1.73
	TA–SMC	RW	−2.38(1.0)	−3.82(1.0)	−1.73(1.0)
		MALA	−2.38(0.1)	−3.84(0.1)	−1.73(0.1)
		SLICE	−2.43(6.8)	−3.71(4.7)	−1.76(10.7)
	NS–SMC	RW	−2.39(0.5)	−3.85(0.8)	−1.73(0.4)
		MALA	−2.41(1.1)	−3.84(1.1)	−1.74(1.0)
		SLICE	−2.40(3.2)	−3.84(3.4)	−1.74(3.9)
$\log \Lambda_{33}$	Gold standard		−1.08	−3.84	−0.27
	TA–SMC	RW	−1.15(1.0)	−3.63(1.0)	−0.28(1.0)
		MALA	−1.12(0.1)	−3.81(0.1)	−0.27(0.1)
		SLICE	−1.09(2.3)	−3.21(3.4)	−0.28(4.9)
	NS–SMC	RW	−1.10(0.4)	−3.75(0.4)	−0.27(0.3)
		MALA	−1.11(0.6)	−3.63(1.1)	−0.28(1.1)
		SLICE	−1.11(1.0)	−3.68(1.2)	−0.27(1.0)
$\log \Lambda_{44}$	Gold standard		−3.17	−4.42	−1.96
	TA–SMC	RW	−3.19(1.0)	−4.41(1.0)	−2.01(1.0)
		MALA	−3.18(0.1)	−4.41(0.1)	−1.99(0.1)
		SLICE	−3.13(4.9)	−4.28(6.6)	−2.05(4.2)
	NS–SMC	RW	−3.18(0.5)	−4.42(0.9)	−1.99(0.4)
		MALA	−3.20(1.0)	−4.43(0.8)	−2.05(2.1)
		SLICE	−3.16(2.1)	−4.37(4.1)	−2.00(1.8)
$\log \Lambda_{55}$	Gold standard		−1.77	−3.80	−1.16
	TA–SMC	RW	−1.77(1.0)	−3.68(1.0)	−1.16(1.0)
		MALA	−1.76(0.1)	−3.74(0.1)	−1.16(0.1)
		SLICE	−1.80(4.6)	−3.40(3.7)	−1.17(6.0)
	NS–SMC	RW	−1.75(0.4)	−3.69(0.5)	−1.16(0.3)
		MALA	−1.73(1.6)	−3.35(2.7)	−1.16(2.3)
		SLICE	−1.77(2.1)	−3.62(2.1)	−1.16(2.1)
$\log \Lambda_{66}$	Gold standard		−1.75	−3.73	−1.14
	TA–SMC	RW	−1.74(1.0)	−3.56(1.0)	−1.14(1.0)
		MALA	−1.73(0.1)	−3.64(0.1)	−1.15(0.1)
		SLICE	−1.75(3.9)	−3.28(3.3)	−1.15(5.5)
	NS–SMC	RW	−1.74(0.5)	−3.64(0.5)	−1.14(0.5)
		MALA	−1.74(1.7)	−3.47(2.5)	−1.14(1.6)
		SLICE	−1.74(1.8)	−3.45(2.0)	−1.15(1.7)

Table 2.10: Inference Results for FA3 — Part 2 of 4

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \beta_{11}$	Gold standard		−0.02	−0.16	0.11
	TA–SMC	RW	−0.02(1.0)	−0.16(1.0)	0.11(1.0)
		MALA	−0.02(0.1)	−0.16(0.1)	0.11(0.2)
		SLICE	−0.03(10.7)	−0.16(7.5)	0.10(11.9)
	NS–SMC	RW	−0.02(1.0)	−0.16(1.3)	0.11(1.4)
		MALA	−0.02(1.3)	−0.16(1.4)	0.11(1.3)
		SLICE	−0.03(7.3)	−0.16(6.7)	0.10(8.2)
β_{21}	Gold standard		0.96	0.83	1.10
	TA–SMC	RW	0.96(1.0)	0.83(1.0)	1.09(1.0)
		MALA	0.96(0.1)	0.83(0.2)	1.09(0.2)
		SLICE	0.96(10.2)	0.84(9.2)	1.09(10.5)
	NS–SMC	RW	0.96(1.0)	0.83(1.4)	1.10(1.6)
		MALA	0.96(1.0)	0.83(1.4)	1.10(1.2)
		SLICE	0.96(5.4)	0.83(6.6)	1.09(9.6)
β_{31}	Gold standard		0.46	0.30	0.63
	TA–SMC	RW	0.46(1.0)	0.30(1.0)	0.63(1.0)
		MALA	0.46(0.1)	0.30(0.2)	0.63(0.2)
		SLICE	0.46(15.0)	0.30(11.0)	0.62(10.3)
	NS–SMC	RW	0.46(1.6)	0.30(2.1)	0.63(1.9)
		MALA	0.46(1.3)	0.30(1.5)	0.63(1.2)
		SLICE	0.46(8.3)	0.30(8.4)	0.62(7.9)
β_{41}	Gold standard		0.40	0.23	0.57
	TA–SMC	RW	0.40(1.0)	0.23(1.0)	0.57(1.0)
		MALA	0.40(0.1)	0.23(0.2)	0.57(0.2)
		SLICE	0.40(16.9)	0.23(17.7)	0.56(12.6)
	NS–SMC	RW	0.40(1.1)	0.23(1.9)	0.57(1.3)
		MALA	0.40(1.0)	0.23(1.5)	0.57(1.4)
		SLICE	0.40(8.0)	0.23(13.2)	0.56(7.3)
β_{51}	Gold standard		0.42	0.25	0.58
	TA–SMC	RW	0.42(1.0)	0.25(1.0)	0.58(1.0)
		MALA	0.41(0.1)	0.25(0.2)	0.58(0.2)
		SLICE	0.41(18.3)	0.25(16.2)	0.58(12.3)
	NS–SMC	RW	0.41(1.5)	0.25(1.9)	0.58(1.5)
		MALA	0.42(1.1)	0.25(1.2)	0.58(1.7)
		SLICE	0.41(8.3)	0.25(11.8)	0.58(7.7)
β_{61}	Gold standard		0.41	0.25	0.58
	TA–SMC	RW	0.41(1.0)	0.25(1.0)	0.58(1.0)
		MALA	0.41(0.1)	0.25(0.2)	0.58(0.3)
		SLICE	0.41(16.1)	0.25(14.8)	0.58(14.1)
	NS–SMC	RW	0.41(1.1)	0.25(1.4)	0.58(1.8)
		MALA	0.41(0.8)	0.25(0.9)	0.58(1.2)
		SLICE	0.41(8.3)	0.25(11.7)	0.58(9.3)

Table 2.11: Inference Results for FA3 — Part 3 of 4

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \beta_{22}$	Gold standard		−3.23	−6.06	−1.63
	TA–SMC	RW	−3.22(1.0)	−5.83(1.0)	−1.67(1.0)
		MALA	−3.23(0.1)	−5.99(0.3)	−1.65(0.1)
		SLICE	−3.20(5.7)	−5.42(2.8)	−1.73(2.5)
	NS–SMC	RW	−3.24(0.6)	−6.01(0.8)	−1.64(0.6)
		MALA	−3.10(2.5)	−5.19(2.1)	−1.64(1.2)
		SLICE	−3.22(1.8)	−5.74(2.1)	−1.68(1.3)
β_{32}	Gold standard		−0.06	−0.83	0.58
	TA–SMC	RW	−0.06(1.0)	−0.79(1.0)	0.57(1.0)
		MALA	−0.06(0.1)	−0.83(0.1)	0.59(0.2)
		SLICE	−0.06(4.0)	−0.69(4.8)	0.55(3.5)
	NS–SMC	RW	−0.06(0.6)	−0.81(0.7)	0.58(0.9)
		MALA	−0.08(1.2)	−0.76(1.8)	0.56(1.6)
		SLICE	−0.06(1.9)	−0.77(2.5)	0.56(1.9)
β_{42}	Gold standard		0.23	−0.87	0.94
	TA–SMC	RW	0.23(1.0)	−0.82(1.0)	0.93(1.0)
		MALA	0.24(0.1)	−0.86(0.1)	0.94(0.1)
		SLICE	0.19(5.4)	−0.72(6.7)	0.88(29.9)
	NS–SMC	RW	0.23(0.6)	−0.85(0.3)	0.94(0.5)
		MALA	0.22(1.9)	−0.78(2.2)	0.92(2.6)
		SLICE	0.22(2.1)	−0.81(2.2)	0.93(4.2)
β_{52}	Gold standard		0.23	−0.79	0.89
	TA–SMC	RW	0.23(1.0)	−0.74(1.0)	0.87(1.0)
		MALA	0.24(0.1)	−0.78(0.1)	0.88(0.1)
		SLICE	0.20(5.7)	−0.65(5.7)	0.82(15.4)
	NS–SMC	RW	0.23(0.7)	−0.77(0.3)	0.88(0.6)
		MALA	0.22(1.9)	−0.70(1.9)	0.86(2.6)
		SLICE	0.22(2.2)	−0.74(2.3)	0.87(2.9)
β_{62}	Gold standard		0.15	−0.81	0.86
	TA–SMC	RW	0.15(1.0)	−0.77(1.0)	0.84(1.0)
		MALA	0.16(0.1)	−0.80(0.1)	0.86(0.1)
		SLICE	0.12(4.9)	−0.70(6.0)	0.79(16.8)
	NS–SMC	RW	0.15(0.5)	−0.80(0.4)	0.86(0.5)
		MALA	0.13(1.8)	−0.74(2.2)	0.82(1.9)
		SLICE	0.14(1.9)	−0.76(1.9)	0.84(3.0)
$\log \beta_{33}$	Gold standard		−1.21	−3.63	−0.16
	TA–SMC	RW	−1.16(1.0)	−3.30(1.0)	−0.19(1.0)
		MALA	−1.23(0.2)	−3.71(0.3)	−0.17(0.1)
		SLICE	−1.23(6.7)	−2.98(2.1)	−0.28(4.2)
	NS–SMC	RW	−1.22(0.7)	−3.69(1.3)	−0.18(0.5)
		MALA	−1.17(2.8)	−3.00(2.5)	−0.21(1.6)
		SLICE	−1.23(3.8)	−3.50(3.4)	−0.20(1.6)

Table 2.12: Inference Results for FA3 — Part 4 of 4.

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
β_{43}	Gold standard		0.57	−0.54	0.97
	TA–SMC	RW	0.57(1.0)	−0.40(1.0)	0.97(1.0)
		MALA	0.56(0.1)	−0.49(0.1)	0.97(0.1)
		SLICE	0.55(5.6)	−0.24(2.3)	0.96(13.1)
	NS–SMC	RW	0.56(0.5)	−0.50(0.7)	0.97(0.5)
		MALA	0.57(1.7)	−0.35(1.2)	0.97(2.0)
		SLICE	0.57(1.7)	−0.38(1.6)	0.97(2.7)
β_{53}	Gold standard		0.47	−0.53	0.89
	TA–SMC	RW	0.47(1.0)	−0.45(1.0)	0.89(1.0)
		MALA	0.46(0.1)	−0.52(0.1)	0.89(0.1)
		SLICE	0.44(5.5)	−0.32(3.6)	0.87(6.7)
	NS–SMC	RW	0.46(0.5)	−0.51(0.6)	0.89(0.6)
		MALA	0.47(1.8)	−0.40(1.6)	0.88(2.0)
		SLICE	0.47(1.5)	−0.43(1.7)	0.88(1.9)
β_{63}	Gold standard		0.53	−0.44	0.90
	TA–SMC	RW	0.53(1.0)	−0.33(1.0)	0.90(1.0)
		MALA	0.51(0.1)	−0.41(0.1)	0.90(0.1)
		SLICE	0.51(6.2)	−0.19(2.6)	0.88(5.5)
	NS–SMC	RW	0.51(0.6)	−0.40(0.8)	0.90(0.6)
		MALA	0.53(1.9)	−0.27(1.4)	0.89(1.3)
		SLICE	0.53(2.1)	−0.30(1.9)	0.90(2.1)

Table 2.13: Inference Results for ODE model – Part 1 of 2

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log k_1$	Gold standard		−3.40	−4.46	−2.71
	TA–SMC	RW	−3.33(1.0)	−4.06(1.0)	−2.71(1.0)
		MALA	−3.18(1.3)	−3.67(0.4)	−2.66(4.2)
		SLICE	−3.36(8.3)	−4.09(6.8)	−2.71(7.3)
	NS–SMC	RW	−3.34(4.2)	−4.10(3.4)	−2.71(2.5)
		MALA	−3.36(2.7)	−4.06(1.7)	−2.72(2.0)
		SLICE	−3.33(9.8)	−3.99(4.8)	−2.70(7.6)
$\log V_1$	Gold standard		−0.98	−2.02	0.19
	TA–SMC	RW	−1.03(1.0)	−2.02(1.0)	0.04(1.0)
		MALA	−1.09(3.2)	−2.03(4.2)	−0.09(2.2)
		SLICE	−1.00(6.7)	−1.98(12.0)	0.00(6.8)
	NS–SMC	RW	−1.03(1.7)	−2.03(2.9)	0.06(2.4)
		MALA	−0.96(1.3)	−1.95(3.9)	0.15(1.4)
		SLICE	−1.02(5.1)	−1.99(11.0)	−0.00(6.9)
$\log K_{m1}$	Gold standard		−1.01	−4.12	1.04
	TA–SMC	RW	−0.98(1.0)	−3.69(1.0)	0.95(1.0)
		MALA	−1.00(3.2)	−3.57(2.0)	0.79(2.2)
		SLICE	−0.98(6.8)	−3.56(4.7)	0.93(4.3)
	NS–SMC	RW	−0.96(2.2)	−3.62(2.9)	0.95(2.6)
		MALA	−0.90(2.8)	−3.43(2.0)	0.94(2.2)
		SLICE	−1.07(4.9)	−3.69(3.3)	0.88(4.6)
$\log K_{m2}$	Gold standard		−2.86	−4.38	−1.79
	TA–SMC	RW	−2.77(1.0)	−3.92(1.0)	−1.81(1.0)
		MALA	−2.58(1.9)	−3.46(1.0)	−1.76(3.2)
		SLICE	−2.80(7.0)	−3.93(4.8)	−1.82(5.6)
	NS–SMC	RW	−2.79(3.9)	−3.97(3.3)	−1.81(2.3)
		MALA	−2.81(2.1)	−3.93(1.4)	−1.81(1.6)
		SLICE	−2.76(8.3)	−3.84(4.4)	−1.80(6.6)
$\log V_2$	Gold standard		−2.05	−2.64	−1.07
	TA–SMC	RW	−2.11(1.0)	−2.65(1.0)	−1.47(1.0)
		MALA	−2.24(1.1)	−2.70(4.7)	−1.82(0.3)
		SLICE	−2.09(9.0)	−2.64(8.8)	−1.43(6.6)
	NS–SMC	RW	−2.10(4.6)	−2.65(2.7)	−1.42(3.3)
		MALA	−2.09(3.0)	−2.64(2.2)	−1.46(1.6)
		SLICE	−2.10(10.4)	−2.65(8.0)	−1.52(4.4)

Table 2.14: Inference Results for ODE model – Part 2 of 2

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log S(0)$	Gold standard		−0.23	−1.24	0.61
	TA–SMC	RW	−0.22(1.0)	−1.18(1.0)	0.59(1.0)
		MALA	−0.24(3.0)	−1.15(2.9)	0.56(2.6)
		SLICE	−0.23(7.8)	−1.14(10.0)	0.56(9.5)
	NS–SMC	RW	−0.21(1.6)	−1.17(2.5)	0.61(1.6)
		MALA	−0.26(1.6)	−1.22(2.3)	0.55(2.8)
		SLICE	−0.24(4.5)	−1.16(7.5)	0.57(7.4)
$\log D(0)$	Gold standard		−2.88	−6.05	−1.02
	TA–SMC	RW	−2.85(1.0)	−5.76(1.0)	−1.01(1.0)
		MALA	−2.85(2.5)	−5.49(1.2)	−1.06(4.4)
		SLICE	−2.84(6.0)	−5.53(5.3)	−1.04(14.5)
	NS–SMC	RW	−2.89(2.7)	−5.84(3.3)	−1.02(5.0)
		MALA	−2.90(1.1)	−5.75(1.0)	−1.02(2.1)
		SLICE	−2.95(6.5)	−5.66(4.3)	−1.06(12.5)
$\log R(0)$	Gold standard		0.31	−0.28	0.83
	TA–SMC	RW	0.30(1.0)	−0.24(1.0)	0.80(1.0)
		MALA	0.30(2.1)	−0.22(1.7)	0.77(2.2)
		SLICE	0.31(7.2)	−0.21(5.3)	0.79(8.2)
	NS–SMC	RW	0.31(1.9)	−0.24(2.6)	0.81(2.6)
		MALA	0.32(2.1)	−0.20(1.9)	0.82(2.4)
		SLICE	0.30(5.0)	−0.23(4.6)	0.80(5.8)
$\log R_{pp}(0)$	Gold standard		−4.17	−6.89	−2.90
	TA–SMC	RW	−4.14(1.0)	−6.60(1.0)	−2.91(1.0)
		MALA	−4.09(1.1)	−6.25(0.9)	−2.93(3.7)
		SLICE	−4.14(4.3)	−6.32(2.6)	−2.93(6.6)
	NS–SMC	RW	−4.19(6.9)	−6.50(2.1)	−2.93(2.8)
		MALA	−4.15(1.6)	−6.40(1.0)	−2.93(2.3)
		SLICE	−4.10(2.8)	−6.27(2.3)	−2.92(3.9)

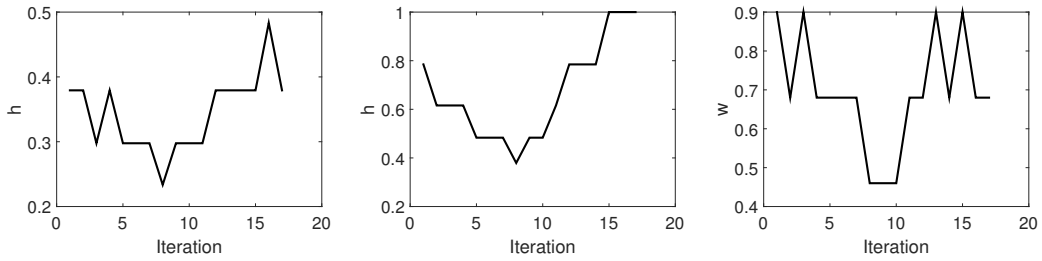
2.10 Calibration Plots

The following plots display the evolution of the automated choice of MCMC kernel parameters h (for RW/MALA) and w (for Slice Sampling), as well as the evolution of the choice of MCMC iterations (repeats) chosen by the Calibration methods described in Section 2.5.

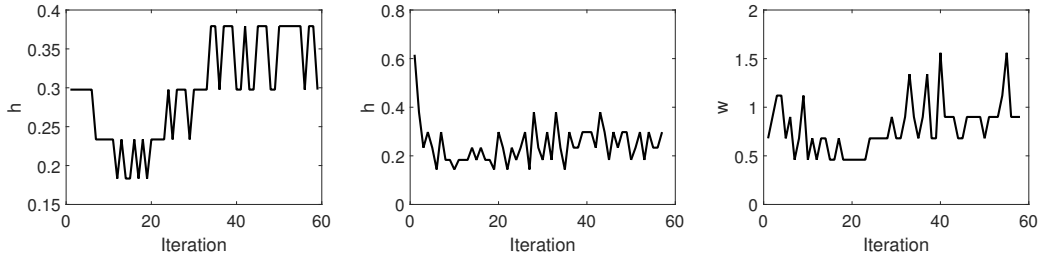
We use a range of twenty possible values for h that are logarithmically spaced on the interval [0.01,1], and ten possible values for w that are linearly spaced on the interval [0.02,2].

Note here that iteration on the x–axis refers not to MCMC iteration, but instead the time step of the SMC sampler.

Factor Analysis – One Factor

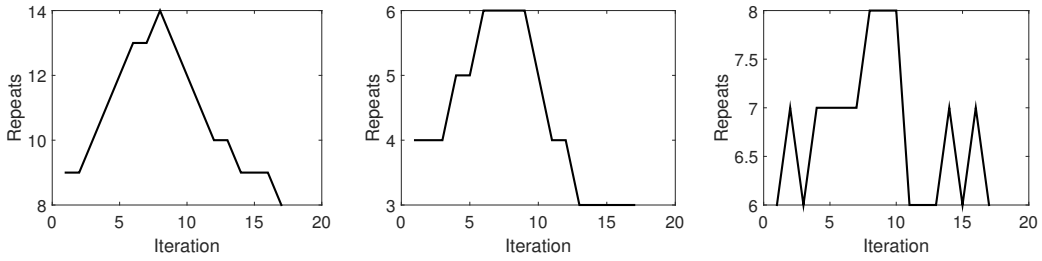


(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE

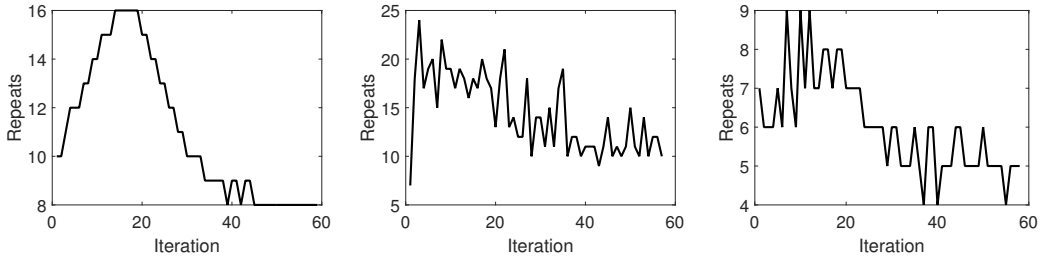


(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.11: MCMC parameter selection for FA1.



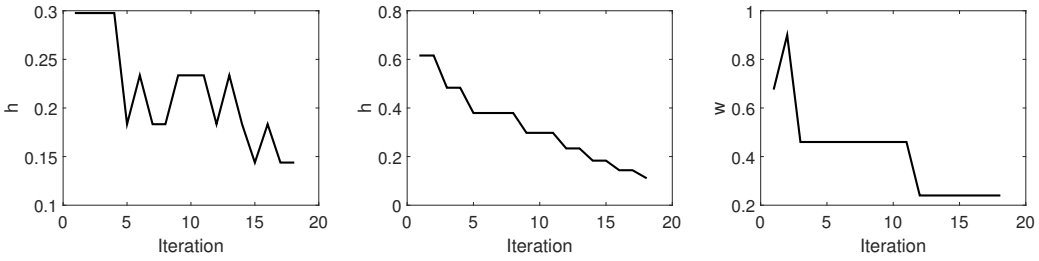
(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE



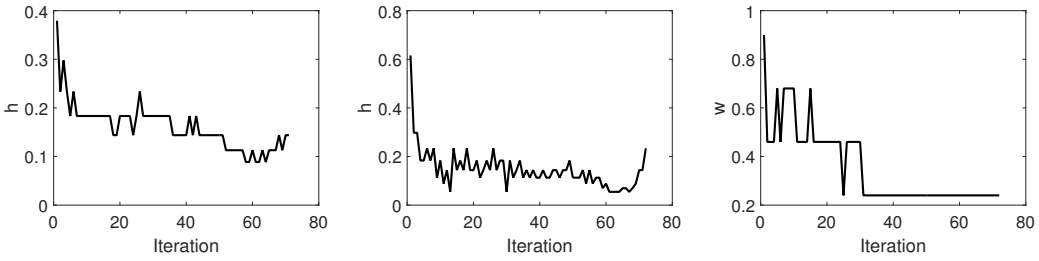
(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.12: Repeats selection for FA1

Factor Analysis – Two Factors

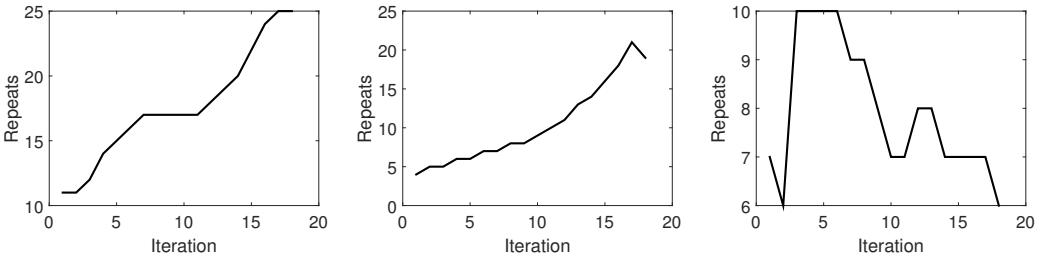


(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE

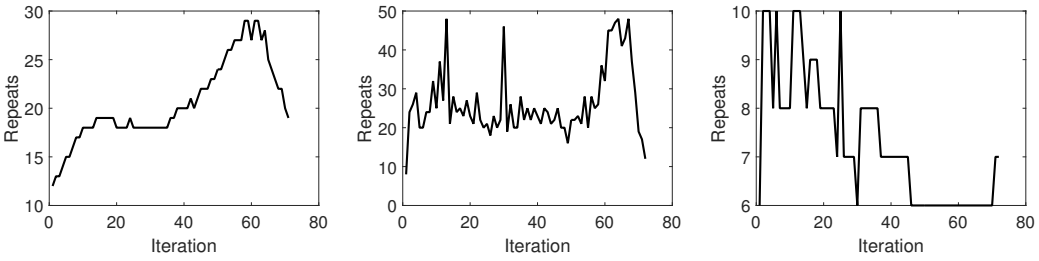


(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.13: MCMC parameter selection for FA2.



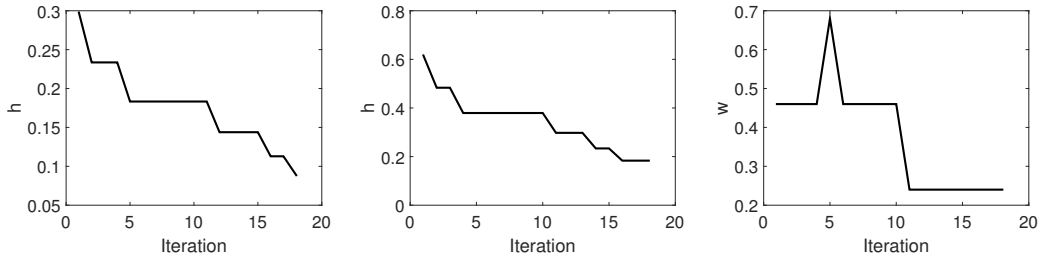
(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE



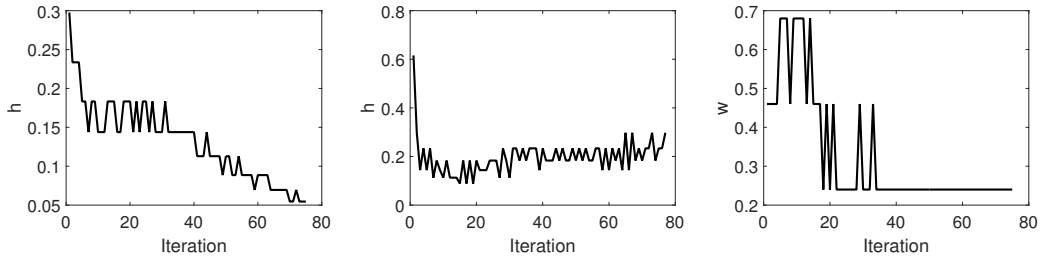
(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.14: Repeats selection for FA2.

Factor Analysis – Three Factors

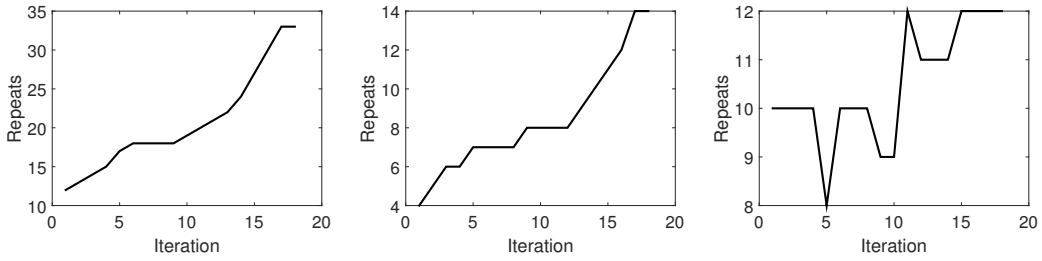


(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE

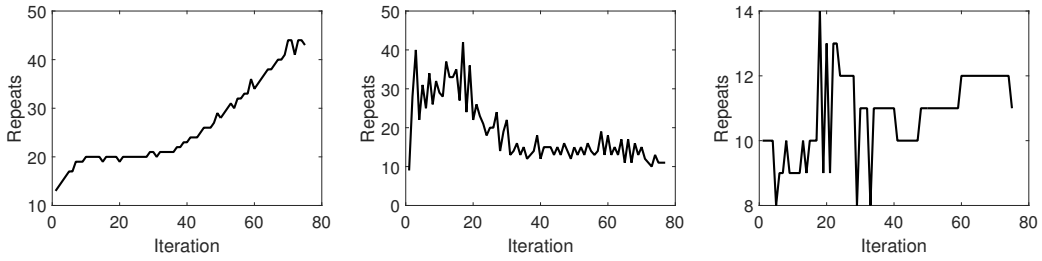


(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.15: MCMC parameter selection for FA3.



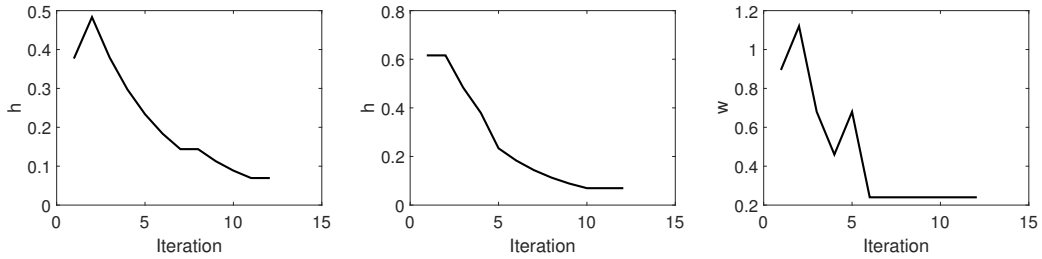
(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE



(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.16: Repeats selection for FA3.

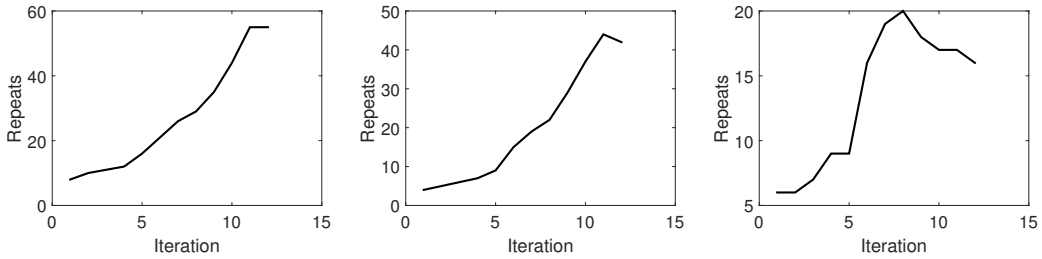
ODE



(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE

(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.17: MCMC parameter selection for the ODE model.



(a) TA–SMC RW (b) TA–SMC MALA (c) TA–SMC SLICE

(d) NS–SMC RW (e) NS–SMC MALA (f) NS–SMC SLICE

Figure 2.18: Repeats selection for the ODE model.

Part II

Computation of Sums of Random Variables

Chapter 3: Authorship Statement

Citation

Botev, Z.I., **Salomone, R.**, and Mackinlay, D. (2017). Fast and Accurate Computation of the Distribution of Sums of Dependent Log-Normals. Submitted to *Annals of Operations Research* on 6th June 2017, with subsequent revision submitted on 24th May 2018.

The manuscript with further revisions is included.

Contributions

I was responsible for 50% of the overall work. Specifically,

- I contributed equally with Z.I. Botev in the writing of the paper, and equally with Z.I. Botev and D. Mackinlay in the conception and development of the project and methodology therein.
- I was responsible for the majority (90%) of the design and implementation of the numerical experiments, as well as the interpretation of their results. Z.I. Botev was responsible for the efficiency proofs of the proposed methods in Section 3.5.
- I contributed equally with Z.I. Botev and D. Mackinlay to the editing of the paper.

Chapter 3

Fast and Accurate Computation of the Distribution of Sums of Dependent Log-Normals

3.1 Introduction

The distribution of the sum of log-normals (SLN) has numerous practical applications [26] — in the pricing of Asian options under a Black-Scholes model [10, 15, 28]; in wireless systems in telecommunications [18, 30]; in insurance value-at-risk computations [16, 27, 33]; and recently even in the modelling of viral social media phenomena [14]. For this reason, the accurate computation of characteristics of the SLN distribution are receiving increasing attention.

The first left-tail efficient Monte Carlo method for the estimation of the SLN cumulative distribution function (cdf) was proposed by Gulisashvili and Tankov [19]. This was then followed by Asmussen et al. [6, 7, 25] who approximate the cdf using Laplace transform techniques. Up until these seminal works, the only available approximations of the cdf were deterministic moment-matching heuristics [17, 21], whose accuracy quickly deteriorates when the sum is not iid, as seen from the example in Figure 3.5.

With the exception of the defining works [5, 19], all of the existing proposals can only deal with the distribution of the sum of independent log-normals, or, in the case of [25], with the Laplace transform of the SLN distribution. Other examples of research in the area include the efficient estimation of the right tail of the SLN distribution under the assumption of independent log-normal factors [8, 29], and,

of more consequence for practical applications, under the assumption of correlated factors [4, 9, 22, 23].

In this article, we present new Monte Carlo estimators for the cdf, pdf (density function), and right tail (complementary distribution function) of SLN distribution. Regarding these three new estimators, our original contributions can be summarized as follows.

Cdf estimator. We propose a new asymptotically efficient estimator of the cdf with competitive practical performance. Overall, we find that our proposed estimator is the preferred choice when the goal is to compute the left tail of the *general* SLN distribution (for the special case of iid log-normals, we also find that the estimator of Asmussen et al. [7] is also an excellent choice).

Pdf estimator. Our novel estimator of the pdf of the SLN distribution is infinitely smooth in the model parameters. As a result of this, in a quasi Monte Carlo setting, this smoothness accelerates the rate of convergence beyond that of the canonical Monte Carlo rate of $O(\sqrt{n})$ (n is the Monte Carlo sample size). Our numerical experiments show that the new estimator is preferable to the existing Fenton–Wilkinson-type [17, 21] approximations and can be used the validate the accuracy of an ingenious orthogonal series [5] approximation.

Right tail estimator. We show both numerically and theoretically that many of the existing proposals for estimating the right tail of the SLN distribution [3, 4, 8, 19, 22] can be unreliable in some simple examples of applied interest. More precisely, while the existing estimators work satisfactory when the log-normal variates are independent, these estimators exhibit exploding variance in cases of positively correlated log-normals. Unfortunately, dependence structures which induce strong positive correlation are precisely the cases of practical interest in finance and reliability (the computation of such tail probabilities arises, for example, in estimating the likelihood of a large loss from a portfolio with asset prices driven by the Black–Scholes geometric Brownian motion model [24, Chapter 15]).

In addition to proving that our estimator is asymptotically efficient as we move deeper and deeper into the right tail, we show that, at least on the limited number of examples we consider, it is more accurate than its competitors by many orders of magnitude.

Further to this, we provide a refinement of the tail asymptotics of the log-normal distribution (item 1 of Lemma 2), and use this refinement to prove that our estimator is second-order efficient. A second-order efficient estimator is one whose precision or standard error can be estimated reliably from simulation, a property only enjoyed by our new estimator (Corollary 2). The second-order efficiency results resolve the following paradox: a weakly efficient estimator can perform significantly better than its strongly

efficient competitors. In other words, a strongly efficient estimator that is not second-order efficient may perform poorly compared to a weakly efficient estimator that is second-order efficient.

Finally, it is frequently the case that we not only wish to estimate the probability of a rare-event, but also wish to draw random states conditional on the rare-event. In this article we propose a sampler for exact simulating from the SLN distribution conditional on a left-tailed rare event.

The rest of the paper is organized around the three qualitatively different parts of the SLN distribution: (1) the left tail of the SLN distribution; (2) the density of the main body of the distribution; (3) the right tail of SLN distribution. In all three cases we wish to control the (quasi) Monte Carlo error of the estimator.

The left tail and main body is covered in Section 3.2, and the right tail is considered in Section 3.3. In Section 3.3.2 we review the *importance sampling vanishing error* (ISVE) estimator [4], and demonstrate that in certain cases it yields highly inaccurate estimates that tend to severely underestimate the true probability. Intuition is provided for the poor practical performance of the estimator and then, in Section 3.3.3, we propose a novel estimator and describe its theoretical properties. Numerical illustrations of the main theoretical findings and a concluding section follow.

3.2 Left Tail and Density

We start by considering the cumulative distribution function of the SLN:

$$\ell(\gamma) = \mathbb{P}(X_1 + \cdots + X_d \leq \gamma),$$

where: (1) $\boldsymbol{X} = (X_1, \ldots, X_d)$ has (dependent) log-normal random components governed by a Gaussian copula, so that $\ln \boldsymbol{X} \sim \mathsf{N}(\boldsymbol{\nu}, \Sigma)$ for some positive definite covariance matrix Σ; and (2) the parameter $\gamma > 0$ is allowed to be a small enough threshold so that ℓ is a small or rare-event probability.

Then, if $\mathrm{L}\mathrm{L}^\top = \Sigma$ is the lower triangular decomposition of the covariance matrix, we can write (l_{ij} is the (i, j)-th element of L)

$$\ell = \mathbb{P}(\exp(\nu_1 + l_{11} Z_1) + \cdots + \exp(\nu_d + \textstyle\sum_j l_{dj} Z_j) \leq \gamma),$$

where under the measure $\mathbb{P}$, we have $\boldsymbol{Z} \sim \mathsf{N}(\mathbf{0}, \mathrm{I})$. In other words, under $\mathbb{P}$ we can set $X_k(\boldsymbol{Z}) = \exp\left(\nu_k + \sum_{i \leq k} l_{ki} Z_i\right)$, which we henceforth assume. To proceed, note that the events

$$\{X_1 \leq \gamma\} \supseteq \{X_1 + X_2 \leq \gamma\} \supseteq \cdots \supseteq \{X_1 + \cdots + X_d \leq \gamma\}$$

are nested. In other words, if we define

$$\alpha_j(z_1,\ldots,z_{j-1}) \stackrel{\text{def}}{=} \frac{\ln(\gamma - \sum_{k<j} x_k(z_1,\ldots,z_{j-1})) - \nu_j - \sum_{k<j} l_{jk} z_k}{l_{jj}}, \quad j > 1,$$

then, the following events are nested:

$$\{Z_1 \le \alpha_1\} \supseteq \{Z_2 \le \alpha_2(Z_1)\} \supseteq \cdots \supseteq \{Z_d \le \alpha_d(Z_1,\ldots,Z_{d-1})\},$$

with the last one being equivalent to the event of interest.

Let $\mathsf{TN}_{(l,u)}(\mu,\sigma^2)$ denote the normal distribution $\mathsf{N}(\mu,\sigma^2)$ truncated to the interval (l,u). Further, let $\mathbb{I}\{\cdot\}$ be the indicator function that equals unity when the statement inside the curly brackets is true and zero otherwise. The above nested sequence of events then suggests that the following sequential simulation of $\boldsymbol{Z}$ will entail the occurrence of the (possibly rare) event[1]:

$$\begin{aligned}
Z_1 &\sim \frac{\phi(z_1)\mathbb{I}\{z_1 \le \alpha_1\}}{\Phi(\alpha_1)} \equiv \mathsf{TN}_{(-\infty,\alpha_1)}(0,1),\\
Z_2|Z_1 &\sim \frac{\phi(z_2)\mathbb{I}\{z_2 \le \alpha_2(z_1)\}}{\Phi(\alpha_2(z_1))} \equiv \mathsf{TN}_{(-\infty,\alpha_2)}(0,1),\\
&\vdots\\
Z_d|Z_1,\ldots,Z_{d-1} &\sim \frac{\phi(z_d)\mathbb{I}\{z_d \le \alpha_d(z_1,\ldots,z_{d-1})\}}{\Phi(\alpha_d(z_1,\ldots,z_{d-1}))} \equiv \mathsf{TN}_{(-\infty,\alpha_d)}(0,1).
\end{aligned}$$

With the above sampling scheme, the unbiased importance sampling estimator of ℓ (based on a single realization) is: $\prod_{j=1}^{d} \Phi(\alpha_j(Z_1,\ldots,Z_{j-1}))$. We note that Ambartzumian et al. [1] also simulate from truncated normal densities sequentially. However, their approach applies only to a Gaussian random vector restricted to a rectangular set and does not apply to the sum of log-normal variables considered here and restricted to a half-line.

3.2.1 Sequential Importance Sampling Estimator

Although under the sequential sampling scheme above, the rare-event occurs with probability one, and the estimator is smooth, it is not necessarily an efficient one when γ is allowed to be arbitrarily small. To achieve asymptotic efficiency, we instead suggest the following parametric change of measure for $\boldsymbol{Z}$,

[1] We denote the standard normal pdf with covariance Σ via $\phi_\Sigma(\cdot)$ ($\phi(\cdot) \equiv \phi_{\mathrm{I}}(\cdot)$) and the univariate cdf and complementary cdf by $\Phi(\cdot)$ and $\overline{\Phi}(\cdot)$, respectively.

where the parameter $\boldsymbol{\mu}$ still remains to be determined:

$$\begin{aligned} Z_1 &\sim \mathsf{TN}_{(-\infty,\alpha_1)}(\mu_1, 1), \\ Z_2|Z_1 &\sim \mathsf{TN}_{(-\infty,\alpha_2)}(\mu_2, 1), \\ &\vdots \\ Z_d|Z_1,\ldots,Z_{d-1} &\sim \mathsf{TN}_{(-\infty,\alpha_d)}(\mu_d, 1). \end{aligned}$$

Denote the measure used to simulate $\boldsymbol{Z}$ as $\mathbb{P}_{\boldsymbol{\mu}}$ and the corresponding expectation (variance) operators as $\mathbb{E}_{\boldsymbol{\mu}}$ ($\mathbb{V}\mathrm{ar}_{\boldsymbol{\mu}}$). Let the logarithm of the Radon-Nikodym derivative, $\mathrm{d}\mathbb{P}/\mathrm{d}\mathbb{P}_{\boldsymbol{\mu}}$, be denoted as

$$\psi(\boldsymbol{z};\boldsymbol{\mu}) \stackrel{\text{def}}{=} \frac{\|\boldsymbol{\mu}\|^2}{2} - \boldsymbol{z}^\top\boldsymbol{\mu} + \sum_{j=1}^{d} \ln\overline{\Phi}(\mu_j - \alpha_j(\boldsymbol{z})),$$

and let $\mathcal{W} = \{\boldsymbol{w} : \boldsymbol{w} \geq \boldsymbol{0}, \boldsymbol{1}^\top\boldsymbol{w} = 1\}$ be the set of discrete probability distributions with support on d states in $\mathbb{R}$. Then, our proposed unbiased estimator is

$$\hat{\ell} = \exp(\psi(\boldsymbol{Z};\boldsymbol{\mu}^*)), \qquad \boldsymbol{Z} \sim \mathbb{P}_{\boldsymbol{\mu}^*}, \tag{3.1}$$

where $\boldsymbol{\mu}^*$ is the solution to the program:

$$(\boldsymbol{w}^*, \boldsymbol{\mu}^*) = \operatorname*{argmin}_{\boldsymbol{w}\in\mathcal{W},\boldsymbol{\mu}} \left\{ \|\boldsymbol{\mu}\|^2 + \ln\overline{\Phi}\left(\frac{\boldsymbol{w}^\top(\boldsymbol{\nu} - \mathrm{L}\boldsymbol{\mu}) - \ln\gamma - \boldsymbol{w}^\top\ln\boldsymbol{w}}{\sqrt{\boldsymbol{w}^\top\Sigma\boldsymbol{w}}} \right) \right\}. \tag{3.2}$$

Why is (3.1) a good estimator? In addition to its superior numerical performance (see Section 3.2.4) compared to the Gulisashvili and Tankov (GT) estimator [19, Equation (65)], it is also a logarithmically efficient estimator as $\gamma \downarrow 0$. The efficiency label stems from the fact that the relative error, $\mathbb{V}\mathrm{ar}(\hat{\ell}_{\mathrm{CMC}})/\ell^2$, of the crude Monte Carlo estimator,

$$\hat{\ell}_{\mathrm{CMC}} = \mathbb{I}\{X_1 + \cdots + X_d \leq \gamma\}, \quad \ln\boldsymbol{X} \sim \mathsf{N}(\boldsymbol{\nu}, \Sigma),$$

grows exponentially (in γ), while the relative error of (3.1) grows only polynomially as $\gamma \downarrow 0$. This is formally stated in the following theorem, which is proven in the appendix.

Theorem 1 (Logarithmic Efficiency of Estimator)**.** The estimator (3.1) is logarithmically efficient; that is,

$$\liminf_{\gamma\downarrow 0} \frac{\ln\mathbb{E}_{\boldsymbol{\mu}^*}\hat{\ell}^2(\gamma)}{\ln\ell(\gamma)} = 2,$$

with relative error [2] that grows as $\frac{\mathbb{E}_{\boldsymbol{\mu}^*}\hat{\ell}^2(\gamma)}{\ell^2(\gamma)} = O((-\ln\gamma)^{(d+1)})$.

[2] The notation $f(x) \simeq g(x)$ as $x \to a$ stands for $\lim_{x\to a} f(x)/g(x) = 1$. Similarly, we define $f(x) = O(g(x)) \Leftrightarrow \lim_{x\to a} |f(x)/g(x)| < \text{const.} < \infty$; $f(x) = o(g(x)) \Leftrightarrow \lim_{x\to a} f(x)/g(x) = 0$; also, $f(x) = \Theta(g(x)) \Leftrightarrow f(x) = O(g(x))$ and $g(x) = O(f(x))$.

A significant advantage of (3.1) is that it is amenable to a randomized quasi Monte Carlo implementation [24, Chapter 2, Algorithm 2.3]. This is because (3.1) is a smooth infinitely differentiable estimator and as a result has finite *Koksma–Hlawka* discrepancy bound [24, Chapter 2, Equation 2.3]. While the standard error of a Monte Carlo estimator, driven by pseudorandom numbers, decays at the canonical rate of $O(n^{-1/2})$, the standard error of a Quasi Monte Carlo estimator, driven by quasirandom numbers, decays at the superior rate of $O(n^{-1/2-\delta})$ for some $\delta > 0$ that depends on the dimension d and the smoothness of the estimator.

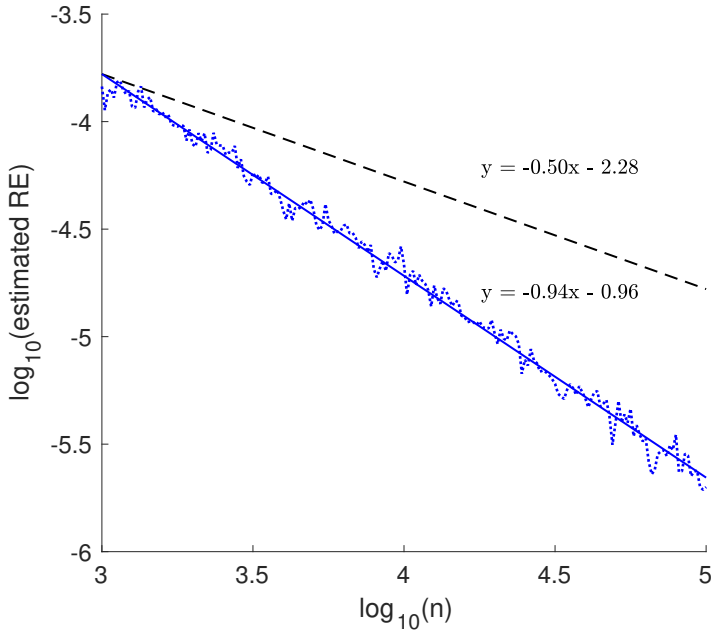


Figure 3.1: The relative error of estimator (3.1) for $\gamma = \mathbb{E}S = 5\exp(1/2), d = 5, \Sigma = \mathrm{I}, \boldsymbol{\nu} = \mathbf{0}$. Also displayed is a reference line with the canonical slope of $-1/2$.

Figure 3.1 below shows that for $d = 5$, the rate of decay of (3.1) improves from the canonical rate of $O(n^{-0.5})$ (when using a pseudorandom sequence) to approximately $O(n^{-0.94})$ when using Sobol's quasirandom sequence [24, Section 2.5]. Here the relative error is estimated using 100 independent random shifts of Sobol's quasirandom pointset [24, Section 2.7], and the number -0.94 is simply the slope of the line of best fit.

The advantage of smoothness even carries over to the estimator of the density of the SLN distribution. The result is that we achieve significant variance reduction — a point illustrated in Figure 3.4 later on.

3.2.2 Density Estimator

To derive the smooth density estimator, we use the so-called *push-out method* [32, Chapter 7]. In particular, observe that we can "push-out" γ as follows:

$$\ell(\gamma) = \mathbb{P}(\exp(\nu_1 - \ln(\gamma) + l_{11}Z_1) + \cdots + \exp(\nu_d - \ln(\gamma) + \textstyle\sum_{j=1}^{d} l_{dj}Z_j) \le 1).$$

Therefore, the pdf of the SLN distribution can written as the integral:

$$\begin{aligned} f(\gamma) = \frac{\partial \ell}{\partial \gamma} &= \int_{\sum_i \exp(u_i)<1} \frac{\partial}{\partial \gamma} \phi_\Sigma(\boldsymbol{u} - \boldsymbol{\nu} + \mathbf{1}\ln\gamma)\mathrm{d}\boldsymbol{u} \\ &= \int_{\sum_i \exp(u_i)<1} \phi_\Sigma(\boldsymbol{u} - \boldsymbol{\nu} + \mathbf{1}\ln\gamma)\frac{-\mathbf{1}^\top\Sigma^{-1}(\boldsymbol{u} - \boldsymbol{\nu} + \mathbf{1}\ln\gamma)}{\gamma}\mathrm{d}\boldsymbol{u} \\ &= \int_{\sum_i \exp(u_i)<\gamma} \phi_\Sigma(\boldsymbol{u} - \boldsymbol{\nu})\frac{-\mathbf{1}^\top\Sigma^{-1}(\boldsymbol{u} - \boldsymbol{\nu})}{\gamma}\mathrm{d}\boldsymbol{u} \\ &= \int_{\mathbb{R}^d} \phi(\boldsymbol{z})\frac{-\boldsymbol{z}^\top \mathrm{L}^{-1}\mathbf{1}}{\gamma}\mathbb{I}\{\exp(\nu_1 + l_{11}z_1) + \cdots + \exp(\nu_d + \textstyle\sum_{j=1}^{d} l_{dj}z_j) < \gamma\}\mathrm{d}\boldsymbol{z} \end{aligned}$$

As a result, our smooth unbiased estimator of the SLN pdf is:

$$\hat{f}(\gamma) = \exp(\psi(\boldsymbol{Z};\boldsymbol{\mu}^*))\frac{-\boldsymbol{Z}^\top \mathrm{L}^{-1}\mathbf{1}}{\gamma}, \qquad \boldsymbol{Z} \sim \mathbb{P}_{\boldsymbol{\mu}^*}. \tag{3.3}$$

Our numerical experiments suggest (see, for example, Table 3.4 and 3.5 below) that this estimator performs very well for the wide range of γ.

3.2.3 Exact Simulation from Conditional Distribution

One of advantages of our approach is that, when d is not too large, it is possible to simulate exactly from the distribution of $\boldsymbol{X}$ conditional on the rare-event $\left\{\sum_{k=1}^{d} X_k \le \gamma\right\}$. As $\psi(\boldsymbol{z};\boldsymbol{\mu}^*)$ is concave in $\boldsymbol{z}$ for any fixed $\boldsymbol{\mu}^*$ (see, for example, [12, Lemma 1]), we can easily obtain its maximum. This can then be used in the following acceptance-rejection sampling procedure.

1. **Require:** $c = \max_{\boldsymbol{z}} \psi(\boldsymbol{z};\boldsymbol{\mu}^*)$
2. **Until** $E > c - \psi(\boldsymbol{Z};\boldsymbol{\mu}^*)$ **do**
   Simulate $\boldsymbol{Z} \sim \mathbb{P}_{\boldsymbol{\mu}^*}$ and $E \sim \mathsf{Exp}(1)$, independently.
3. **Return** $\boldsymbol{X} = \exp(\boldsymbol{\nu} + \mathrm{L}\boldsymbol{Z})$ as a sample from the conditional distribution.

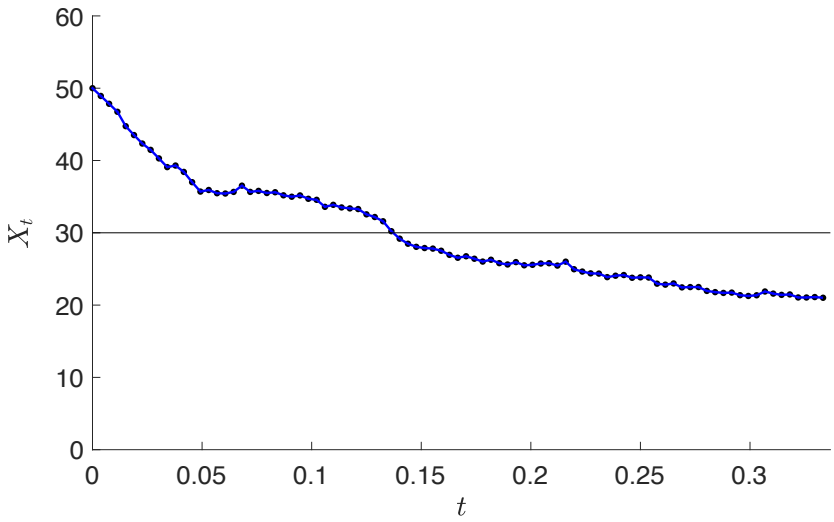


Figure 3.2: Stock price trajectory, conditional on $\bar{X}_T \leq 30$.

As an example, we consider the exact simulation of the stock price trajectories of an Asian option with positive payoff, where the average value of a stock price observed at a set of discrete times on the interval $[0,T]$:

$$\bar{X}_T = \frac{1}{d+1}\sum_{i=0}^{d} X_{t_i}, \quad t_0 = 0 < t_1 < t_2 < \cdots < t_d = T\ .$$

Recall that, under the Black–Scholes model, $X_t = X_0 \exp\left((r-\sigma^2/2)t + \sigma W_t\right)$, where (1) W_t is the Wiener process at time t; (2) σ is the volatility coefficient; (3) r is the risk-free interest rate. Then, for an Asian Put option with maturity T and strike price K the payout is $(K-\bar{X}_T)^+$. Since $\boldsymbol{X} = \left(X_{t_1}, X_{t_2}, \ldots, X_{t_d}\right)$ is a log-normal random vector with

$$\nu_i = \ln(X_0) + (r-\sigma^2/2)t_i, \quad i = 1,\ldots,d$$
$$\Sigma_{ij} = \sigma^2 \min\left(t_i, t_j\right), \quad i,j = 1,\ldots,d,$$

we can use our algorithm above to simulate a realization of the stock price path conditional on the event $\{\bar{X}_T < K\} = \{X_{t_1} + \cdots + X_{t_d} < (d+1)K - X_0\}$. Simulation of such an $\boldsymbol{X}$ conditional on the rare-event $\{\bar{X}_T \leq 30\}$ provides insight into how the rare event occurs, that is, how the stock price must behave for a positive payoff.

Figure 3.2 shows one stock price realization with parameters $X_0 = 50, K = 30, \sigma = 0.25, r = 0.07, T = 4/12, d = 88$. We note that, since $\mathbb{P}(\bar{X}_T \leq 30) \approx 2\times 10^{-11}$ is a rare-event probability, exact simulation of such a stock price trajectory is not possible using a naive acceptance-rejection, because the acceptance

rate would be approximately 2×10^{-11}. Instead, our algorithm enjoys the (estimated) acceptance rate of 5.9%.

3.2.4 Numerical Comparison with Monte Carlo Estimators

In the following two subsections we separately consider Monte Carlo estimators for the cdf (left tail) and the density function.

Cumulative distribution function (left tail)

First, we compare the performance (3.1) against the GT estimator [19, Equation (65)]. In comparing relative performance, we use the (estimated) relative error in percentage, $\mathrm{RE}(\widehat{\ell}) = \sqrt{\mathbb{V}\mathrm{ar}(\widehat{\ell})/n}\,/\ell$ and *work-normalized relative variance*,

$$\mathrm{WNRV}(\widehat{\ell}) = \mathrm{RE}^2(\widehat{\ell})\times(\text{total computing time in seconds}).$$

Table 3.1 shows the numerical results using a sample size of $n = 10^6$ and the parameters $d = 20, \boldsymbol{\nu} = \mathbf{0}, \Sigma = \mathrm{diag}(\boldsymbol{\sigma})$, where $\sigma_k^2 = k$. The results are self-explanatory — we can see that for $\gamma = 1$, the relative

Table 3.1: Results for $d = 20, \boldsymbol{\nu} = \mathbf{0}, \Sigma = \mathrm{diag}(\boldsymbol{\sigma})$, where $\sigma_k^2 = k$.

γ	$\widehat{\ell}$	$\widehat{\ell}_{\mathrm{GT}}$	$\mathrm{RE}(\widehat{\ell})\%$	$\mathrm{RE}(\widehat{\ell}_{\mathrm{GT}})\%$	$\mathrm{WNRV}(\widehat{\ell})$	$\mathrm{WNRV}(\widehat{\ell}_{\mathrm{GT}})$
12	1.68×10^{-4}	1.67×10^{-4}	0.198	4.81	2.37×10^{-5}	1.75×10^{-3}
10	6.82×10^{-5}	6.88×10^{-5}	0.217	6.66	2.84×10^{-5}	3.26×10^{-3}
8	2.01×10^{-5}	2.02×10^{-5}	0.244	4.91	3.69×10^{-5}	1.85×10^{-3}
6	3.54×10^{-6}	3.46×10^{-6}	0.285	5.17	5.00×10^{-5}	2.00×10^{-3}
4	2.13×10^{-7}	2.17×10^{-7}	0.368	5.46	8.44×10^{-5}	2.27×10^{-3}
3	2.20×10^{-8}	2.35×10^{-8}	0.439	6.89	1.21×10^{-4}	3.65×10^{-3}
2	6.05×10^{-10}	5.63×10^{-10}	0.567	10.9	2.04×10^{-4}	9.19×10^{-3}
1	4.24×10^{-13}	4.31×10^{-13}	0.937	17.8	5.47×10^{-4}	2.35×10^{-2}

error of the GT estimator is large.

In our numerical simulations we observe that the GT estimator performs at its best when all ν's are the same, and otherwise it may not perform so well. For example, in Table 3.2 the relative error is larger, because we use the different means $\nu_k = k - d,\ k = 1,\ldots,d$.

In the above setting, it appears that the accuracy of the GT estimator initially deteriorates before it improves. One explanation for this phenomenon is that the asymptotic approximation upon which the GT estimator is built is poor in a non-asymptotic regime – a point explained in detail in [13, Section 2.2].

Table 3.2: Results for $\Sigma = \mathrm{diag}(\boldsymbol{\sigma}), \nu_k = k - d, \sigma_k^2 = k, d = 10$.

			relative error %		work normalized rel. var.	
γ	$\widehat{\ell}$	$\widehat{\ell}_{\mathrm{GT}}$	$\mathrm{RE}(\widehat{\ell})$	$\mathrm{RE}(\widehat{\ell}_{\mathrm{GT}})$	$\mathrm{WNRV}(\widehat{\ell})$	$\mathrm{WNRV}(\widehat{\ell}_{\mathrm{GT}})$
1	1.25×10^{-1}	5.47×10^{-9}	0.0389	41	4.68×10^{-7}	6.58×10^{-2}
1×10^{-1}	2.75×10^{-3}	5.39×10^{-5}	0.0956	51.4	2.82×10^{-6}	1.02×10^{-1}
1×10^{-2}	7.10×10^{-7}	7.47×10^{-7}	0.209	38	1.33×10^{-5}	5.67×10^{-2}
1×10^{-3}	8.59×10^{-14}	8.13×10^{-14}	0.466	7.58	6.80×10^{-5}	2.29×10^{-3}
1×10^{-4}	1.03×10^{-25}	1.07×10^{-25}	0.967	9.68	2.99×10^{-4}	3.77×10^{-3}
1×10^{-5}	1.10×10^{-43}	8.92×10^{-44}	1.79	11.9	1.01×10^{-3}	5.49×10^{-3}
1×10^{-6}	4.27×10^{-68}	2.61×10^{-68}	2.81	14.2	2.48×10^{-3}	8.03×10^{-3}

Finally, we observe that both estimators benefit from positive correlation. For example, if we take $\boldsymbol{\nu}$ to be a linearly spaced vector on the interval $[0, 1/4]$ with $d = 50$, and $\rho = 0.25, \Sigma = 0.25^2(\rho\mathbf{1}\mathbf{1}^\top + (1-\rho)\mathrm{I})$, then Table 3.3 shows the slowly increasing relative error for both estimators as γ becomes smaller. Again, observe that the variance of the new estimator (3.1) is typically more than a hundred times smaller.

Table 3.3: Results for covariance matrix with positive correlation.

			relative error %		work normalized rel. var.	
γ	$\widehat{\ell}$	$\widehat{\ell}_{\mathrm{GT}}$	$\mathrm{RE}(\widehat{\ell})$	$\mathrm{RE}(\widehat{\ell}_{\mathrm{GT}})$	$\mathrm{WNRV}(\widehat{\ell})$	$\mathrm{WNRV}(\widehat{\ell}_{\mathrm{GT}})$
40	1.85×10^{-3}	1.86×10^{-3}	0.169	2.45	4.41×10^{-5}	1.07×10^{-3}
38	4.83×10^{-4}	5.10×10^{-4}	0.178	3.24	4.92×10^{-5}	1.93×10^{-3}
36	9.96×10^{-5}	9.63×10^{-5}	0.189	2.01	5.49×10^{-5}	7.57×10^{-4}
34	1.57×10^{-5}	1.56×10^{-5}	0.199	4.47	6.09×10^{-5}	3.62×10^{-3}
32	1.79×10^{-6}	1.89×10^{-6}	0.209	7.62	6.78×10^{-5}	1.06×10^{-2}
30	1.41×10^{-7}	1.36×10^{-7}	0.219	2.86	7.43×10^{-5}	1.52×10^{-3}
28	7.06×10^{-9}	7.10×10^{-9}	0.230	2.70	8.25×10^{-5}	1.32×10^{-3}
26	2.09×10^{-10}	2.13×10^{-10}	0.241	4.11	9.00×10^{-5}	3.18×10^{-3}
24	3.25×10^{-12}	3.37×10^{-12}	0.251	3.13	9.73×10^{-5}	1.84×10^{-3}
22	2.28×10^{-14}	2.42×10^{-14}	0.263	3.83	1.08×10^{-4}	2.66×10^{-3}

Recently, Asmussen et al. [7] proposed a highly accurate Monte Carlo estimator that is designed to work exclusively in the special case when the sum of log-normals is iid. We found that, at least on the numerical examples considered in the supplementary material by Asmussen et al. [7], our estimator is as accurate as theirs (results not shown here). Thus, in the special case of iid sums, their simpler estimator may be preferable to our more complex estimator with one important caveat — our estimator will be useful in the few cases (e.g., when $d = 64, \sigma^2 = 0.035$) for which [7] report NaN or numerical errors.

Density Function

Using estimator (3.3), we can estimate accurately the effect of the correlation coefficient ρ on the shape of the SLN pdf. Figure 3.3 shows that as ρ increases the tail of the SLN pdf becomes thicker.

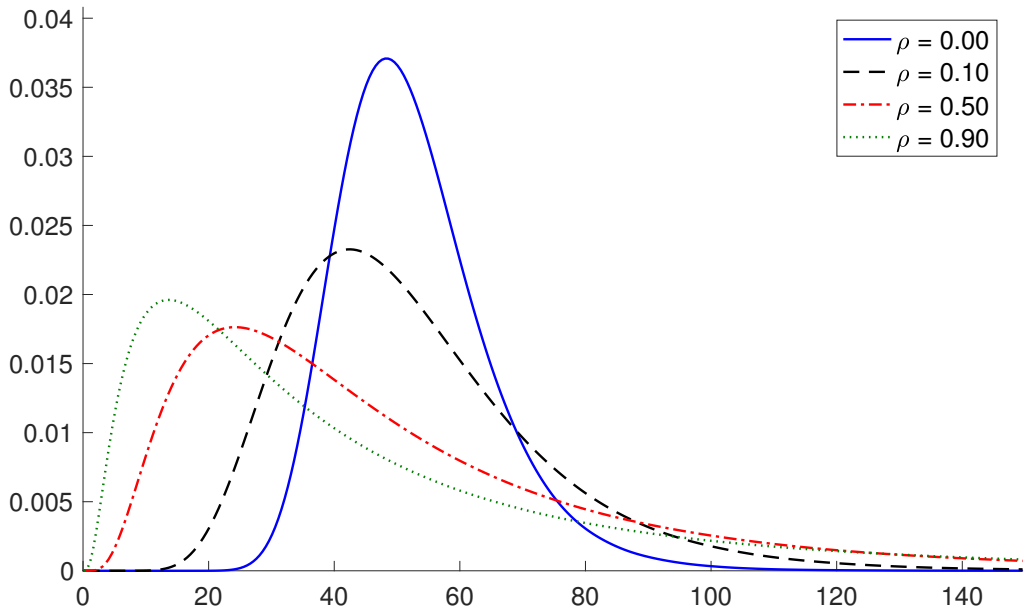


Figure 3.3: Estimate of the SLN pdf for $d = 32, \boldsymbol{\nu} = \mathbf{0}$, $\Sigma = \rho \mathbf{1}\mathbf{1}^\top + (1-\rho)\mathrm{I}$, and varying ρ.

The pdf estimator proposed in [7, Equation 13] works only when the log-normal factors are independent, but one can extend it to the dependent case as shown in [2]. Let us denote the estimator proposed in [2, 7] by $\widehat{f_{\mathrm{A}}}$. Tables 3.4 and 3.5 below compare the two estimators on two distinct numerical examples. The results suggest that (3.3) becomes significantly more efficient than $\widehat{f_{\mathrm{A}}}$ when the X_i's have different marginal distributions. Note that, as expected, the efficiency of (3.3) deteriorates as we approach the right tail.

Finally, we confirmed that, as expected, quasi Monte Carlo again accelerates the speed of convergence of the smooth estimator (3.3) by as much as approximately $O(n^{-0.92})$. The qualitative behavior is depicted on Figure 3.4, where we also show the rate of convergence of its competitor $\hat{\ell}_{\mathrm{A}}$. Here, again, the relative error (in percent) is estimated using 100 independent random shifts of Sobol's quasirandom pointset [24, Section 2.5].

The reason that estimator (3.3) achieves a better convergence rate is that, while $\widehat{f_{\mathrm{A}}}$ is continuous, but not differentiable, the estimator (3.3) is infinitely differentiable, and hence more amenable to acceleration

Table 3.4: The SLN distribution for $d = 32, \boldsymbol{\nu} = \mathbf{0}, \rho = 0.5, \Sigma = \rho \mathbf{1}\mathbf{1}^\top + (1-\rho)\mathrm{I}$, using $n = 10^6$ samples.

			relative error %		work normalized rel. var.	
γ	$\hat{\ell}(\gamma)$	$\widehat{f}(\gamma)$	RE($\widehat{f}$)	RE($\widehat{f}_{\mathrm{A}}$)	WNRV($\widehat{f}$)	WNRV($\widehat{f}_{\mathrm{A}}$)
140	0.957	9.12×10^{-4}	0.960	0.914	7.93×10^{-4}	1.50×10^{-3}
100	0.894	2.53×10^{-3}	0.421	0.643	1.52×10^{-4}	7.42×10^{-4}
80	0.826	4.46×10^{-3}	0.260	0.538	5.84×10^{-5}	5.05×10^{-4}
60	0.705	7.96×10^{-3}	0.151	0.462	2.00×10^{-5}	3.61×10^{-4}
50	0.613	1.06×10^{-2}	0.113	0.436	1.14×10^{-5}	3.23×10^{-4}
40	0.490	1.38×10^{-2}	0.090	0.426	7.05×10^{-6}	3.08×10^{-4}
30	0.336	1.69×10^{-2}	0.084	0.444	6.23×10^{-6}	3.31×10^{-4}
20	0.163	1.71×10^{-2}	0.098	0.543	8.70×10^{-6}	4.94×10^{-4}
15	0.083	1.41×10^{-2}	0.113	0.693	1.17×10^{-5}	7.99×10^{-4}

Table 3.5: The SLN distribution for $d = 10, \rho = 0$, $\nu_i = i - d$, $\sigma_i^2 = i$, estimated with $n = 10^6$ samples.

			relative error %		work normalized rel. var.	
γ	$\hat{\ell}(\gamma)$	$\widehat{f}(\gamma)$	RE($\widehat{f}$)	RE($\widehat{f}_{\mathrm{A}}$)	WNRV($\widehat{f}$)	WNRV($\widehat{f}_{\mathrm{A}}$)
500	0.964	5.28×10^{-5}	6.22	12.0	1.09×10^{-2}	2.60×10^{-2}
100	0.881	8.01×10^{-4}	1.86	30.2	9.91×10^{-4}	1.66×10^{-1}
30	0.746	4.81×10^{-3}	0.88	13.3	2.08×10^{-4}	3.23×10^{-2}
15	0.633	1.21×10^{-2}	0.59	7.23	9.56×10^{-5}	9.52×10^{-3}
7	0.484	2.96×10^{-2}	0.39	5.26	4.26×10^{-5}	5.05×10^{-3}
3	0.310	6.58×10^{-2}	0.27	3.51	2.09×10^{-5}	2.22×10^{-3}
1	0.125	1.29×10^{-1}	0.17	2.42	8.86×10^{-6}	1.03×10^{-3}
0.5	0.0548	1.50×10^{-1}	0.14	2.24	5.56×10^{-6}	9.12×10^{-4}

with quasirandom sequences.

3.2.5 Numerical Comparison with Deterministic or Hybrid Approximations

While in this work we use the Monte Carlo method to estimate the right/left tails and pdf of the SLN distribution, there is a long history of using deterministic methods to approximate the pdf of the SLN distribution.

The most notable classical method is the Fenton-Wilkinson approximation [17], which finds the log-normal density that matches the first and second moments of the SLN distribution and delivers this log-normal density as a global approximation.

A more recent deterministic method is the skew log-normal approximation [21], which builds on the Fenton-Wilkinson idea to propose a *skew* log-normal pdf as the approximating density (instead of the simpler log-normal).

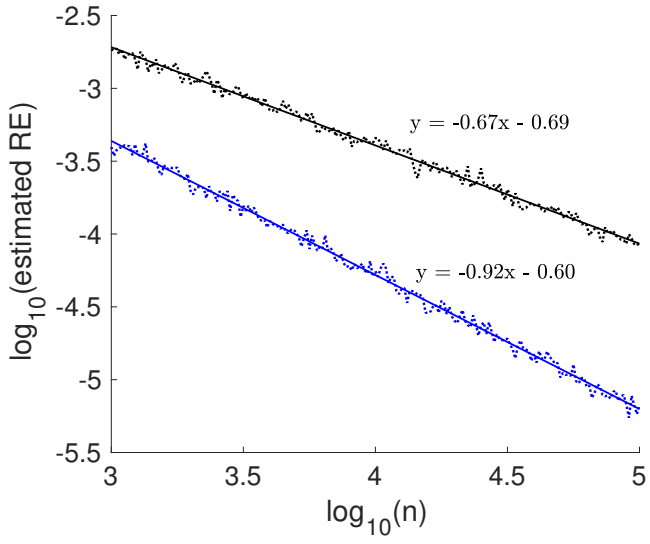


Figure 3.4: The relative error of estimator (3.3) (in blue with slope -0.92) for $\gamma = \mathbb{E}S = 5\exp(1/2), d = 5$, $\Sigma = \mathrm{I}, \boldsymbol{\nu} = \mathbf{0}$, as well as that of $\hat{f}_{\mathrm{A}}$ (in black with slope -0.67).

Finally, Asmussen et al. [5] propose an ingenious approximation based on the sum of k orthogonal Hermite polynomials, whose unknown coefficients are estimated from n Monte Carlo draws from the SLN pdf (which we wish to approximate). This method is in fact a hybrid method, as it combines features from deterministic and Monte Carlo methods.

In what follows we provide a numerical example of approximating the SLN pdf with parameters $\boldsymbol{\mu} = (4,3,2,1)^\top, \Sigma = \rho\boldsymbol{\sigma}\boldsymbol{\sigma}^\top + (1-\rho)\mathrm{I}$, where $\rho = 1/5, \boldsymbol{\sigma} = (1,2,3,4)^\top$, and discuss the pros and cons of each method.

For our pdf estimator $\hat{f}(\gamma)$ in (3.3) we take $n = 10^4$ and run 50 quasi Monte Carlo runs to estimate the pdf values as well as the margin of error for a 95% confidence intervals. For the orthogonal Hermite polynomials, we use $n = 10^7$ and experiment with $k = 10$ and $k = 25$.

The results are given in the Table 3.6 and Figure 3.5. The Fenton-Wilkinson approximation can be seen to be widely inaccurate in Figure 3.5, and for this reason is omitted from the table. Also, the Hermite polynomial approximation with $k = 25$ is indiscernible from our Monte Carlo estimate on Figure 3.5.

We make a number of comments and recommendations regarding each of the methods.

First, both the Fenton-Wilkinson and skew log-normal approximations are extremely fast and simple to implement. In all cases we tested, the skew log-normal approximation performed better than the

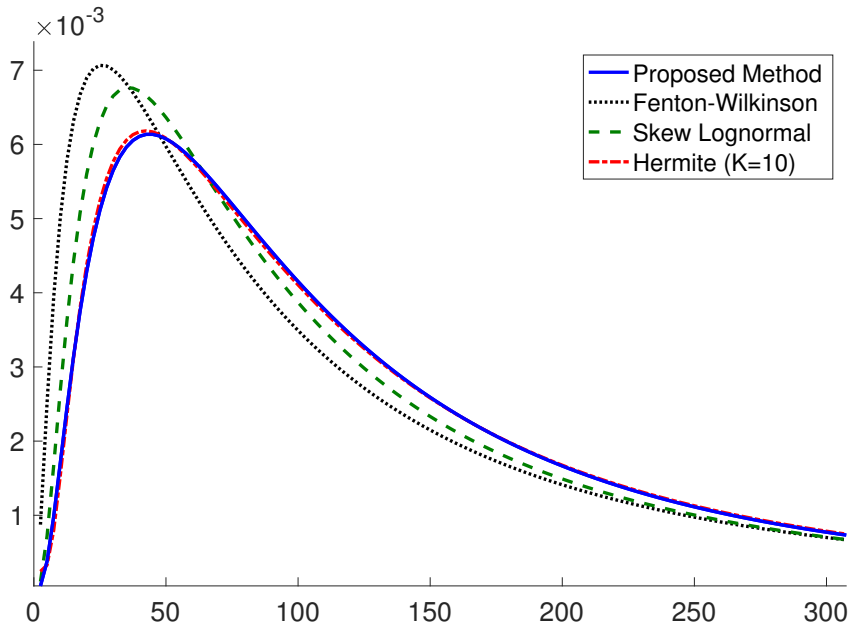


Figure 3.5: Comparison of four different methods for approximating the main body of the SLN density.

Table 3.6: Pdf estimates for $\boldsymbol{\sigma} = (1,2,3,4)^\top$, $\boldsymbol{\mu} = (4,3,2,1)^\top$, $\rho = 1/5$ and $\Sigma = \rho\boldsymbol{\sigma}\boldsymbol{\sigma}^\top + (1-\rho)\mathrm{I}$. The relative error multiplied by 1.96 (the error margin for 95% confidence interval) is displayed in brackets for the Monte Carlo estimator. No simple error estimates are available for the other methods.

γ	$\widehat{f}_\gamma$	Hermite, $k = 10$	Hermite, $k = 25$	Skew-Lognomal
5	3.79×10^{-4} (2.99×10^{-8})	3.29×10^{-4}	3.79×10^{-4}	7.19×10^{-4}
50	6.08×10^{-3} (4.11×10^{-7})	6.08×10^{-3}	6.08×10^{-3}	6.36×10^{-3}
100	4.15×10^{-3} (3.79×10^{-7})	4.11×10^{-3}	4.15×10^{-3}	3.87×10^{-3}
150	2.59×10^{-3} (3.46×10^{-7})	2.58×10^{-3}	2.59×10^{-3}	2.33×10^{-3}
200	1.66×10^{-3} (3.14×10^{-7})	1.68×10^{-3}	1.67×10^{-3}	1.49×10^{-3}
250	1.11×10^{-3} (2.79×10^{-7})	1.13×10^{-3}	1.11×10^{-3}	1.00×10^{-3}
300	7.72×10^{-4} (2.51×10^{-7})	7.85×10^{-4}	7.72×10^{-4}	7.06×10^{-4}

simpler Fenton-Wilkinson approximation.

Second, both deterministic approximations do not provide simple error estimates and can be wildly inaccurate when the SLN distribution is not a sum of iid variables, as seen in Figure 3.5. The inaccuracy of these methods is exacerbated in the right and left tail of the SLN distribution. Unless we are dealing with iid sums, we do not recommend the use of the Fenton-Wilkinson or the skew log-normal approximations.

Third, in theory the error of the Asmussen et al. [5] orthogonal series approximation can be made

arbitrarily small by increasing k (the truncation parameter in the infinite orthogonal series representation of the SLN density) and n (which improves the accuracy with which the coefficients of the series representation are estimated).

However, in practice, with a finite precision arithmetic, a large k may pose numerical instability issues. One consequence of this is that the Hermite polynomial approximation may oscillate wildly in the right tail or be inaccurate in the left tail (the approximation may take negative values in the tail).

In addition, with a fixed computational budget, there appears to be a tradeoff between two distinct sources of error — the series approximation and Monte Carlo estimation errors. A large k reduces the truncation approximation error (in the infinite series representation), but also increases the Monte Carlo estimation error, because there is a greater number of coefficients to estimate.

Finally, just like with the Fenton–Wilkonson-type approximations, there is no simple error estimate for the orthogonal series approximation, given a pair of values for k and n. In practice, one has to experiment with different combinations for k and n, and check if the approximation converges free from any numerical instability artifacts.

For the above stated reasons, our recommendation is to use both the Hermite polynomial approximation and the Monte Carlo estimator (3.3). In this way, the two methods can independently validate the accuracy of their output and ensure that there are no numerical artifacts due to numerical instability (in the orthogonal series approximation). In addition, if the sum is comprised of iid log-normals, then the Asmussen et al. [7] Monte Carlo estimator is also an excellent choice and can be used as an alternative to our Monte Carlo estimator.

3.3 Accurate Estimation of the Right Tail

In this section, we provide an estimator of the right tail of the SLN distribution, that is,

$$\mathbb{P}(X_1 + \cdots + X_d \geq \gamma), \qquad \ln \boldsymbol{X} \sim \mathsf{N}(\boldsymbol{\nu}, \Sigma),$$

that works well for many parameter settings for which all existing estimators fail. In order to keep the notation minimal, we recycle the notation for the left tail and henceforth let

$$\ell(\gamma) \stackrel{\text{def}}{=} \mathbb{P}(\exp(Y_1) + \cdots + \exp(Y_d) \geq \gamma), \qquad \boldsymbol{Y} \sim \mathsf{N}(\boldsymbol{\nu}, \Sigma).$$

Further, we set $S = X_1 + \cdots + X_d, M = \max_i X_i$, and $\sigma_i^2 = \Sigma_{ii}, \sigma = \max_k \sigma_k, \nu = \max\{\nu_k : \sigma_k = \sigma\}$. Note that with all random variates defined on the same probability space, we can write $\boldsymbol{X} = \exp(\boldsymbol{Y})$.

One of the reasons why estimating the right tail is difficult is due to the heavy-tailed behavior of $\ell(\gamma)$ as $\gamma \uparrow \infty$ (see Corollary 1 here or [9]):

$$\ell(\gamma) \simeq \ell_{\text{as}} \stackrel{\text{def}}{=} \sum_{k=1}^{d} \mathbb{P}(Y_i \geq \ln \gamma) = \sum_{k=1}^{d} \overline{\Phi}((\ln \gamma - \nu_k)/\sigma_k). \tag{3.4}$$

To tackle this problem, the authors of [4,11,19,22] propose a number of theoretically efficient estimators. The problem with these estimators, however, is that their established theoretical efficiency does not necessarily translate into estimators with reasonably low Monte Carlo variance. Before proceeding to remedy this problem, we next explain why these existing proposals can fail to estimate $\ell(\gamma)$ — we provide both numerical evidence and theoretical insight.

3.3.1 Variance Boosted Estimator

We call the first estimator proposed in the literature the *variance boosted* estimator [4, 11, 13], which is defined as follows.

Define $\mathbb{P}_\theta$ to be a change of measure such that $\boldsymbol{Y} \sim \mathsf{N}(\boldsymbol{\nu}, \Sigma/(1-\theta))$, where $\theta \in [0,1)$ is a parameter to be determined. If we take θ sufficiently close to unity, then we can inflate the variance of $\boldsymbol{Y}$ to induce the event $\{S > \gamma\}$. We thus obtain the variance boosted estimator:

$$\hat{\ell}_\theta(\gamma) = \frac{\exp(-\theta(\boldsymbol{Y}-\boldsymbol{\nu})^\top \Sigma^{-1}(\boldsymbol{Y}-\boldsymbol{\nu})/2)}{(1-\theta)^{d/2}} \mathbb{I}\{S > \gamma\}, \qquad \boldsymbol{Y} \sim \mathbb{P}_\theta. \tag{3.5}$$

One can choose θ optimally and show [4] that:

$$\frac{\mathbb{E}_\theta \hat{\ell}_\theta^2}{\ell^2} = \Theta([\ln \gamma]^{d/2+1} \gamma^{1/4}).$$

Therefore, we expect that the variance boosted estimator will only be useful for small d and γ. In contrast, in Section 3.3.3 we show that our new proposal has relative error which grows at the much slower rate of $\Theta(\ln \gamma)$, independently of d.

Suppose that all log-normals are iid with $\sigma = 0.25, \Sigma = \mathrm{I} \times \sigma^2, \boldsymbol{\nu} = \boldsymbol{0}, d = 30$. Using $n = 10^7$ replications, the (estimated) values for the probability $\ell(\gamma)$ are shown in Table 3.7 for varying γ and for the following competitors: our proposed estimator $\hat{\ell}$ in Section 3.3.3; the variance boosted $\hat{\ell}_\theta$; the *Asmussen–Kroese estimator* [8],

$$\hat{\ell}_{\text{AK}} = d\overline{\Phi}\left(\tfrac{1}{\sigma} \ln\left[(\gamma - \textstyle\sum_{j<d} X_j) \vee \max_{j<d} X_j\right]\right), \quad \ln \boldsymbol{X} \sim \mathsf{N}(\boldsymbol{0}, \sigma^2 \mathrm{I}). \tag{3.6}$$

The computing times are not given in the table, because they were all very similar for all methods (in the range of 7 to 10 seconds). The table suggests that the variance boosted estimator, true to its name, suffers from very high statistical variance.

Table 3.7: Efficiency of variance-boosted and Asmussen—Kroese methods. Column two gives the estimate $\hat{\ell}$ from Section 3.3.3 — our novel estimator.

			relative error %		
γ	$\hat{\ell}$	$\hat{\ell}_{\mathrm{AK}}$	$\mathrm{RE}(\hat{\ell})$	$\mathrm{RE}(\hat{\ell}_{\mathrm{AK}})$	$\mathrm{RE}(\hat{\ell}_{\theta})$
30	0.74	0.74	0.199	0.0321	0.314
33	0.079	0.079	0.26	0.0871	3.67
36	0.00052	0.00052	0.403	0.684	39.8
39	2.94×10^{-7}	3.31×10^{-7}	0.725	17.9	51.9
42	2.29×10^{-11}	9.23×10^{-14}	1.45	54.6	99.9
45	3.92×10^{-16}	7.78×10^{-20}	2.57	64.4	97.8
48	1.93×10^{-21}	2.13×10^{-25}	4.44	31.7	97
51	3.98×10^{-27}	2.40×10^{-29}	7.85	25.2	81.5
54	8.58×10^{-33}	3.96×10^{-33}	3.22	15.3	100
57	3.44×10^{-36}	3.07×10^{-36}	0.418	13.3	69.8
60	4.26×10^{-39}	3.86×10^{-39}	0.203	5.21	99.7
63	1.06×10^{-41}	1.01×10^{-41}	0.18	2.92	99
66	4.38×10^{-44}	4.39×10^{-44}	0.162	1.58	64.8
69	2.75×10^{-46}	2.74×10^{-46}	0.16	1.09	100
72	2.42×10^{-48}	2.40×10^{-48}	0.155	0.686	98.3
75	2.83×10^{-50}	2.81×10^{-50}	0.153	0.498	72.1
78	4.24×10^{-52}	4.21×10^{-52}	0.151	0.414	95.7
81	7.87×10^{-54}	7.86×10^{-54}	0.15	0.287	99.3
84	1.78×10^{-55}	1.78×10^{-55}	0.15	0.26	100
87	4.74×10^{-57}	4.75×10^{-57}	0.15	0.251	90.5
90	1.48×10^{-58}	1.48×10^{-58}	0.15	0.189	100

3.3.2 Vanishing Relative Error Estimator

Since the previous estimator is too variable, researchers have also studied the *importance sampling vanishing relative error* (ISVE) estimator [4, 11], in which the probability ℓ is represented as:

$$\ell = \mathbb{P}(M > \gamma) + \mathbb{P}(S > \gamma, M < \gamma).$$

With this representation, the terms $\ell_1 = \mathbb{P}(M > \gamma)$ and $\ell_2 = \mathbb{P}(S > \gamma, M < \gamma)$ are estimated via two different estimators. The estimator for ℓ_1 is

$$\hat{\ell}_1 = \frac{\ell_{\mathrm{as}}(\gamma)}{\sum_{k=1}^{d} \mathbb{I}\{X_k > \gamma\}}, \quad \boldsymbol{X} \sim g(\boldsymbol{x}), \tag{3.7}$$

where $\boldsymbol{X}$ is simulated from:

$$g(\boldsymbol{x}) \overset{\text{def}}{=} \frac{\phi_{\Sigma}(\boldsymbol{x} - \boldsymbol{\nu}) \sum_{k=1}^{d} \mathbb{I}\{x_k > \gamma\}}{\ell_{\mathrm{as}}(\gamma)}. \tag{3.8}$$

For the second term we use the estimator with $\theta = 1 - \Theta(\ln^{-2}(\gamma))$:

$$\hat{\ell}_{2,\theta}(\gamma) = \frac{\exp(-\theta(\boldsymbol{Y}-\boldsymbol{\nu})^\top \Sigma^{-1}(\boldsymbol{Y}-\boldsymbol{\nu})/2)}{(1-\theta)^{d/2}} \mathbb{I}\{S > \gamma, M < \gamma\}, \qquad \boldsymbol{Y} \sim \mathbb{P}_\theta, \tag{3.9}$$

which is of the variance-boosted type. Hence, we obtain the ISVE Monte Carlo estimator:

$$\hat{\ell}_{\mathrm{ISVE}} = \hat{\ell}_1 + \hat{\ell}_2,$$

which has been shown [4] to have vanishing relative error:

$$\frac{\mathbb{V}\mathrm{ar}(\hat{\ell}_{\mathrm{ISVE}})}{\ell^2(\gamma)} \downarrow 0, \quad \gamma \uparrow \infty.$$

We now describe one problem with this estimator.

In practical simulations one estimates the precision of an estimator $\hat{l}$ of ℓ by first generating n independent realizations $\hat{l}_1, \ldots, \hat{l}_n$, and then computing the sample variance of the estimator $S_n^2 = \frac{1}{n}\sum_{i=1}^n (\hat{l}_i - \bar{l})^2$, where $\bar{l}_n = (\hat{l}_1 + \cdots + \hat{l}_n)/n$. Generally, it is typical to use this to estimate the relative error of the estimator or to construct confidence intervals for the quantity of interest. However, the sample variance of n independent replications of (3.7) is not an accurate estimator of the variance of $\hat{\ell}_1$, and so these simple precision estimates cannot be applied to $\hat{\ell}_{\mathrm{ISVE}}$. This is formalized in the following result, proved in the Appendix.

Lemma 1 (Variance Estimate Inefficiency)**.** Suppose S_n^2 is the sample variance computed from n iid versions of (3.7). It follows that S_n^2 is not a logarithmically efficient estimator:

$$\limsup_{\gamma\uparrow\infty} \frac{\ln \mathbb{V}\mathrm{ar}(S_n^2)}{\ln \mathbb{V}\mathrm{ar}(\hat{\ell}_1)} < 2,$$

so that the relative error in estimating the precision of $\hat{\ell}_{\mathrm{ISVE}}$ grows at an exponential rate in $\ln^2(\gamma)$.

This lemma suggests that the relative error of $\hat{\ell}_{\mathrm{ISVE}}$ can be severely underestimated in practice. In other words, the ISVE estimator is not second-order efficient. In contrast to this negative result for the ISVE estimator, in Corollary 2 we show that our new estimator enjoys an asymptotically efficient estimator of its true variability. Even better, estimation of the precision of our estimator is not more difficult than estimating ℓ itself.

3.3.3 Exponentially Tilted Estimator

Given the failure of the estimators described above, a natural question arises. What kind of estimator will succeed in being both theoretically efficient and exhibit low variance in practical simulations?

To answer this question we start by examining the quite natural proposal of Gulisashvili and Tankov (GT) [19, Equation (70)], which can be written as follows:

$$\hat{\ell}_{\mathrm{GT}} \stackrel{\mathrm{def}}{=} \exp\left(\frac{\boldsymbol{\mu}^\top \Sigma^{-1} \boldsymbol{\mu}}{2} - \boldsymbol{\mu}^\top \Sigma^{-1}(\boldsymbol{Y} - \boldsymbol{\nu})\right) \mathbb{I}\{S \geq \gamma\}, \qquad \boldsymbol{Y} \sim \mathsf{N}(\boldsymbol{\mu} + \boldsymbol{\nu}, \Sigma), \tag{3.10}$$

where the parameter $\boldsymbol{\mu}$ is chosen by minimizing an asymptotic approximation to the second moment [19, Equation (71)].

Unfortunately, (3.10) also performs poorly, just like the estimators in the last section.[3] This poor practical performance is compounded by the fact that there is no proof of the asymptotic efficiency of (3.10) as $\gamma \uparrow \infty$ (see [19, Page 40]).

The reason why the estimator (3.10) performs poorly is that it uses a *single* exponential tilting parameter $\boldsymbol{\mu}$, which is insufficient to induce the mutually-exclusive mode of occurrence of the rare-event: $\mathbb{P}(X_k = M | S > \gamma) \simeq \frac{\mathbb{P}(X_k > \gamma)}{\ell}$ (see part 1 of Lemma 2). In other words, with asymptotic probability $\mathbb{P}(X_k > M)/\ell$, each X_k is the maximal term that causes the sum to up-cross γ, and the single exponential tilting parameter $\boldsymbol{\mu}$ in (3.10) cannot account for this mutually-exclusive behavior.

Instead, to obtain a provably efficient estimator with excellent practical and theoretical performance, we must introduce d distinct exponential tilting parameter vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_d$, where each $\boldsymbol{\mu}_k$ is tasked to deal with the event $\{S > \gamma, X_k = M\}$. The new set of d tilting parameters are also determined using an error estimate different from the one used in (3.10) when we have a single tilting parameter. Thus, our proposal uses an estimator of the form (3.10) for each term, $\hbar_i(\gamma)$, in the decomposition:

$$\ell(\gamma) = \sum_{i=1}^{d} \underbrace{\mathbb{P}(S > \gamma, X_i = M)}_{\hbar_i(\gamma)}.$$

The estimator of each $\hbar_k$ based on one replication is

$$\hat{\hbar}_k(\gamma) = \exp\left(\frac{\boldsymbol{\mu}_k^\top \Sigma^{-1} \boldsymbol{\mu}_k}{2} - \boldsymbol{\mu}_k^\top \Sigma^{-1}(\boldsymbol{Y} - \boldsymbol{\nu})\right) \mathbb{I}\{S > \gamma, X_k = M\}, \tag{3.11}$$

where under the measure $\mathbb{P}_{\boldsymbol{\mu}_k}$ with expectation $\mathbb{E}_{\boldsymbol{\mu}_k}$ we have $\ln \boldsymbol{X} = \boldsymbol{Y} \sim \mathsf{N}(\boldsymbol{\nu} + \boldsymbol{\mu}_k, \Sigma)$, and $\boldsymbol{\mu}_k$ is the solution to the non-linear optimization:

$$\min_{\boldsymbol{\mu}} \frac{1}{2} \boldsymbol{\mu}^\top \Sigma^{-1} \boldsymbol{\mu}, \tag{3.12}$$

[3] We remark that the GT estimator applies to the more general setting of sums *and differences* of log-normals. This generality of the GT estimator, however, comes at the cost of not being the most efficient estimator for sums — the case we consider here.

subject to:

$$\begin{aligned} g_1(\boldsymbol{\mu}) &= \exp(\mu_k + \nu_k) + \textstyle\sum_{i\neq k} \exp(\mu_i + \nu_i + \frac{\sigma_i^2}{2}) - \gamma \geq 0 \\ g_2(\boldsymbol{\mu}) &= \mu_k + \nu_k + \tfrac{\sigma_k^2}{2} - \max_{j\neq k}\{\mu_j + \nu_j + \tfrac{\sigma_j^2}{2}\} \geq 0 \end{aligned} \tag{3.13}$$

To construct the overall estimator $\hat{\ell}$ of ℓ, we can use stratification with a total computing budget of $n = n_1 + \cdots + n_d$ replications, whereby we allocate n_k samples to estimate each $\hbar_k$ independently. We then take the sum of the estimators of $\hbar_k$'s as our stratified estimator of ℓ. In other words,

$$\hat{\ell}(\gamma) = \sum_{k=1}^{d} \frac{1}{n_k} \sum_{j=1}^{n_k} \hat{\hbar}_{k,j}(\gamma), \tag{3.14}$$

where $\sum_k n_k = n$ and $\hat{\hbar}_{k,1}, \ldots, \hat{\hbar}_{k,n_k}$ are iid copies of (3.11). We then have the following efficiency result.

Theorem 2 (Logarithmic Efficiency)**.** Suppose we select the stratified allocation such that $n_i \propto n \times \mathbb{P}(X_i > \gamma)$. Then, the estimator (3.14) is unbiased and logarithmically efficient with relative error $\mathbb{V}\mathrm{ar}(\hat{\ell}(\gamma))/\ell^2(\gamma) = O(\ln \gamma)$ as $\gamma \uparrow \infty$.

Proof. First note that choosing $n_k = n \times \mathbb{P}(X_k > \gamma)/\ell_{\mathrm{as}}$ satisfies the constraint $n = \sum_k n_k$, but is in conflict with the constraint that the n_k's have to be integers. One simple solution is to simply round up to the nearest integer, and violate the constraint $n = \sum_k n_k$. For large enough n, the residual $n - \sum_k n_k$ will be negligible. Another solution, which we adopt in our computer implementation, is to use a widely-used *randomized* stratification scheme, as described in, for example, [24, Algorithm 14.2].

Next, with the above allocation for each n_k, the variance of the stratified estimator (3.14) is:

$$\mathbb{V}\mathrm{ar}(\hat{\ell}) = \frac{1}{n} \sum_{k=1}^{d} \frac{n}{n_k} \mathbb{V}\mathrm{ar}(\hat{\hbar}_k) = \frac{\ell_{\mathrm{as}}}{n} \sum_{k=1}^{d} \frac{\mathbb{V}\mathrm{ar}(\hat{\hbar}_k)}{\mathbb{P}(X_k > \gamma)}.$$

Therefore, using the result 3. in Lemma 2, the relative error of $\hat{\ell}$ as $\gamma \uparrow \infty$ is

$$\frac{n\mathbb{V}\mathrm{ar}(\hat{\ell}(\gamma))}{\ell^2(\gamma)} \simeq \frac{n\mathbb{V}\mathrm{ar}(\hat{\ell}(\gamma))}{\ell_{\mathrm{as}}^2(\gamma)} = \frac{1}{\ell_{\mathrm{as}}(\gamma)} \sum_{k=1}^{d} \mathbb{P}(X_k > \gamma) \frac{\mathbb{V}\mathrm{ar}(\hat{\hbar}_k)}{[\mathbb{P}(X_k > \gamma)]^2} = O(\ln \gamma). \tag{3.15}$$

The last equation shows that $\frac{\ln \mathbb{V}\mathrm{ar}(\hat{\ell}^2(\gamma))}{\ln \ell(\gamma)} \to 2$ as $\gamma \uparrow \infty$. □

We can now see that the rate of growth of the relative error of our estimator, namely $O(\ln \gamma)$, is significantly slower than the rate of growth of the variance boosted estimator, $O([\ln \gamma]^{d/2+1} \gamma^{1/4})$.

Since the proof of the following lemma is long, it is delegated to the appendix.

Lemma 2 (Asymptotics for $\hat{\hbar}_k$)**.** As $\gamma \uparrow \infty$, we have that:

1. $\mathbb{E}_{\boldsymbol{\mu}}[\hat{\hbar}_k(\gamma)] = \mathbb{P}(S > \gamma, X_k = M) \simeq \mathbb{P}(X_k > \gamma)$.
2. The asymptotic solution to (3.12) is
$$\boldsymbol{\mu}^* = \frac{\ln(\gamma) - \nu_k}{\sigma_k^2} \Sigma \boldsymbol{e}_k,$$
where $\boldsymbol{e}_k$ is the unit vector with 1 in the k-th position.
3. We have $\frac{\mathbb{V}\mathrm{ar}(\hat{\hbar}_k)}{[\mathbb{P}(X_k > \gamma)]^2} = O(\ln \gamma)$, and with $\boldsymbol{\mu}$ solving (3.12) the $(m+1)$-st moment satisfies:
$$\mathbb{E}_{\boldsymbol{\mu}} \hat{\hbar}_k^{m+1} = \Theta(\ln^m(\gamma) \hbar_k^{m+1}). \tag{3.16}$$

Note that part 1. of the above lemma immediately yields the following corollary, which was originally proved in [9] using a different argument.

Corollary 1 (Right-Tail Asymptotics)**.** $\ell(\gamma) \simeq \sum_{i=1}^d \mathbb{P}(X_k > \gamma)$ *as* $\gamma \uparrow \infty$.

More importantly, part 3. of Lemma 2 gives us a robustness guarantee that is not enjoyed by any of the competing estimators.

Corollary 2 (Logarithmically Efficient Variance Estimator)**.** *Let* $S_{n_k}^2$ *be the sample variance based on* n_k *independent replications of* (3.11)*. Then,* S_n^2 *is a logarithmically efficient estimator:*
$$\liminf_{\gamma \uparrow \infty} \frac{\ln \mathbb{V}\mathrm{ar}(S_{n_k}^2)}{\ln \mathbb{V}\mathrm{ar}(\hat{\hbar}_k)} = 2,$$
where the rate of growth is $\frac{\mathbb{V}\mathrm{ar}(S_{n_k}^2)}{\mathbb{V}\mathrm{ar}^2(\hat{\hbar}_k)} = O(\ln \gamma)$.

Proof. Using (3.16), consider the following calculations:
$$\begin{aligned}
\frac{n_k \mathbb{V}\mathrm{ar}(S_{n_k}^2)}{\mathbb{V}\mathrm{ar}^2(\hat{\hbar}_k)} &= \frac{\mathbb{E}_{\boldsymbol{\mu}}(\hat{\hbar}_k(\gamma) - \hbar_k(\gamma))^4}{[\mathbb{E}_{\boldsymbol{\mu}}(\hat{\hbar}_k(\gamma) - \hbar_k(\gamma))^2]^2} - 1 + \frac{2}{n_k - 1} \\
&= \frac{\Theta(\ln^3(\gamma)\hbar_k^4) + \hbar_k^4 + \hbar_k^2 \Theta(\ln(\gamma)\hbar_k^2) - \hbar_k \Theta(\ln^2(\gamma)\hbar_k^3) - 4\hbar_k^4}{[\Theta(\ln(\gamma)\hbar_k^2) - \hbar_k^2]^2} - 1 + \frac{2}{n_k - 1} \\
&= \frac{\Theta(\ln^3 \gamma) + \Theta(\ln \gamma) - \Theta(\ln^2 \gamma) - 3}{[\Theta(\ln \gamma) - 1]^2} - 1 + \frac{2}{n_k - 1} \\
&= \Theta(\ln(\gamma)) - 1 + \frac{2}{n_k - 1}.
\end{aligned}$$
□

Therefore, a major advantage of our proposed estimator (3.14) is that estimating its variance is asymptotically not more difficult than estimating ℓ itself.

In retrospect, we can see that the excellent theoretical properties of our estimator are due mainly to the breaking of the symmetry in the sum $S = X_1 + \cdots + X_d$ by distinguishing each and every X_i as the overall maximum. In contrast, the poorly performing estimators (3.10) and (3.9) (and hence $\hat{\ell}_{\mathrm{ISVE}}$) both induce a simple change of measure that does not conform to the mutually-exclusive asymptotic behavior of $\mathbb{P}(S > \gamma, X_k = M) \simeq \mathbb{P}(X_k > \gamma),\ k = 1, \ldots, d$.

Finally, we remark on the unusual way of selecting $\boldsymbol{\mu}$ via the optimization (3.12). Why do we not simply use the asymptotic approximation $\boldsymbol{\mu}^*$ in Lemma 2? The answer is that, while asymptotically the matrix Σ is irrelevant, it is still relevant for very large values of γ, and our change of measure should reflect this dependence. The asymptotic solution $\boldsymbol{\mu}^*$ does not reflect this dependence. Thus, (3.12) was designed with two objectives in mind: (1) good practical performance for finite $\gamma < \infty$, where the full Σ is relevant; (2) asymptotic optimality as $\gamma \uparrow \infty$, where Σ is irrelevant. The optimization program (3.12) transitions from objective (1) to objective (2) in a continuous way.

3.3.4 Numerical Comparison

In this section we show that the better theoretical properties of (3.14) convert into excellent practical performance.

Comparison with ISVE estimator

For the first example we use

$$d = 30, \rho = 0.9,\ \boldsymbol{\nu} = \mathbf{0},\ \Sigma = 0.25^2 \times (\rho \times \mathbf{1}\mathbf{1}^\top + (1-\rho) \times \mathrm{I}).$$

Table 3.8 gives the results using $n = 10^6$ replications for different values of γ. We tried to maximize the accuracy of the ISVE estimator by experimentally choosing the best θ (see brackets, column three).

From the table we can observe that the variance of the ISVE estimator is large for almost any γ. Further, in the last experiment with $\gamma = 10^4$, despite our best effort at selecting an optimal θ via careful experimental tuning, we were not able to induce the event $\{S > \gamma, M < \gamma\}$. The event $\{S > \gamma, M < \gamma\}$ simply persists in being rare whatever $\theta \in [0, 1)$ we choose. In other words, $\hat{\ell}_{\mathrm{ISVE}} = \hat{\ell}_1 + \hat{\ell}_2$ is with high probability equal to $\hat{\ell}_1$.

In summary, the performance of the ISVE estimator is consistent with an estimator that lacks second-order efficiency, and even worsens its performance for large d (number of log-normals in the sum). In

Table 3.8: Efficiency of ISVE and exponentially tilted estimators for $\rho = 0.9$.

			relative error %		work normalized rel. var.	
γ	$\hat{\ell}$	$\hat{\ell}_{\mathrm{ISVE}}$	$\mathrm{RE}(\hat{\ell})$	$\mathrm{RE}(\hat{\ell}_{\mathrm{ISVE}})$	$\mathrm{WNRV}(\hat{\ell})$	$\mathrm{WNRV}(\hat{\ell}_{\mathrm{ISVE}})$
40	0.116	0.114 ($\theta = 0.5$)	0.63	2.0	0.00032	0.00080
100	2.17×10^{-7}	1.18×10^{-7} ($\theta = 0.6$)	0.98	40	0.00061	0.31
150	6.83×10^{-12}	5.75×10^{-13} ($\theta = 0.75$)	1.1	84	0.00093	1.12
200	7.75×10^{-16}	2.09×10^{-17} ($\theta = 0.8$)	1.2	95	0.0010	1.22
400	6.57×10^{-28}	3.08×10^{-39} ($\theta = 0.9$)	1.4	80	0.0011	1.34
10^3	1.61×10^{-49}	1.21×10^{-80} ($\theta = 0.95$)	1.7	100	0.002	2.02
10^4	3.60×10^{-132}	1.80×10^{-294} ($\theta = ?$)	2.1	-	0.0024	-

contrast, the next example in Table 3.9 suggests that our estimator remains robust even in very high dimensions of up to $d = 60$.

Table 3.9: Performance for $d = 60$, $n = 10^6$, $\boldsymbol{\nu} = \mathbf{0}$, $\Sigma = 0.5\times\mathbf{1}\mathbf{1}^\top + 0.5\times\mathrm{I}$.

			relative error %		work normalized rel. var.	
γ	$\widehat{\ell}$	$\widehat{\ell}_{\mathrm{ISVE}}$	$\mathrm{RE}(\widehat{\ell})$	$\mathrm{RE}(\widehat{\ell}_{\mathrm{ISVE}})$	$\mathrm{WNRV}(\widehat{\ell})$	$\mathrm{WNRV}(\widehat{\ell}_{\mathrm{ISVE}})$
600	1.98×10^{-3}	5.77×10^{-7}	0.837	51.6	1.02×10^{-3}	4.88
900	2.81×10^{-4}	4.00×10^{-10}	0.893	15.4	1.20×10^{-3}	4.42×10^{-1}
1200	5.91×10^{-5}	4.16×10^{-11}	0.93	3.16	1.36×10^{-3}	1.75×10^{-2}
1500	1.57×10^{-5}	7.92×10^{-12}	0.964	1.23	1.52×10^{-3}	2.56×10^{-3}
1800	5.01×10^{-6}	2.01×10^{-12}	0.987	1.50	1.57×10^{-3}	3.41×10^{-3}
2100	1.79×10^{-6}	6.05×10^{-13}	1.012	0.0543	2.03×10^{-3}	5.10×10^{-6}
2400	7.18×10^{-7}	2.12×10^{-13}	1.029	2.84×10^{-3}	1.56×10^{-3}	1.11×10^{-8}
2700	3.08×10^{-7}	8.30×10^{-14}	1.046	8.93×10^{-4}	1.63×10^{-3}	1.15×10^{-9}
3000	1.44×10^{-7}	3.54×10^{-14}	1.057	6.82×10^{-4}	2.02×10^{-3}	5.97×10^{-10}
3300	7.02×10^{-8}	1.63×10^{-14}	1.069	8.30×10^{-4}	2.05×10^{-3}	9.60×10^{-10}

Comparison with Modified Asmussen–Kroese estimator

In addition to the ISVE estimator, the *modified Asmussen–Kroese* (MAK) estimator [22, Equation 3.6] also enjoys the vanishing relative error property. In comparing (3.14) with the MAK estimator, we make the following observations.

First, the MAK estimator requires the solution of a non-linear equation for every replication. This aspect of the estimator poses nontrivial problems: (1) sometimes no solution exists; (2) Newton's method may take many iterations to converge, making the running time of the estimator large.

Table 3.10: Comparative performance for $d = 10, n = 10^6, \boldsymbol{\nu} = \mathbf{0}, \rho = 0.2, \Sigma = 0.25^2(\rho \mathbf{1}\mathbf{1}^\top + (1-\rho)\mathrm{I})$.

			relative error %		work normalized rel. var.	
γ	$\hat{\ell}_{\mathrm{MAK}}$	$\hat{\ell}$	$\mathrm{RE}(\hat{\ell}_{\mathrm{MAK}})$	$\mathrm{RE}(\hat{\ell})$	$\mathrm{WNRV}(\hat{\ell}_{\mathrm{MAK}})$	$\mathrm{WNRV}(\hat{\ell})$
15	0.00195	0.00198	0.420	0.669	0.00531	4.15×10^{-5}
16	0.000373	0.000370	0.660	0.724	0.0130	5.03×10^{-5}
17	6.47×10^{-5}	6.47×10^{-5}	1.07	0.775	0.0349	5.20×10^{-5}
18	1.00×10^{-5}	1.02×10^{-5}	1.80	0.823	0.096	0.000102
19	1.57×10^{-6}	1.52×10^{-6}	3.10	0.87	0.28	6.71×10^{-5}
20	2.02×10^{-7}	2.15×10^{-7}	5.60	0.937	0.941	8.09×10^{-5}
21	3.11×10^{-8}	2.99×10^{-8}	9.2	1.00	2.56	0.000166
22	3.80×10^{-9}	3.91×10^{-9}	15.9	1.06	7.58	0.000108
23	3.22×10^{-10}	5.15×10^{-10}	19.0	1.07	11.2	0.000327
24	5.63×10^{-11}	6.61×10^{-11}	44.7	1.14	61.5	0.000123
25	6.09×10^{-12}	8.42×10^{-12}	42.0	1.18	55.0	0.00036
26	4.63×10^{-13}	1.05×10^{-12}	71.6	1.23	159	0.000197
27	1.90×10^{-14}	1.33×10^{-13}	31.5	1.40	30.0	0.000176
28	4.85×10^{-15}	1.69×10^{-14}	60.0	1.49	110	0.000187
29	9.12×10^{-17}	2.15×10^{-15}	60.4	1.5	113	0.000213
30	6.37×10^{-18}	2.74×10^{-16}	53.1	1.54	87.5	0.000188

Second, while our estimator (3.14) was shown to be second-order efficient, ensuring reliable estimation of its precision, no such efficiency result is provided for the MAK estimator, and in numerical experiments we sometimes observed significant underestimation of the true variance of the MAK estimator.

Third, observe that the MAK estimator reduces to the Asmussen–Kroese (AK) estimator (3.6) in the independent case: $\boldsymbol{\nu} = \nu\mathbf{1}, \Sigma = \sigma^2\mathrm{I}$. Table 3.7 shows that when σ is small our estimator can outperform the (modified) Asmussen-Kroese estimator by orders of magnitude. For example, note how for $\gamma \in [42, 51]$ the AK estimator severely underestimates the true probability by as much as an order of 10^{-4}. Interestingly, the Asmussen-Kroese estimator has superior and unrivaled performance only in cases with larger σ, say $\sigma \geq 1$.

Finally, Table 3.10 confirms that the MAK estimator inherits the poor performance of the Asmussen–Kroese estimator for small σ. In this particular example we use

$$d = 10,\ \boldsymbol{\nu} = \mathbf{0},\ \rho = 0.2,\ \Sigma = 0.25^2(\rho \mathbf{1}\mathbf{1}^\top + (1-\rho)\mathrm{I}).$$

Observe how for $\gamma \in [26, 30]$ the MAK estimator underestimates the true probability by as much as an order of 10^{-3}.

3.4 Conclusions

In this paper, we propose a number of novel Monte Carlo estimators for the cdf, pdf, and tails of the SLN distribution. In the cdf and pdf cases, we propose estimators that permit variance reduction via quasi Monte Carlo. In the right-tail case, we propose an exponentially tilted estimator that performs well for certain parameter settings that are currently intractable with existing methods. This new exponentially tilted estimator is shown to be, not only logarithmically/weakly efficient, but also second-order efficient. This permits us to have greater confidence in all error estimates derived from simulation.

One of the crucial observations drawn from our experiments is that sometimes a strongly efficient estimator ($\hat{\ell}_{\mathrm{ISVE}}$, $\hat{\ell}_{\mathrm{MAK}}$) can have high variance in practical settings, and be bettered by a much simpler weakly efficient estimator.

In subsequent work, we will explore using the sequential sampling ideas in Section 3.2 for the estimation of the distribution of the sum of dependent random variables with an arbitrary distribution (not just log-normal).

3.5 Appendix: Proofs

Proof of Theorem 1

Proof. To proceed with the proof we recall the following three facts. First, note that $\ell(\gamma) = \mathbb{P}(\mathbf{1}^\top \exp(\boldsymbol{Y}) \leq \gamma)$, where $\boldsymbol{Y} = \boldsymbol{\nu} + \mathrm{L}\boldsymbol{Z}$. Using Jensen's inequality, we have that for any $\boldsymbol{w} \in \mathcal{W}$:

$$\begin{aligned} \ell(\gamma) &= \mathbb{P}(\boldsymbol{w}^\top \exp(\boldsymbol{Y} - \ln \boldsymbol{w}) \leq \gamma) \leq \mathbb{P}(\boldsymbol{w}^\top \ln(\boldsymbol{w}) - \boldsymbol{w}^\top \boldsymbol{Y} \geq -\ln \gamma) \\ &\leq \overline{\Phi}\left(\tfrac{\boldsymbol{w}^\top \boldsymbol{\nu} - \ln\gamma - \boldsymbol{w}^\top \ln \boldsymbol{w}}{\sqrt{\boldsymbol{w}^\top \Sigma \boldsymbol{w}}}\right) \leq \exp\left(-\tfrac{(\boldsymbol{w}^\top \boldsymbol{\nu} - \ln\gamma - \boldsymbol{w}^\top \ln \boldsymbol{w})^2}{2\boldsymbol{w}^\top \Sigma \boldsymbol{w}}\right) \end{aligned} \tag{3.17}$$

Second, denote $\bar{\boldsymbol{w}} = \mathrm{argmin}_{\boldsymbol{w}\in\mathcal{W}} \boldsymbol{w}^\top \Sigma \boldsymbol{w}$ and the set $C_\gamma \equiv \{\boldsymbol{z} : \mathbf{1}^\top \exp(\mathrm{L}\boldsymbol{z} + \boldsymbol{\nu}) \leq \gamma\}$. Then, we have the asymptotic formula, proved in [19, Formulas (13) and (63)]:

$$\ln \ell^2(\gamma) \simeq c_1 - \frac{(\ln(\gamma) - \bar{\boldsymbol{w}}^\top \boldsymbol{\nu} + \bar{\boldsymbol{w}}^\top \ln \bar{\boldsymbol{w}})^2}{\bar{\boldsymbol{w}}^\top \Sigma \bar{\boldsymbol{w}}} - (1+d)\ln(-\ln\gamma), \qquad \gamma \downarrow 0, \tag{3.18}$$

where c_1 is a constant, independent of γ. Thirdly, consider the nonlinear optimization

$$\bar{\boldsymbol{\mu}} = \underset{\boldsymbol{\mu}}{\mathrm{argmin}} \left\{ \|\boldsymbol{\mu}\|^2 - \frac{(\ln(\gamma) - \bar{\boldsymbol{w}}^\top (\boldsymbol{\nu} - \mathrm{L}\boldsymbol{\mu}) + \bar{\boldsymbol{w}}^\top \ln \bar{\boldsymbol{w}})^2}{2\bar{\boldsymbol{w}}^\top \Sigma \bar{\boldsymbol{w}}} \right\} \tag{3.19}$$

with explicit solution

$$\bar{\boldsymbol{\mu}} = \frac{\ln\gamma - \bar{\boldsymbol{w}}^\top \boldsymbol{\nu} + \bar{\boldsymbol{w}}^\top \ln \bar{\boldsymbol{w}}}{\bar{\boldsymbol{w}}^\top \Sigma \bar{\boldsymbol{w}}} \mathrm{L}^\top \bar{\boldsymbol{w}} \tag{3.20}$$

Then, we obtain the following bound on the second moment:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\mu}^*}\hat{\ell}^2(\gamma) = \mathbb{E}_{\boldsymbol{\mu}^*}\exp(2\psi(\boldsymbol{Z};\boldsymbol{\mu}^*)) &= \mathbb{E}\exp(\psi(\boldsymbol{Z};\boldsymbol{\mu}^*))\mathbb{I}\{\boldsymbol{Z}\in C_\gamma\} \\
&= \mathbb{E}\exp(\|\boldsymbol{\mu}^*\|_2^2)\mathbb{I}\{(\boldsymbol{Z}-\boldsymbol{\mu}^*)\in C_\gamma\}\textstyle\prod_j \overline{\Phi}(\mu_j^* - \alpha_j(\boldsymbol{Z}-\boldsymbol{\mu}^*)) \\
&\le \exp(\|\boldsymbol{\mu}^*\|_2^2)\mathbb{P}((\boldsymbol{Z}-\boldsymbol{\mu}^*)\in C_\gamma) \\
\text{using (3.17)}\quad &\le \exp(\|\boldsymbol{\mu}^*\|_2^2)\overline{\Phi}\left(\tfrac{(\boldsymbol{\nu}-\mathrm{L}\boldsymbol{\mu}^*)^\top\boldsymbol{w}^* - \ln\gamma - (\boldsymbol{w}^*)^\top\ln\boldsymbol{w}^*}{\sqrt{(\boldsymbol{w}^*)^\top\Sigma\boldsymbol{w}^*}}\right) \\
\text{via (3.17)+(3.19)}\quad &\le \exp\left(\|\bar{\boldsymbol{\mu}}\|_2^2 - \tfrac{(\bar{\boldsymbol{w}}^\top(\boldsymbol{\nu}-\mathrm{L}\bar{\boldsymbol{\mu}}) - \ln\gamma - \bar{\boldsymbol{w}}^\top\ln\bar{\boldsymbol{w}})^2}{2\bar{\boldsymbol{w}}^\top\Sigma\bar{\boldsymbol{w}}}\right)
\end{aligned}$$

By substituting (3.20) in the last line, we obtain the upper bound

$$\mathbb{E}_{\boldsymbol{\mu}^*}\hat{\ell}^2 \le \exp\left(-\tfrac{(\ln\gamma - \bar{\boldsymbol{w}}^\top\boldsymbol{\nu} + \bar{\boldsymbol{w}}^\top\ln\bar{\boldsymbol{w}})^2}{\bar{\boldsymbol{w}}^\top\Sigma\bar{\boldsymbol{w}}}\right)$$

In other words, from (3.18) we deduce that

$$\frac{\mathbb{E}_{\boldsymbol{\mu}^*}\hat{\ell}^2(\gamma)}{\ell^2(\gamma)} = O((-\ln\gamma)^{(d+1)}), \qquad \gamma\downarrow 0$$

and therefore

$$\liminf_{\gamma\downarrow 0}\frac{\ln\mathbb{E}_{\boldsymbol{\mu}^*}\hat{\ell}^2(\gamma)}{\ln\ell(\gamma)} = 2,$$

which implies that the algorithm is logarithmically efficient with respect to γ. □

Proof of Lemma 1

Proof. Let $N \overset{\text{def}}{=} \sum_{i=1}^d \mathbb{I}\{X_i > \gamma\}$, so that $\ell_1(\gamma) = \mathbb{P}(N \ge 1) \simeq \ell_{\text{as}}$ and the residual

$$r(\gamma) \overset{\text{def}}{=} \ell_{\text{as}} - \ell_1(\gamma) = \textstyle\sum_{i<j}\mathbb{P}(X_i > \gamma, X_j > \gamma) + o\left(\sum_{i<j}\mathbb{P}(X_i > \gamma, X_j > \gamma)\right).$$

Note that $\mathbb{P}(N > 1) = \Theta(r(\gamma))$ and $\mathbb{P}_g(N = 1) = \mathbb{P}(N = 1)/\ell_{\text{as}}(\gamma) = \Theta(1)$, where g is the mixture density defined in (3.8). We thus obtain

$$\begin{aligned}
\mathbb{E}_g\left|\hat{\ell}_1 - \ell_1(\gamma)\right|^m &= \sum_{j=1}^d \mathbb{E}_g\left[\left|\hat{\ell}_1 - \ell_1(\gamma)\right|^m \mathbb{I}\{N = j\}\right] \\
&= |\ell_{\text{as}}(\gamma) - \ell_1(\gamma)|^m\mathbb{P}_g(N=1) + \sum_{j=2}^d\left|\frac{\ell_{\text{as}}(\gamma)}{j} - \ell_1(\gamma)\right|^m\mathbb{P}_g(N=j) \\
&= r^m(\gamma)\mathbb{P}_g(N=1) + \Theta(\ell_{\text{as}}^m)\mathbb{P}_g(N>1) \\
&= r^m(\gamma)\mathbb{P}_g(N=1) + \Theta(\ell_{\text{as}}^{m-1})\mathbb{P}(N>1) \\
&= \Theta(r^m(\gamma)) + \Theta\left(\ell_{\text{as}}^{m-1}r(\gamma)\right).
\end{aligned}$$

Therefore, since $r(\gamma) = o(\ell_{\text{as}}(\gamma))$, we have:

$$\begin{aligned} n\mathbb{V}\text{ar}(S_n^2) &= \mathbb{E}_g(\hat{\ell}_1 - \ell_1(\gamma))^4 + \left(\frac{2}{n-1} - 1\right)\mathbb{V}\text{ar}^2(\hat{\ell}_1) \\ &= \Theta(r^4) + \Theta(\ell_{\text{as}}^3(\gamma)r(\gamma)) + \Theta(\mathbb{V}\text{ar}^2(\hat{\ell}_1)) \\ &= \Theta(\ell_{\text{as}}^3(\gamma)r(\gamma)) + \Theta(\mathbb{V}\text{ar}^2(\hat{\ell}_1)), \end{aligned}$$

and

$$\mathbb{V}\text{ar}^2(\hat{\ell}_1) = \Theta(r^4) + \Theta(\ell_{\text{as}}(\gamma)r^3(\gamma)) + \Theta(\ell_{\text{as}}^2(\gamma)r^2(\gamma)) = \Theta(\ell_{\text{as}}^2(\gamma)r^2(\gamma))$$

Therefore, the relative error is $\mathbb{V}\text{ar}(S_n^2)/\mathbb{V}\text{ar}^2(\hat{\ell}_1) = \Theta(\ell_{\text{as}}(\gamma)/r(\gamma))$. By Lemma 4 there exists an $\alpha > 1$ such that

$$\frac{r(\gamma)}{\ell_{\text{as}}(\gamma)} = \frac{r(\gamma)}{\ell_{\text{as}}(\gamma^\alpha)} \times \frac{\ell_{\text{as}}(\gamma^\alpha)}{\ell_{\text{as}}(\gamma)} = o(1) \times O\left(\exp(-\tfrac{(\alpha^2-1)\ln^2(\gamma)}{2\sigma^2})\right),$$

which shows that $\frac{\ell_{\text{as}}(\gamma)}{r(\gamma)}$ grows at least at the exponential rate $\exp(\frac{(\alpha^2-1)\ln^2(\gamma)}{2\sigma^2})$. □

This completes the proof. The proof also fills in the omitted details for [13, Proposition 1].

Proof of Lemma 2

Proof. First we show 1. To this end, recall that $\boldsymbol{X} = \exp(\boldsymbol{Y})$, where $\boldsymbol{Y} \sim \mathsf{N}(\boldsymbol{\nu}, \Sigma)$. Further, recall the well-known property (which is strengthened in Lemma 4) that for $i \neq j$ and $\text{Corr}(Y_i, Y_j) < 1$, the pair Y_i, Y_j is asymptotically independent in the sense that

$$\mathbb{P}(Y_i > \gamma | Y_j > \gamma) = o(1), \qquad \gamma \uparrow \infty.$$

In fact, Lemma 4 shows that this decay to zero is exponential. The consequences of this are $\mathbb{P}(\max_i Y_i > \gamma) \simeq \sum_i \mathbb{P}(Y_i > \gamma)$, and

$$\mathbb{P}(Y_k > \gamma, \max_{i \neq k} Y_i > \gamma) = o(\mathbb{P}(Y_k > \gamma)).$$

With these properties, we then have the lower bound:

$$\begin{aligned} \mathbb{P}(S > \gamma, X_k = M) &\geq \mathbb{P}(X_k = M > \gamma) \\ &\geq \mathbb{P}(X_k > \gamma, \max_{j \neq k} X_j < \gamma) \\ &= \mathbb{P}(Y_k > \ln\gamma, \max_{j \neq k} Y_j < \ln\gamma) \\ &= \mathbb{P}(Y_k > \ln\gamma) + o(\mathbb{P}(Y_k > \ln\gamma)) \\ &= \mathbb{P}(X_k > \gamma) + o(\mathbb{P}(X_k > \gamma)). \end{aligned}$$

Next, using the result $\mathbb{P}(S > \gamma, X_k = M < \ln\gamma) = o(\mathbb{P}(X_k > \ln\gamma))$ from Lemma 3, we also have the analogous upper bound:

$$\begin{aligned}\mathbb{P}(S > \gamma, X_k = M) &= \mathbb{P}(X_k = M > \gamma) + \mathbb{P}(S > \gamma, X_k = M < \gamma)\\ &\leq \mathbb{P}(X_k > \gamma) + \mathbb{P}(S > \gamma, X_k = M < \gamma)\\ &= \mathbb{P}(X_k > \gamma) + o(\mathbb{P}(X_k > \gamma)),\end{aligned}$$

whence we conclude that $\mathbb{P}(S > \gamma, X_k = M) \simeq \mathbb{P}(X_k > \gamma)$.

Next, we show point 2. Using the facts that: (1) the fewer the active constraints in any solution, the closer its minimum is to zero (without constraints the minimum of (3.12) is zero); (2) any solution satisfies the *Karush-Kuhn-Tucker* (KKT) necessary conditions:

$$\begin{aligned}\Sigma^{-1}\boldsymbol{\mu} - \lambda_1\nabla g_1(\boldsymbol{\mu}) - \lambda_2\nabla g_2(\boldsymbol{\mu}) &= \mathbf{0}\\ \boldsymbol{\lambda} \geq \mathbf{0}, \quad \boldsymbol{g}(\boldsymbol{\mu}) \geq \mathbf{0}, \quad \boldsymbol{\lambda}^\top \boldsymbol{g}(\boldsymbol{\mu}) &= 0,\end{aligned}$$

we can verify by direct substitution that $\boldsymbol{\mu}^*$ satisfies the KKT conditions asymptotically as $\gamma \uparrow \infty$ and that it causes only one constraint to be active ($g_1(\boldsymbol{\mu}^*) = o(1)$). Moreover, it yields the asymptotic minimum:

$$\frac{1}{2}(\boldsymbol{\mu}^*)^\top\Sigma^{-1}\boldsymbol{\mu}^* = \frac{(\ln(\gamma) - \nu_k)^2}{2\sigma_k^4}\boldsymbol{e}_k^\top\Sigma\Sigma^{-1}\Sigma\boldsymbol{e}_k = \frac{(\ln(\gamma) - \nu_k)^2}{2\sigma_k^2}$$

Finally, we show point 3, which is the linchpin of the proposed methodology. To this end, consider the $(m+1)$-st moment with $\boldsymbol{\mu} \to \boldsymbol{\mu}^*$ as $\gamma \uparrow \infty$:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\mu}}\hat{\hbar}_k^{m+1} = \mathbb{E}_{\mathbf{0}}\hat{\hbar}_k^m &= \mathbb{E}\exp\left(\frac{m\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}}{2} - m\boldsymbol{\mu}^\top\Sigma^{-1}(\boldsymbol{Y} - \boldsymbol{\nu})\right)\mathbb{I}\{S > \gamma, X_k = M\}\\ &= \exp\left(\frac{(m^2+m)\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}}{2}\right)\mathbb{P}_{-m\boldsymbol{\mu}}(S > \gamma, X_k = M)\\ &\simeq \exp\left(\frac{(m^2+m)(\ln(\gamma)-\nu_k)^2}{2\sigma_k^2}\right)\mathbb{P}_{-m\boldsymbol{\mu}^*}(S > \gamma, X_k = M).\end{aligned}$$

Next, notice that the measure $\mathbb{P}_{-m\boldsymbol{\mu}^*}$ is equivalent to first simulating

$$Y_k \sim \mathsf{N}(\nu_k - m(\ln(\gamma) - \nu_k), \sigma_k^2),$$

and then, given $Y_k = y_k$, simulating all the rest of the components, denoted $\boldsymbol{Y}_{-k}$, from the nominal Gaussian density $\phi_\Sigma(\boldsymbol{y} - \boldsymbol{\nu})$ conditional on $Y_k = y_k$, that is, $\boldsymbol{Y}_{-k} \sim \phi_\Sigma(\boldsymbol{y} - \boldsymbol{\nu}|y_k)$. In other words, asymptotically, the effect of the change of measure induced by (3.12) is to modify the marginal distribution of X_k only. Thus, repeating the same argument used to prove part 1, we have

$$\mathbb{P}_{-m\boldsymbol{\mu}^*}(S > \gamma, X_k = M) \simeq \mathbb{P}_{-m\boldsymbol{\mu}^*}(Y_k > \ln\gamma) = \overline{\Phi}\left(\frac{(m+1)(\ln\gamma - \nu_k)}{\sigma_k}\right).$$

Therefore, as $\gamma \uparrow \infty$,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\mu}} \hat{\hbar}_k^{m+1} &\simeq \exp\left(\tfrac{(m^2+m)(\ln(\gamma)-\nu_k)^2}{2\sigma_k^2}\right)\overline{\Phi}\left(\tfrac{(m+1)(\ln\gamma-\nu_k)}{\sigma_k}\right)\\ &= \Theta\left(\tfrac{1}{\ln\gamma}\exp\left(-\tfrac{(m+1)(\ln(\gamma)-\nu_k)^2}{2\sigma_k^2}\right)\right) = \Theta(\ln^m(\gamma)\hbar_k^{m+1}).\end{aligned}$$

Then, the part 3 of Lemma 2 follows from putting $m = 1$, and observing that

$$\frac{\mathbb{V}\text{ar}(\hat{\hbar}_k)}{\hbar_k^2} = \frac{\mathbb{E}_{\boldsymbol{\mu}}\hat{\hbar}_k^2}{[\mathbb{P}(S>\gamma, X_k = M)]^2} - 1 \simeq \frac{\mathbb{E}_{\boldsymbol{\mu}}\hat{\hbar}_k^2}{[\mathbb{P}(X_k>\gamma)]^2} - 1 = \Theta(\ln(\gamma)).$$

□

Lemma 3. We have $\mathbb{P}(S > \gamma, X_k = M < \gamma) = o(\mathbb{P}(X_k > \gamma))$ as $\gamma \uparrow \infty$.

Proof. Let $\beta \in (0,1)$ and $M_{-k} = \max_{j\neq k} X_j$. Then, using the facts:

$$\frac{\overline{\Phi}(\ln(\gamma-\gamma^\beta))}{\overline{\Phi}(\ln\gamma)} \simeq \exp\left(-\frac{\ln^2(\gamma-\gamma^\beta)-\ln^2(\gamma)}{2}\right)\frac{\gamma-\beta\gamma^\beta}{\gamma-\gamma^\beta}$$

and

$$\ln^2(\gamma)-\ln^2(\gamma-\gamma^\beta) \simeq 2\frac{\ln(\gamma)}{\gamma^{1-\beta}} + o\left(\frac{\ln(\gamma)}{\gamma^{1-\beta}}\right),$$

we obtain $\overline{\Phi}(\ln(\gamma-\gamma^\beta)) \simeq \overline{\Phi}(\ln\gamma)$ for any $\beta \in (0,1)$. More generally,

$$\mathbb{P}(\ln(\gamma-\gamma^\beta) \leq Y_k \leq \ln\gamma) = o(\mathbb{P}(Y_k > \ln\gamma)).$$

Then, we have $\mathbb{P}(S > \gamma, X_k = M < \gamma) =$

$$\begin{aligned}&= \mathbb{P}(M_{-k} > \gamma^\beta, S > \gamma, X_k = M < \gamma) + \mathbb{P}(M_{-k} < \gamma^\beta, S > \gamma, X_k = M < \gamma)\\ &\leq \mathbb{P}(\gamma^\beta < M_{-k} < X_k < \gamma) + \mathbb{P}(M_{-k} < \gamma^\beta,\ \gamma-(d-1)M_{-k} < X_k < \gamma)\\ &\leq \mathbb{P}(\gamma^\beta < M_{-k},\ \gamma^\beta < X_k) + \mathbb{P}(\gamma-(d-1)\gamma^\beta < X_k < \gamma).\end{aligned}$$

Since for large enough γ there exists a $\beta' \in (\beta,1)$ such that $(d-1)\gamma^\beta < \gamma^{\beta'}$, we have

$$\mathbb{P}(\gamma-(d-1)\gamma^\beta < X_k < \gamma) \leq \mathbb{P}(\gamma-\gamma^{\beta'} < X_k < \gamma) = o(\mathbb{P}(X_k > \gamma))$$

The proof will then be complete if we can find a $\beta \in (0,1)$, such that $(u = \ln\gamma)$.

$$\mathbb{P}(M_{-k} > \gamma^\beta, X_k > \gamma^\beta) = \mathbb{P}(\max_{j\neq k} Y_j > \beta u, Y_k > \beta u) = o(\mathbb{P}(Y_k > u)).$$

Since $\mathbb{P}(\max_{j\neq k} Y_j > \beta u, Y_k > \beta u) = O\left(\sum_{j\neq k}\mathbb{P}(Y_j > \beta u, Y_k > \beta u)\right)$, the last is equivalent to showing that the bivariate normal probability $\mathbb{P}(Y_j > \beta u, Y_k > \beta u) = o(\mathbb{P}(Y_k > u))$ for some $\beta \in (0,1)$. This last part then follows from Lemma 4, which completes the proof. □

Lemma 4 (Gaussian Tail Probability)**.** Let $Y_1 \sim \mathsf{N}(\nu_1, \sigma_1^2)$ and $Y_2 \sim \mathsf{N}(\nu_2, \sigma_2^2)$ be jointly bivariate normal with correlation coefficient $\rho \in (-1, 1)$. Then, there exists an $\alpha > 1$ such that

$$\mathbb{P}(Y_1 > \gamma, Y_2 > \gamma) = o(\mathbb{P}(Y_1 > \alpha\gamma) \wedge \mathbb{P}(Y_2 > \alpha\gamma)),$$

where $a \wedge b$ stands for $\min\{a, b\}$.

Proof. Without loss of generality, we may assume that $\sigma_1 > \sigma_2$, so that

$$\mathbb{P}(Y_1 > \alpha\gamma) \wedge \mathbb{P}(Y_2 > \alpha\gamma) \simeq \mathbb{P}(Y_2 > \alpha\gamma) = \Theta(\gamma^{-1} \exp(-\tfrac{(\alpha\gamma-\nu_2)^2}{2\sigma_2^2})).$$

Define the convex quadratic program:

$$\begin{aligned} &\min_{\boldsymbol{y}} \frac{1}{2}\boldsymbol{y}^\top \Sigma^{-1} \boldsymbol{y} \\ &\text{subject to: } \boldsymbol{y} \geq \gamma \mathbf{1} - \boldsymbol{\nu}, \end{aligned} \tag{3.21}$$

where $\Sigma_{11} = \sigma_1^2, \Sigma_{12} = \Sigma_{21} = \rho\sigma_1\sigma_2, \Sigma_{22} = \sigma_2^2$. Denote the solution as $\boldsymbol{y}^*$. Then, we have the following asymptotic result [20]:

$$\mathbb{P}(Y_1 > \gamma, Y_2 > \gamma) = \Theta\left(\gamma^{-d_1} \exp\left(-\tfrac{(\boldsymbol{y}^*)^\top \Sigma^{-1} \boldsymbol{y}^*}{2}\right)\right),$$

where $d_1 \in \{1, 2\}$ is the number of active constraints in (3.21). Next, consider the quadratic programing problem which is the same as (3.21), except that we drop the first constraint (that is, we drop $y_1 \geq \gamma - \nu_1$). The minimum of this second quadratic programing problem is $\frac{(\gamma-\nu_2)^2}{2\sigma_2^2}$, and is achieved at the point $\tilde{\boldsymbol{y}} = ((\gamma - \nu_1)\rho\sigma_2/\sigma_1, \gamma - \nu_2)^\top$. Note that since $\tilde{y}_1 < \gamma - \nu_1$, we have dropped an active constraint. Since dropping an active constraint in a convex quadratic minimization achieves an even lower minimum, we have the strict inequality between the minima of the two quadratic minimization problems:

$$0 < \frac{(\gamma - \nu_2)^2}{2\sigma_2^2} < \frac{(\boldsymbol{y}^*)^\top \Sigma^{-1} \boldsymbol{y}^*}{2}.$$

for any large enough $\gamma > \nu_2$. Hence, after rearrangement of the last inequality, we have

$$\frac{\nu_2 + \sigma_2\sqrt{(\boldsymbol{y}^*)^\top \Sigma^{-1} \boldsymbol{y}^*}}{\gamma} > 1,$$

and therefore there clearly exists an α in the range

$$1 < \alpha < \frac{\nu_2 + \sigma_2\sqrt{(\boldsymbol{y}^*)^\top \Sigma^{-1} \boldsymbol{y}^*}}{\gamma}.$$

For such an α (in the above range), we have

$$\frac{(\alpha\gamma - \nu_2)^2}{2\sigma_2^2} < \frac{(\boldsymbol{y}^*)^\top \Sigma^{-1} \boldsymbol{y}^*}{2}.$$

Therefore, $\exp(-\frac{(\boldsymbol{y}^*)^\top \Sigma^{-1} \boldsymbol{y}^*}{2}) = o\left(\exp(-\frac{(\alpha\gamma-\nu_2)^2}{2\sigma_2^2})\right)$, $\gamma \uparrow \infty$, and the exponential rate of decay of $\mathbb{P}(Y_1 > \gamma, Y_2 > \gamma)$ is greater than that of $\mathbb{P}(Y_2 > \alpha\gamma)$. This completes the proof. □

Bibliography

[1] R. Ambartzumian, A. Der Kiureghian, V. Ohaniana, and H. Sukiasiana. Multinormal probability by sequential conditioned importance sampling: theory and application. *Probabilistic Engineering Mechanics*, 13(4):299–308, 1998.

[2] S. Asmussen. Conditional Monte Carlo for sums, with applications to insurance and finance. *Annals of Actuarial Science*, pages 1–24, 2018.

[3] S. Asmussen and K. Binswanger. Simulation of ruin probabilities for subexponential claims. *Astin Bulletin*, 27(02):297–318, 1997.

[4] S. Asmussen, J. Blanchet, S. Juneja, and L. Rojas-Nandayapa. Efficient simulation of tail probabilities of sums of correlated lognormals. *Annals of Operations Research*, 189(1):5–23, 2011.

[5] S. Asmussen, P.-O. Goffard, and P. J. Laub. Orthonormal polynomial expansions and lognormal sum densities. *arXiv preprint arXiv:1601.01763*, 2016.

[6] S. Asmussen, J. L. Jensen, and L. Rojas-Nandayapa. On the Laplace transform of the lognormal distribution. *Methodology and Computing in Applied Probability*, 18:441–458, 2014.

[7] S. Asmussen, J. L. Jensen, and L. Rojas-Nandayapa. Exponential family techniques for the lognormal left tail. *Scandinavian Journal of Statistics*, 2016.

[8] S. Asmussen and D. P. Kroese. Improved algorithms for rare event simulation with heavy tails. *Advances in Applied Probability*, 38(02):545–558, 2006.

[9] S. Asmussen and L. Rojas-Nandayapa. Asymptotics of sums of lognormal random variables with Gaussian copula. *Statistics & Probability Letters*, 78(16):2709–2714, 2008.

[10] E. Bacry, A. Kozhemyak, and J. F. Muzy. Log-normal continuous cascade model of asset returns: aggregation properties and estimation. *Quantitative Finance*, 13(5):795–818, 2013.

[11] J. Blanchet, S. Juneja, and L. Rojas-Nandayapa. Efficient tail estimation for sums of correlated lognormals. In *Proceedings of the 2008 Winter Simulation Conference*, pages 607–614. IEEE, 2008.

[12] Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016. DOI: 10.1111/rssb.12162.

[13] Z. I. Botev and P. L'Ecuyer. Accurate computation of the right tail of the sum of dependent log-normal variates. In *WSC 2017-Winter Simulation Conference*, 2017.

[14] C. Doerr, N. Blenn, and P. V. Mieghem. Lognormal infection times of online information spread. *PloS one*, 8(5):e64349, 2013.

[15] D. Dufresne. The log-normal approximation in financial and other computations. *Advances in Applied Probability*, 36(3):747–773, 2004.

[16] P. Embrechts, G. Puccetti, L. Rüschendorf, R. Wang, and A. Beleraj. An academic response to basel 3.5. *Risks*, 2(1):25–48, 2014.

[17] L. Fenton. The sum of log-normal probability distributions in scatter transmission systems. *IRE Transactions on Communications Systems*, 8(1):57–67, 1960.

[18] J. A. Gubner. A new formula for lognormal characteristic functions. *IEEE Transactions on Vehicular Technology*, 55(5):1668–1671, 2006.

[19] A. Gulisashvili and P. Tankov. Tail behavior of sums and differences of log-normal random variables. *Bernoulli*, 22(1):444–493, 2016.

[20] E. Hashorva and J. Hüsler. On multivariate Gaussian tails. *Annals of the Institute of Statistical Mathematics*, 55(3):507–522, 2003.

[21] M. B. Hcine and R. Bouallegue. Highly accurate log skew normal approximation to the sum of correlated lognormals. *arXiv preprint arXiv:1501.02347*, 2015.

[22] D. Kortschak and E. Hashorva. Efficient simulation of tail probabilities for sums of log-elliptical risks. *Journal of Computational and Applied Mathematics*, 247:53–67, 2013.

[23] D. Kortschak and E. Hashorva. Second order asymptotics of aggregated log-elliptical risk. *Methodology and Computing in Applied Probability*, 16(4):969–985, 2014.

[24] D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo methods*, volume 706. John Wiley & Sons, 2011.

[25] P. J. Laub, S. Asmussen, J. L. Jensen, and L. Rojas-Nandayapa. Approximating the Laplace transform of the sum of dependent lognormals. *Advances in Applied Probability*, 48(A):203–215, 2016.

[26] E. Limpert, W. A. Stahel, and M. Abbt. Log-normal distributions across the sciences: Keys and clues. *BioScience*, 51(5):341–352, 2001.

[27] A. McNeil, R. Frey, and P. Embrechts. *Quantitative risk management: Concepts, techniques and tools*. Princeton university press, 2015.

[28] M. A. Milevsky and S. E. Posner. Asian options, the sum of lognormals, and the reciprocal gamma distribution. *Journal of financial and quantitative analysis*, 33(03):409–422, 1998.

[29] Q. H. Nguyen and C. Y. Robert. New efficient estimators in rare event simulation with heavy tails. *Journal of Computational and Applied Mathematics*, 261:39–47, 2014.

[30] N. B. Rached, A. Kammoun, M.-S. Alouini, and R. Tempone. Unified importance sampling schemes for efficient simulation of outage capacity over generalized fading channels. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):376–388, 2016.

[31] N. B. Rached, A. Kammoun, M.-S. Alouini, and R. Tempone. On the efficient simulation of the left-tail of the sum of correlated log-normal variates. *arXiv preprint arXiv:1705.07635*, 2017.

[32] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, 3 edition, 2017.

[33] D. Zuanetti, C. Diniz, and J. Leite. A lognormal model for insurance claims data. *REVSTAT-Statistical Journal*, 4(2):131–142, 2006.

Addendum: Vanishing Relative Error Estimator for the Left Tail

In this section, an additional theoretical result and numerical example is provided for the left tail estimator. These results were obtained after submission and thus do not appear in the main manuscript.

Recall the sequential sampling scheme from Section 3.2.1. Denote the measure used to simulate $\boldsymbol{Z}$ with $\boldsymbol{\mu} = 0$ as $\mathbb{P}_{\mathbf{0}}$ and the corresponding expectation (variance) operators as $\mathbb{E}_{\mathbf{0}}$ ($\mathbb{V}\mathrm{ar}_{\mathbf{0}}$). With the above sampling scheme, recall that the unbiased importance sampling estimator of the cdf ℓ (based on a single realization) is:

$$\hat{\ell}_{\mathbf{0}} = \prod_{j=1}^{d} \Phi(\alpha_j(Z_1, \ldots, Z_{j-1})), \qquad \boldsymbol{Z} \sim \mathbb{P}_{\mathbf{0}}. \tag{3.22}$$

Under the condition that $\Sigma_{ii} < \Sigma_{ij}$ for some $i \neq j$ (see [31]), the estimator (3.22) is *strongly efficient*.

Theorem 3 (Vanishing Relative Error)**.** Suppose there exists an index i such that $\Sigma_{ii} < \Sigma_{ij}$ for all $i \neq j$, and, without loss of generality, assume that $i = 1$. Then, the estimator (3.22) enjoys the vanishing relative error property:

$$\lim_{\gamma \downarrow 0} \frac{\mathbb{V}\mathrm{ar}_{\mathbf{0}} \hat{\ell}_{\mathbf{0}}(\gamma)}{\ell^2(\gamma)} = 0$$

Proof. Under the assumption that $\Sigma_{11} < \Sigma_{1j}$ for $j \neq 1$, we have $\ell(\gamma) \simeq \mathbb{P}(X_1 < \gamma)$, see [19] for a proof. Therefore, using the upper bound

$$\mathbb{E}_{\mathbf{0}} \hat{\ell}_{\mathbf{0}}^2(\gamma) = \mathbb{E}_{\mathbf{0}} \prod_{j=1}^{d} \Phi^2(\alpha_j(Z_1, \ldots, Z_{j-1})) \leq \mathbb{E}\Phi^2(\alpha_1) = [\mathbb{P}(X_1 < \gamma)]^2 \simeq \ell^2(\gamma),$$

we have as $\gamma \downarrow 0$,

$$\frac{\mathbb{V}\mathrm{ar}_{\mathbf{0}}(\hat{\ell}_{\mathbf{0}})}{\ell^2} = \frac{\mathbb{E}_{\mathbf{0}} \hat{\ell}_{\mathbf{0}}^2(\gamma)}{\ell^2(\gamma)} - 1 \leq \frac{[\mathbb{P}(X_1 < \gamma)]^2}{\ell^2(\gamma)} - 1 \to 0.$$

□

Consider the numerical example from [31], where $\nu = (4,4,4,4)^\top$ and

$$\Sigma = \begin{bmatrix} 1 & 2 & 2 & 2 \\ 2 & 5 & 4 & 4 \\ 2 & 4 & 4.5 & 4 \\ 2 & 4 & 4 & 4.5 \end{bmatrix}$$

This Σ satisfies the property that $\Sigma_{11} < \Sigma_{1j}$ for all $j \neq 1$. Table 3.11 shows that in this case the gains from using the strongly efficient estimator are significant — the relative error is easily more than a thousand times smaller.

Table 3.11: Comparison between the strongly efficient estimator (3.22) and the weakly efficient estimator (3.1).

			relative error %	
γ	$\widehat{\ell_{\mathbf{0}}}(\gamma)$	$\widehat{\ell}(\gamma)$	RE($\widehat{\ell_{\mathbf{0}}}$)	RE($\widehat{\ell}$)
10	1.91×10^{-2}	1.91×10^{-2}	0.104	0.098
1	2.40×10^{-5}	2.39×10^{-5}	5.05×10^{-2}	0.137
10^{-1}	1.39×10^{-10}	1.39×10^{-10}	1.98×10^{-2}	0.182
10^{-2}	3.78×10^{-18}	3.80×10^{-18}	7.36×10^{-3}	0.218
10^{-3}	5.29×10^{-28}	5.25×10^{-28}	2.51×10^{-3}	0.249
10^{-4}	3.82×10^{-40}	3.81×10^{-40}	9.04×10^{-4}	0.276
10^{-5}	1.42×10^{-54}	1.42×10^{-54}	3.09×10^{-4}	0.300
10^{-6}	2.68×10^{-71}	2.68×10^{-71}	1.58×10^{-4}	0.323

Chapter 4: Authorship Statement

Citation

Laub, P.J., **Salomone, R.**, and Botev, Z.I. (2017). Monte Carlo Estimation of the Density of the Sum of Dependent Random Variables. Submitted to *Mathematics and Computers in Simulation* on 30th November 2017, with subsequent revision submitted on 30th June 2018.

The manuscript with further revisions is included.

Contributions

Overall, I was responsible for 40% of the work. Specifically,

- I was primarily responsible for the conception and initial design of the project.
- I contributed equally with the P.J. Laub and Z.I. Botev to the writing and editing of the paper, technical arguments, and development of the methodology.
- I contributed equally with P.J. Laub in the implementation of the methodology, design of numerical experiments, and the interpretation of their results. P.J. Laub was responsible for the MATHEMATICA implementation of copula examples in Section 4.4, and I was responsible for the marginal density example in Section 4.5, as well as some preliminary tests in MATLAB .
- I contributed equally with P.J. Laub and Z.I. Botev to the editing of the paper.

Chapter 4

Monte Carlo Estimation of the Density of the Sum of Dependent Random Variables

4.1 Introduction

Sums of random variables are fundamental to modeling stochastic phenomena. In finance, risk managers need to predict the distribution of a portfolio's future value which is the sum of multiple assets; similarly, the distribution of the sum of an individual asset's returns over time is needed for valuation of some exotic (e.g. Asian) options [16, 21]. In insurance, the probability of ruin (i.e. bankruptcy) is determined by the distribution of aggregate losses (sums of individual claims of random size) [3, 13]. Lastly, wireless system engineers model total interference in a wireless communications network as the sum of all interfering signals (often lognormally distributed) [10].

In this article, we consider estimating the probability density function (pdf) of sums of random variables (rvs). That is, we wish to estimate the pdf of $S = \sum_{k=1}^{n} X_k$, where $\boldsymbol{X}$ is simulated according to the joint pdf $f_{\boldsymbol{X}}$. A major motivation for obtaining accurate pdf estimates of a rv is to produce confidence intervals for quantiles. For example, the US Nuclear Regulatory Commission specifies regulations in terms of the "95/95" rule, i.e. the upper 95% confidence interval for a 95% quantile [9]. The most common approach [22] is to first estimate the cumulative distribution function (cdf) via

$$\widehat{F}_X(x) = \frac{1}{R} \sum_{r=1}^{R} \mathbb{I}_{\{X^{[r]} \leq x\}} \quad \text{for } X^{[1]}, \ldots, X^{[R]} \overset{\text{iid}}{\sim} F_X,$$

and then the quantile $\widehat{q}_\alpha = \widehat{F}_X^{-1}(\alpha)$. In the obvious notation, we then have the convergence in distribution:

$$\sqrt{R}\,(\widehat{q}_\alpha - q_\alpha) \xrightarrow{\mathcal{D}} \mathsf{N}\left(0, \alpha(1-\alpha)/[f_X(q_\alpha)]^2\right)$$

as $R \to \infty$, where the limiting variance depends on the unknown density $f_X(q_\alpha)$. Thus, any confidence intervals for $\widehat{q}_\alpha$ require estimation of the density $f_X(q_\alpha)$, which is a highly nontrivial problem.

In general, the pdf of a sum of random variables is only available via an n-dimensional convolution. The convolution usually cannot be computed analytically (except in some special cases, e.g., iid gammas or normals) or numerically via quadrature (unless n is very small). Approximations have long been applied to this problem in the iid case for large n. These include the *central limit theorems* [17], *Edgeworth expansions* [7], and *inversion of integral transforms* [1].

An Edgeworth expansion is a generalization of the Central Limit Theorem, which constructs a non-normal approximation based on the first K moments (equivalently, cumulants) of S (the first term of the edgeworth expansion is the Central Limit Theorem approximation, which may not be accurate for small n). Moreover, when the summands of S are dependent, then the moment sequence of S is unknown and needs to be estimated (e.g. by Monte Carlo), and small errors in the approximation of the higher moments can lead to large errors in the approximation. Hence, the method is not fully deterministic (as it may first appear), and requires careful calibration of the value K to avoid numerical instabilities.

Another common method is to construct the Laplace transform (or characteristic function) of S and numerically invert it, using a method such as those described in [1]. However, when the summands are dependent, the Laplace transform of the sum is unknown, so one has to first estimate it (e.g. by Monte Carlo), and then numerically invert this approximation. Specialized methods have been developed for certain marginals and dependence structures (for example, the sum of lognormals case is considered by [14]), but an approach for general distributions is still too difficult.

Finally, Monte Carlo estimators such as Conditional Monte Carlo [2] and the Asmussen–Kroese estimator [6] utilize details of $\boldsymbol{X}$'s distribution to produce unbiased estimates with a dimension–independent rate of convergence of $O(1/n)$.

The purpose of this work is to explore an unbiased Monte Carlo estimator for the problem, with a focus on dependent summands. The estimator is based on treating the pdf estimation problem as a derivative (of the cumulative distribution function) estimation problem. There are several advantages to the proposed estimator. First, we show that in certain settings it enjoys smaller variance than those based on the Conditional Monte Carlo approach. Secondly, the estimator only requires evaluation of the joint pdf up to a (typically unknown) normalizing constant, a situation similar to the application of

Markov chain Monte Carlo. As a result of this, the sensitivity–based approach is useful in estimating posterior marginal densities in Bayesian inference (Section 4.5).

Remark (Notation). In our notation, we use lowercase boldface letters like $\boldsymbol{c}$, $\boldsymbol{x}$, $\boldsymbol{y}$ for non-random vectors and uppercase boldface letters like $\boldsymbol{X}$ for random vectors, and $\mathbf{1}$ for the vector of 1's. If $\boldsymbol{X}$ is of length n, we write: $\boldsymbol{X} = (X_1, \ldots, X_n)^\top$. The inner-product is denoted $\boldsymbol{x} \cdot \boldsymbol{y}$. For a differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$, we write

$$\nabla f(z) = (\partial f(\boldsymbol{x})/\partial x_1, \ldots, \partial f(\boldsymbol{x})/\partial x_n)^\top \big|_{\boldsymbol{x}=z},$$

and use $\nabla_i f(z)$ to denote the i'th component of $\nabla f(z)$.

4.2 Sensitivity Estimator

The estimator is derived from a simple application of Likelihood Ratio method [12, 18], also known as the Score Function method [20]), that is typically used for derivative estimation of performance measures in Discrete Event Systems. We thus tackle the pdf estimation problem by viewing it as a special type of sensitivity analysis. The basic idea appears in [5, Chapter VII, Example 5.7], and our contribution is to consider the approach in a more general setting, weaken a technical condition, and use a control variate to reduce variance.

Assumption 1. The random vector $\boldsymbol{X}$ has a density $f_{\boldsymbol{X}}$, each X_i is supported either on the entire real line or a half-real line, the gradient $\nabla f_{\boldsymbol{X}}$ is a continuous function on the support of $\boldsymbol{X}$, and we have the integrability condition $\mathbb{E}\,|\boldsymbol{X} \cdot \nabla \log f_{\boldsymbol{X}}(\boldsymbol{X})| < \infty$ (here $\boldsymbol{X} \sim F_{\boldsymbol{X}}$). ◇

This assumption is slightly weaker than the one in [5, Prop. 3.5 on page 222], which requires that $|\frac{\mathrm{d}}{\mathrm{d}s}(f_{\boldsymbol{X}}(s\boldsymbol{x})s^n)|$ is uniformly bounded by an $f_{\boldsymbol{X}}$-integrable function of $\boldsymbol{x}$. The proposed estimator is based on the following simple formula, proved in the appendix.

Proposition 1. *For the rv $S = \sum_{i=1}^n X_i = \mathbf{1} \cdot \boldsymbol{X}$ where $\boldsymbol{X}$ satisfies Assumption 1,*

$$f_S(s) = \frac{1}{s}\,\mathbb{E}\left\{\mathbb{I}_{\{\mathbf{1}\cdot\boldsymbol{X}\le s\}}[\boldsymbol{X} \cdot \nabla \log f_{\boldsymbol{X}}(\boldsymbol{X}) + n]\right\} \tag{4.1}$$

for any $s \neq 0$. ◇

It is straightforward to show that (4.1) still holds if the indicators $\mathbb{I}_{\{\cdot\}}$ are replaced by $-(1 - \mathbb{I}_{\{\cdot\}})$. This

suggests the pair of (unbiased) estimators ($\boldsymbol{X} \sim F_{\boldsymbol{X}}$):

$$\underbrace{\frac{1}{s}\mathbb{I}_{\{\mathbf{1}\cdot\boldsymbol{X}\le s\}}\left[\boldsymbol{X}\cdot\nabla\log f_{\boldsymbol{X}}(\boldsymbol{X})+n\right]}_{\widehat{f_1}(s)}, \text{ and } \underbrace{-\frac{1}{s}\mathbb{I}_{\{\mathbf{1}\cdot\boldsymbol{X}>s\}}\left[\boldsymbol{X}\cdot\nabla\log f_{\boldsymbol{X}}(\boldsymbol{X})+n\right]}_{\widehat{f_2}(s)}.$$

We make use of both of these estimators by using one as a base estimator and the difference of the two as a control variate (the difference has a known expectation, namely, zero) [5]. In order to ensure the unbiasedness, we may, for example, obtain the control variate coefficient from a pilot (independent) sample, as explained in Section 4.4.

4.3 Conditional Monte Carlo Methods

In the following Sections 4.3.1 and 4.3.2 we describe the Conditional Monte Carlo approach [2], as well as an extension of the Asmussen–Kroese estimator. We then use these methods as benchmarks to illustrate the performance of the proposed estimator in various settings.

4.3.1 Conditional Monte Carlo estimator

The Conditional Monte Carlo estimator [2] takes the form

$$\widehat{f}_{\text{Cond}}(s) = \frac{1}{n}\sum_{i=1}^{n} f_{X_i|\boldsymbol{X}_{-i}}(s - S_{-i}), \quad \boldsymbol{X} \sim F_{\boldsymbol{X}},$$

where the notation $\boldsymbol{X}_{-i}$ denotes the vector $\boldsymbol{X}$ with the i-th component removed and $S_{-i} = \mathbf{1}\cdot\boldsymbol{X}_{-i}$. This is particularly simple for the independent case, as $f_{X_i|\boldsymbol{X}_{-i}} = f_{X_i}$.

We now examine the dependent case where $\boldsymbol{X}$'s dependence structure is given by an Archimedean copula with generator ψ; i.e., the cdf yields

$$\mathbb{P}(X_1 \le F_{X_1}^{-1}(u_1), \ldots, X_n \le F_{X_n}^{-1}(u_n)) = \phi\big(\textstyle\sum_{i=1}^{n}\psi(u_i)\big), \quad \boldsymbol{u} \in [0,1]^n,$$

where $\phi \equiv \psi^{-1}$ is the functional inverse of ψ. The conditional densities of $\boldsymbol{X}$ can be calculated from the formula ($\phi^{(n)}$ denotes n-th derivative)

$$f_{X_i|\boldsymbol{X}_{-i}}(x_i|\boldsymbol{x}_{-i}) = f_{X_i}(x_i)\psi^{(1)}(F_{X_i}(x_i))\frac{\phi^{(n)}(\sum_{j=1}^{n}\psi(F_{X_j}(x_j)))}{\phi^{(n-1)}(\sum_{j\neq i}\psi(F_{X_j}(x_j)))}. \tag{4.2}$$

Some Archimedean copulas, such as the Clayton and Gumbel–Hougaard copulas, have what is called a Marshall–Olkin representation. An Archimedean copula is in the Marshall–Olkin representation class

if $\phi(s) = \mathbb{E}[e^{-sZ}]$ for some positive rv Z with cdf F_Z. Then an $\boldsymbol{X}$ with this dependence structure can be simulated via

$$\boldsymbol{X} = \left(F_{X_1}^{-1}\left(\phi\left(\frac{E_1}{Z}\right)\right), \ldots, F_{X_n}^{-1}\left(\phi\left(\frac{E_n}{Z}\right)\right)\right), \quad E_i \overset{\text{iid}}{\sim} \mathsf{Exp}(1),\ Z \sim F_Z\,. \tag{4.3}$$

For this case, Asmussen [2, Proposition 8.3] conditions upon the Z as well as $\boldsymbol{X}_{-i}$ to obtain what we call the *extended Conditional Monte Carlo estimator*

$$\widehat{f}_{\text{ExtCond}}(s) = \frac{1}{n}\sum_{i=1}^{n} f_{X_i|\boldsymbol{X}_{-i},Z}(s - S_{-i}), \tag{4.4}$$

where $f_{X_i|\boldsymbol{X}_{-i},Z}(x_i) = -z\psi'(F_i(x_i))f_{X_i}(x_i)e^{-z\psi(F_i(x_i))}$ and $\boldsymbol{X}$ is given by (4.3).

We will use this estimator as a benchmark in our comparisons later on.

4.3.2 Asmussen–Kroese estimator

The Asmussen–Kroese estimator [6] (typically for tail probabilities) is defined as

$$\widehat{F}_{\text{AK}}(s) = 1 - \sum_{i=1}^{n} \overline{F}_{X_i|\boldsymbol{X}_{-i}}(\max\{M_{-i}, s - S_{-i}\})$$

where: $M_{-i} = \max\{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n\}$ and $\overline{F}_{X_i|\boldsymbol{X}_{-i}}(x) = 1 - F_{X_i|\boldsymbol{X}_{-i}}(x)$.

Each $\overline{F}_{X_i|\boldsymbol{X}_{-i}}(\max\{M_{-i}, s - S_{-i}\}) = \overline{F}_{X_i|\boldsymbol{X}_{-i}}(s - S_{-i})$, whenever $M_{-i} + S_{-i} < s$. Thus, we can take the derivative of this piecewise estimator to obtain

$$\widehat{f}_{\text{AK}}(s) = \sum_{i=1}^{n} f_{X_i|\boldsymbol{X}_{-i}}(s - S_{-i})\mathbb{I}_{\{M_{-i}+S_{-i}\le s\}},$$

which can be viewed as alternative conditional estimator. When it is applicable, we use the "extended" form of this estimator where $f_{X_i|\boldsymbol{X}_{-i}}$ is replaced with $f_{X_i|\boldsymbol{X}_{-i},Z}$ as in Section 4.3.1. Notice that the term $1/n$ in (4.4) does not appear here. We remark that (to the best of our knowledge) this variant of the AK estimator for estimation of a density has not been previously considered.

4.4 Numerical Comparisons

In this section, for various distributions of $\boldsymbol{X}$ we compare: i) our proposed method, ii) the conditional MC estimator, and iii) the Asmussen–Kroese (AK) estimator.

We conduct 3 experiments, each one depicted on Figures 4.1 to 4.3 below. Each experiment uses $R = 10^5$ iid replicates of $\boldsymbol{X}$ which are common to all estimators (our estimator uses the first 5% of these to obtain the control variate coefficient, and the remaining samples for pdf estimation).

For each experiment we display a subplot of the estimated density function, as well as the estimated standard deviation and (square root of the) *work-normalized relative variance*: $\mathrm{WNRV}(\widehat{f}(x)) = (\mathrm{CPU\_Time}) \times \mathbb{V}\mathrm{ar}(\widehat{f}(x))/(R[\widehat{f}(x)]^2)$. Here, CPU_Time is the (wall) time taken the by method to produce the estimates for the grid of 50 points.

These examples show sums with dependent summands. When the copula has a Marshall–Olkin representation (4.3) we use it to simulate $\boldsymbol{X}$ and give results for the extended version (4.4) of the conditional MC estimator. All distributions and copulas are parametrized as they are in MATHEMATICA.

Figure 4.1 considers the sum of dependent identically-distributed heavy-tailed variables. The estimates plot shows us that the estimators basically agree with each other, as is to be expected when all methods perform well. In terms of WNRV and standard deviation the sensitivity estimator outperforms the others. Figure 4.2 considers a sum of dependent light-tailed variables. The results here are similar to Figure 4.1. Again, the sensitivity estimator outperforms the others on WNRV and standard deviation.

Figure 4.3 shows the sum of dependent heavy-tailed variables. Instead of the standard multivariate lognormal distribution which has a Gaussian copula, we take the Frank copula. The Frank copula is unique among these tests as it is an Archimedean copula which lacks a Marshall–Olkin representation. Here, the Asmussen–Kroese estimator outperforms the other estimators.

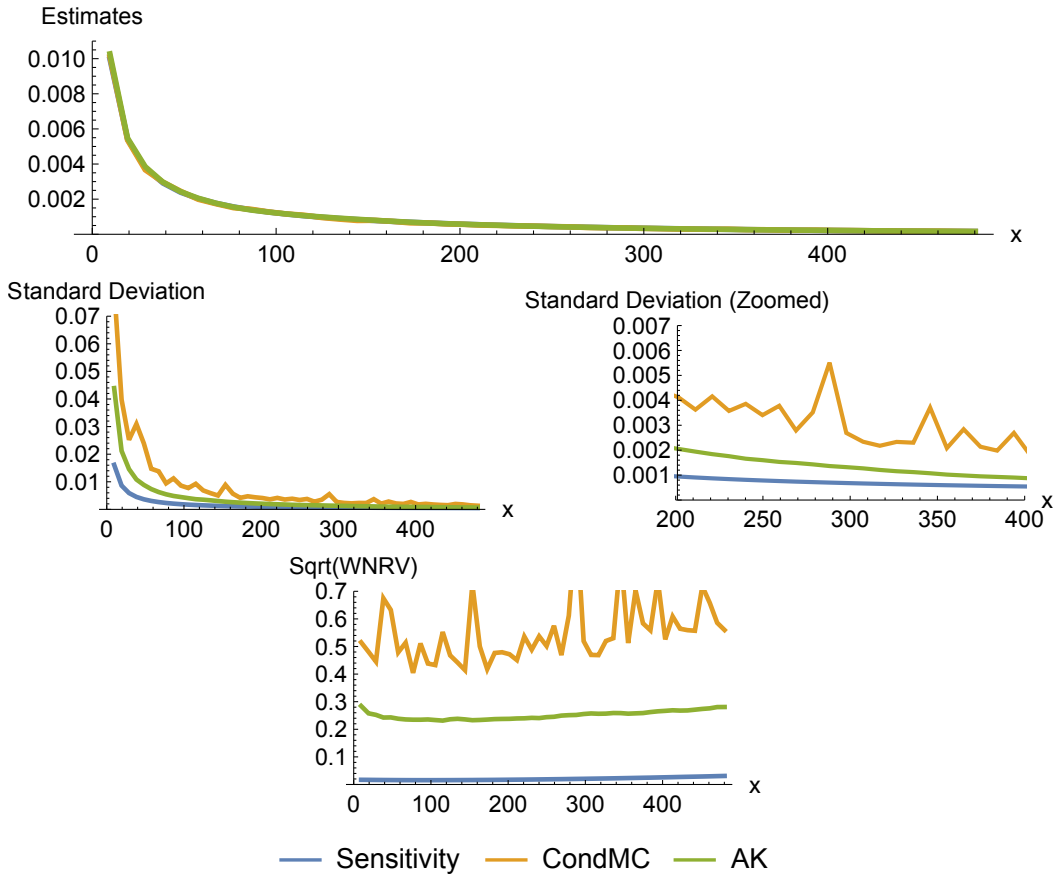


Figure 4.1: Sum of $n = 10$ Weibull$(0.3, 1)$ random variables with a Clayton$(1/5)$ copula.

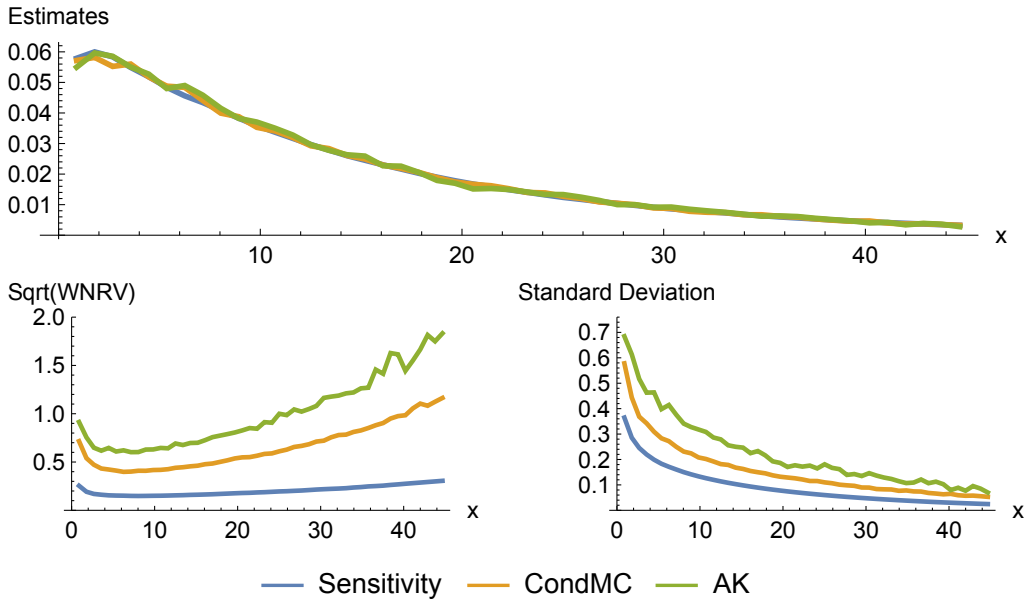


Figure 4.2: Sum of $n = 15$ Exp(1) random variables with a GumbelHougaard(5) copula.

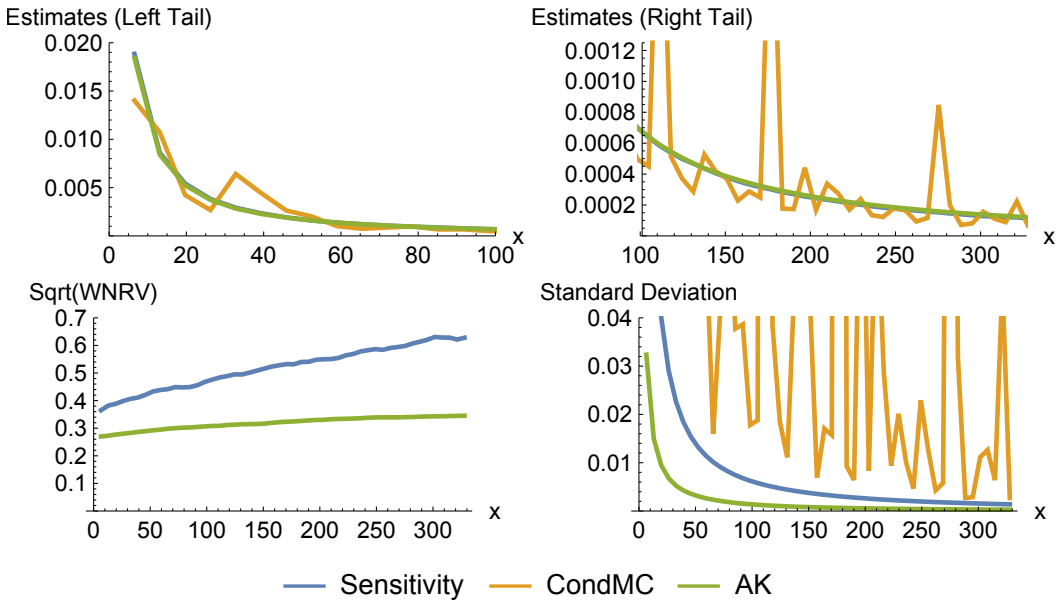


Figure 4.3: Sum of $n = 10$ random variables where $X_i \sim \mathsf{Lognormal}(i - 10, \sqrt{i}\,)$ with a $\mathsf{Frank}(1/1000)$ copula. The choice of marginals mimic the challenging (and somewhat pathological) example considered in [4].

4.5 Extension to Estimation of Marginal Densities

One extension of the sensitivity estimator is in the estimation of marginal densities, which has applications in Bayesian statistics. For an $\boldsymbol{X}$ which satisfies Assumption 1, a similar derivation to the one in Proposition 1 gives the following representation of the marginal densities:

$$f_{X_i}(s) = \frac{1}{s}\,\mathbb{E}\left\{\mathbb{I}_{\{X_i \le s\}}\left(X_i \nabla_i \log f_{\boldsymbol{X}}(\boldsymbol{X}) + 1\right)\right\} \tag{4.5}$$

for $i = 1, \ldots, n$, and $s \neq 0$. We use the estimator with associated control variate that is based on (4.5). A nice feature of the corresponding estimator is that, due to the presence of the $\nabla \log f_{\boldsymbol{X}}(\boldsymbol{x})$ term, the normalizing constant of f need not be known. As an example, we use Markov Chain Monte Carlo to obtain samples from the posterior density of a Bayesian model, and use these to estimate the posterior marginal pdfs with our sensitivity estimator.

We consider the well-known "Pima Indians" dataset (standardized), which records a binary response variable (the incidence of diabetes) for 532 women, along with seven possible predictors. We specify a Logistic Regression model with predictors: *Number of Pregnancies*, *Plasma Glucose Concentration*, *Body Mass Index*, *Diabetes Pedigree Function*, and *Age* (see [11] for justification). The prior is $\boldsymbol{\beta} \sim \mathsf{N}(\mathbf{0}, \mathrm{I})$, as in [11].

To obtain samples from the posterior density, we implement an isotropic Random Walk sampler, using a radially symmetric Gaussian density with $\sigma^2 = 7.5 \times 10^{-3}$ (trace plots indicate this choice mixes well for the model). We ran the Random Walk sampler for 10^3 steps for burn-in, then used the next 2.5×10^4 samples (without any thinning) to obtain a KDE, as well as density estimates using our sensitivity estimator (with control variate). As a benchmark, we compare the accuracy with a KDE constructed using every 50-th sample from an MCMC chain of length $50 \times 5 \times 10^6$. The result of this comparison is depicted in Figure 4.4.

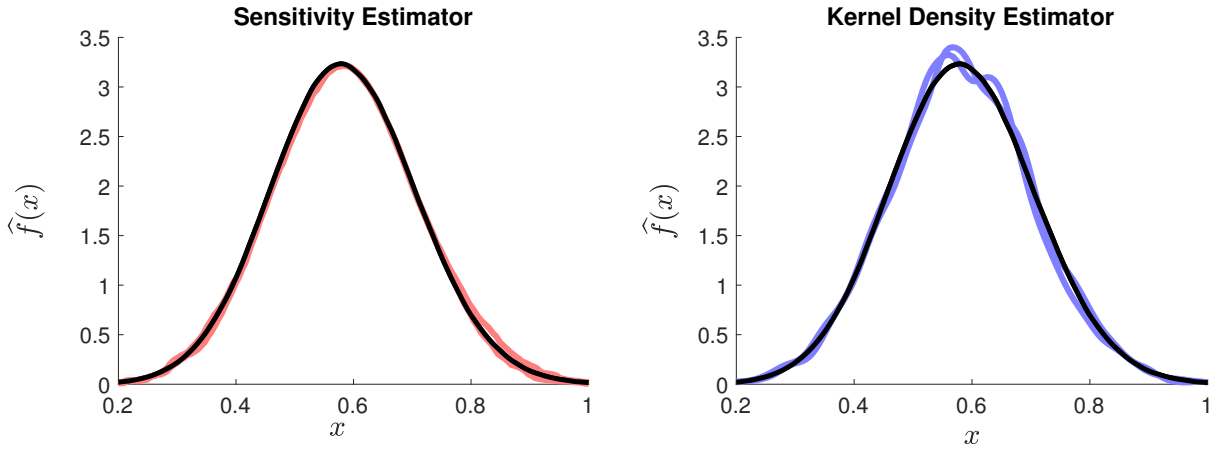


Figure 4.4: Density estimation of posterior marginal corresponding to the coefficient parameter of the *Body Mass Index* predictor variable (results from two runs are shown).

As expected, using the same set of samples, the sensitivity estimator yields a more accurate estimate than KDE. The reason for the lower accuracy of KDE in this context is well-known — a mean square error convergence of $O(n^{-4/5})$, instead of the canonical Monte Carlo rate of $O(n^{-1})$, due to the presence of non-negligible bias in the KDE estimator (see [8], for example).

Remark (Shifting). It is worth noting that, due to the $1/s$ term, it is possible that the sensitivity estimator can have large variance for very small s, even when $F(s)$ or $1 - F(s)$ is not close to zero. This problem can be resolved with a simple linear shift, as follows. If one summand, say X_1, is supported on $\mathbb{R}$, then $f_S(s) = f_{\widetilde{S}}(s - a)$ for $a \in \mathbb{R}$, where $\widetilde{S} = (X_1 + a) + X_2 + \cdots + X_n$. We can then use the original estimator (with shifted values of s and X_1) to obtain estimates of the density of S near or at zero.

4.6 Conclusion

In this paper we derived a sensitivity-based estimator of the pdf of the sum of (dependent) random variables and performed a short numerical comparison. Overall, the numerical comparison indicates

that there isn't a single best estimator in all settings. Nevertheless, the proposed sensitivity estimator will likely be preferable in settings where $\nabla \log f_{\boldsymbol{X}}$ can be computed very quickly, and most useful when the conditional Monte Carlo approach is difficult to apply.

4.7 Appendix: Proof of Proposition 1

There are many ways to derive this formula. One of the simplest is to use the likelihood ratio method ([6, Ch.VII, (4.1)], [12, 18, 20]), which requires the interchange of differentiation and integration. A general sufficient condition for this interchange to be valid is given in [15, Theorem 1]. The proof in this reference uses the dominated convergence theorem, which requires that $|\frac{\mathrm{d}}{\mathrm{d}s} f_{\boldsymbol{X}}(s\boldsymbol{x})s^n|$ is uniformly bounded by an $f_{\boldsymbol{X}}$-integrable function of $\boldsymbol{x}$. In our derivation below, we instead use the Fubini-Tonelli theorem, which only requires the integrability of $|\boldsymbol{x} \cdot \nabla \log f_{\boldsymbol{X}}(\boldsymbol{x})|$ with respect to $f_{\boldsymbol{X}}$.

Define the cdf $F_S(s) = \int_{\mathbf{1}\cdot\boldsymbol{x}\le s} f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, so that the pdf is $f_S(s) = \frac{\mathrm{d}}{\mathrm{d}s}F_S(s)$. The change of variables $\boldsymbol{x} = s\boldsymbol{y}$ yields:

$$F_S(s) = \int_{\mathcal{R}_s} f_{\boldsymbol{X}}(s\boldsymbol{y})|s|^n\,\mathrm{d}\boldsymbol{y} \quad s \neq 0,$$

where the notation $\int_{\mathcal{R}_s}$ means $\int_{\mathbf{1}\cdot\boldsymbol{y}\le 1}$ if $s > 0$, else $\int_{\mathbf{1}\cdot\boldsymbol{y}>1}$ for $s < 0$.

Let $\varphi(s) := \int_{\mathcal{R}_s} \frac{\mathrm{d}}{\mathrm{d}s}(f_{\boldsymbol{X}}(s\boldsymbol{y})|s|^n)\,\mathrm{d}\boldsymbol{y}$. We will use the fact that $\varphi(s) = f_S(s)$ almost everywhere (i.e. except possibly on sets of zero Lebesgue measure) on $s \notin (-\epsilon, \epsilon)$ for an arbitrarily small $\epsilon > 0$.

In order to justify the identity $\varphi(s) = f_S(s)$ (almost everywhere) in the case of $s > \epsilon$ (similar arguments apply for $s < \epsilon$), we use the Fubini-Tonelli theorem for exchanging the order of integration. This exchange holds under the integrability condition

$$\int_\epsilon^s \int_{\mathbf{1}\cdot\boldsymbol{y}\le 1} \left|\frac{\mathrm{d}}{\mathrm{d}t}(f_{\boldsymbol{X}}(t\boldsymbol{y})t^n)\right| \mathrm{d}\boldsymbol{y}\,\mathrm{d}t < \infty \tag{4.6}$$

and the existence of a continuous $\nabla f_{\boldsymbol{X}}$, both of which follow from Assumption 1 (verified at the end of this proof). Using the Fubini-Tonelli theorem [19] we then write:

$$\begin{aligned}\int_\epsilon^s \varphi(t)\,\mathrm{d}t &= \int_\epsilon^s \int_{\mathbf{1}\cdot\boldsymbol{y}\le 1} \frac{\mathrm{d}}{\mathrm{d}t}(f_{\boldsymbol{X}}(t\boldsymbol{y})t^n)\,\mathrm{d}\boldsymbol{y}\,\mathrm{d}t = \int_{\mathbf{1}\cdot\boldsymbol{y}\le 1}\int_\epsilon^s \frac{\mathrm{d}}{\mathrm{d}t}(f_{\boldsymbol{X}}(t\boldsymbol{y})t^n)\,\mathrm{d}t\,\mathrm{d}\boldsymbol{y} \\ &= \int_{\mathbf{1}\cdot\boldsymbol{y}\le 1}(f_{\boldsymbol{X}}(s\boldsymbol{y})s^n - f_{\boldsymbol{X}}(\epsilon\boldsymbol{y})\epsilon^n)\,\mathrm{d}\boldsymbol{y} = F_S(s) - F_S(\epsilon)\end{aligned}$$

Hence, by the fundamental theorem of Calculus, φ equals the derivative of F_S up to a set of measure zero. In other words, $\varphi(s) = f_S(s), s > \epsilon$ almost everywhere. To proceed, we write $\mathrm{sign}(x) = x/|x| = \frac{\mathrm{d}}{\mathrm{d}x}|x|$

$$f_S(s) = \varphi(s) = \int_{\mathcal{R}_s} \left[\boldsymbol{y}\cdot\nabla \log f_{\boldsymbol{X}}(s\boldsymbol{y}) + \tfrac{n\,\mathrm{sign}(s)}{|s|}\right]|s|^n f_{\boldsymbol{X}}(s\boldsymbol{y})\,\mathrm{d}\boldsymbol{y},$$

so after a change of variables $\boldsymbol{y} = \boldsymbol{x}/s$ and using $\text{sign}(x)/|x| = 1/x$, we obtain

$$f_S(s) = \int_{\mathbf{1}\cdot\boldsymbol{x}\le s} \left[\frac{\boldsymbol{x}}{s}\cdot\nabla \log f_{\boldsymbol{X}}(\boldsymbol{x}) + \frac{n}{s}\right] f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \frac{1}{s}\mathbb{E}\left\{\mathbb{I}_{\{\mathbf{1}\cdot\boldsymbol{X}\le s\}}[\boldsymbol{X}\cdot\nabla \log f_{\boldsymbol{X}}(\boldsymbol{X}) + n]\right\}.$$

To verify (4.6), note that after using the change of variable above, it can be upper bounded by

$$\int_\epsilon^s \tfrac{1}{t}\mathbb{E}\left\{\mathbb{I}_{\{\mathbf{1}\cdot\boldsymbol{X}\le t\}}|\boldsymbol{X}\cdot\nabla \log f_{\boldsymbol{X}}(\boldsymbol{X}) + n|\right\}\mathrm{d}t \le (\mathbb{E}\left|\boldsymbol{X}\cdot\nabla \log f_{\boldsymbol{X}}(\boldsymbol{X})\right| + n)\int_\epsilon^s \tfrac{1}{t}\,\mathrm{d}t < \infty,$$

which is bounded by assumption.

Bibliography

[1] J. Abate and W. Whitt. A unified framework for numerically inverting Laplace transforms. *INFORMS Journal on Computing*, 18(4):408–421, 2006.

[2] S. Asmussen. Conditional Monte Carlo for sums, with applications to insurance and finance. *Annals of Actuarial Science*, pages 1–24, 2017.

[3] S. Asmussen and H. Albrecher. *Ruin probabilities*. World Scientific Publishing Co Pte Ltd, 2010.

[4] S. Asmussen, J. Blanchet, S. Juneja, and L. Rojas-Nandayapa. Efficient simulation of tail probabilities of sums of correlated lognormals. *Annals of Operations Research*, 189(1):5–23, 2011.

[5] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*, volume 57. Springer, 2007.

[6] S. Asmussen and D. P. Kroese. Improved algorithms for rare event simulation with heavy tails. *Adv. in Appl. Probab.*, 38(2):545–558, June 2006.

[7] O. E. Barndorff-Nielsen and D. R. Cox. *Asymptotic techniques for use in statistics*. Monographs on Statistics and Applied Probability. Springer Science & Business Media, 1989.

[8] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[9] U. N. R. Commission. *Applying Statistics. U.S. Nuclear Regulatory Commission Report NUREG-1475, Rev.1*. U.S. Nuclear Regulatory Commission, Washington, DC., 2011.

[10] C. Fischione, F. Graziosi, and F. Santucci. Approximation for a sum of on-off lognormal processes with wireless applications. *IEEE Transactions on Communications*, 55(10):1984–1993, 2007.

[11] N. Friel and J. Wyse. Estimating the evidence – a review. *Statistica Neerlandica*, 66(3):288–308, August 2012.

[12] P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Commun. ACM*, 33(10):75–84, Oct. 1990.

[13] S. A. Klugman, H. H. Panjer, and G. E. Willmot. *Loss models: from data to decisions*, volume 715. John Wiley & Sons, 2012.

[14] P. J. Laub, S. Asmussen, J. L. Jensen, and L. Rojas-Nandayapa. Approximating the Laplace transform of the sum of dependent lognormals. *Advances in Applied Probability.*, 2016.

[15] P. L'Ecuyer. A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science*, 36(11):1364–1383, 1990.

[16] A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press, 2nd edition, 2015.

[17] V. V. Petrov. *Sums of independent random variables*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1975.

[18] M. I. Reiman and A. Weiss. Sensitivity analysis for simulations via likelihood ratios. *Operations Research*, 37(5):830–844, 1989.

[19] J. S. Rosenthal. *A first look at rigorous probability theory*. World Scientific Publishing Co Inc, 2006.

[20] R. Y. Rubinstein. The score function approach for sensitivity analysis of computer simulation models. *Mathematics and Computers in Simulation*, 28(5):351–379, 1986.

[21] L. Rüschendorf. *Mathematical risk analysis*. Springer, 2013.

[22] R. J. Serfling. *Approximation theorems of mathematical statistics*. Wiley series in probability and mathematical statistics. Wiley, New York, 1980.

Chapter 5

Conclusion

Several novel Monte Carlo methods have been proposed and their unique advantages over existing methdology demonstrated.

In Chapter 2, a novel class of SMC methods was proposed. By interpreting the Nested Sampling method through the lens of sequential Monte Carlo, novel methodology was introduced that allowed for an unbiased and consistent variant in the non–idealized setting where MCMC is used. The new perspective was also valuable in allowing us to derive an improved version of the of the original Nested Sampling algorithm without requiring the assumption of independent particles. The SMC approach to Nested Sampling has opened up a new avenue through which to approach the theoretical analysis of Nested Sampling, as well as a means to develop methodological extensions or improvements thereof.

Novel SMC sampler calibration methods were introduced in Section 2.5, and applied as part of a simulation study in Section 2.6. The results indicated that NS–SMC is capable of handling difficult statistical problems and performing similarly to the temperature–annealed SMC approach, with the unique advantage of being able to handle problems with phase transitions. While the calibration methods presented were developed out of necessity in an attempt to provide an even–handed comparison between the two SMC methods, they provide a means for practitioners to ensure more robust performance regardless of their chosen SMC approach.[1] The further development of methods for the principled calibration of SMC samplers is another possible avenue of future research.

In Chapter 3, new methods for estimating distributional quantities of the sum of dependent log–normals were introduced. Estimators for the left and right tail, as well as the probability density function were

[1] MATLAB code for the methods in Chapter 2 is available at https://github.com/LeahPrice/SMC-NS.

presented. A number of examples were provided that demonstrated the proposed methodology performs well in a wide variety of settings, including those where existing methods perform poorly. Additionally, a method for exactly sampling conditional on a rare event in the left tail was proposed. A key insight from the chapter is that stronger theoretical properties of rare–event estimators do not necessarily translate into good practical performance, and that great caution should be taken in placing trust solely in asymptotic efficiency results. For this reason, future development of rare-event probability estimation methods would ideally place increased focus on ensuring good practical performance in the pre-limit.

In Chapter 4, an unbiased estimator for the pdf of sums of random variables in a more general setting was explored. Overall, the estimator performed well when compared with other unbiased estimators. An extension for the estimation of marginal pdfs was also provided, and exploring further potential applications of the latter (beyond visualization) would be interesting.